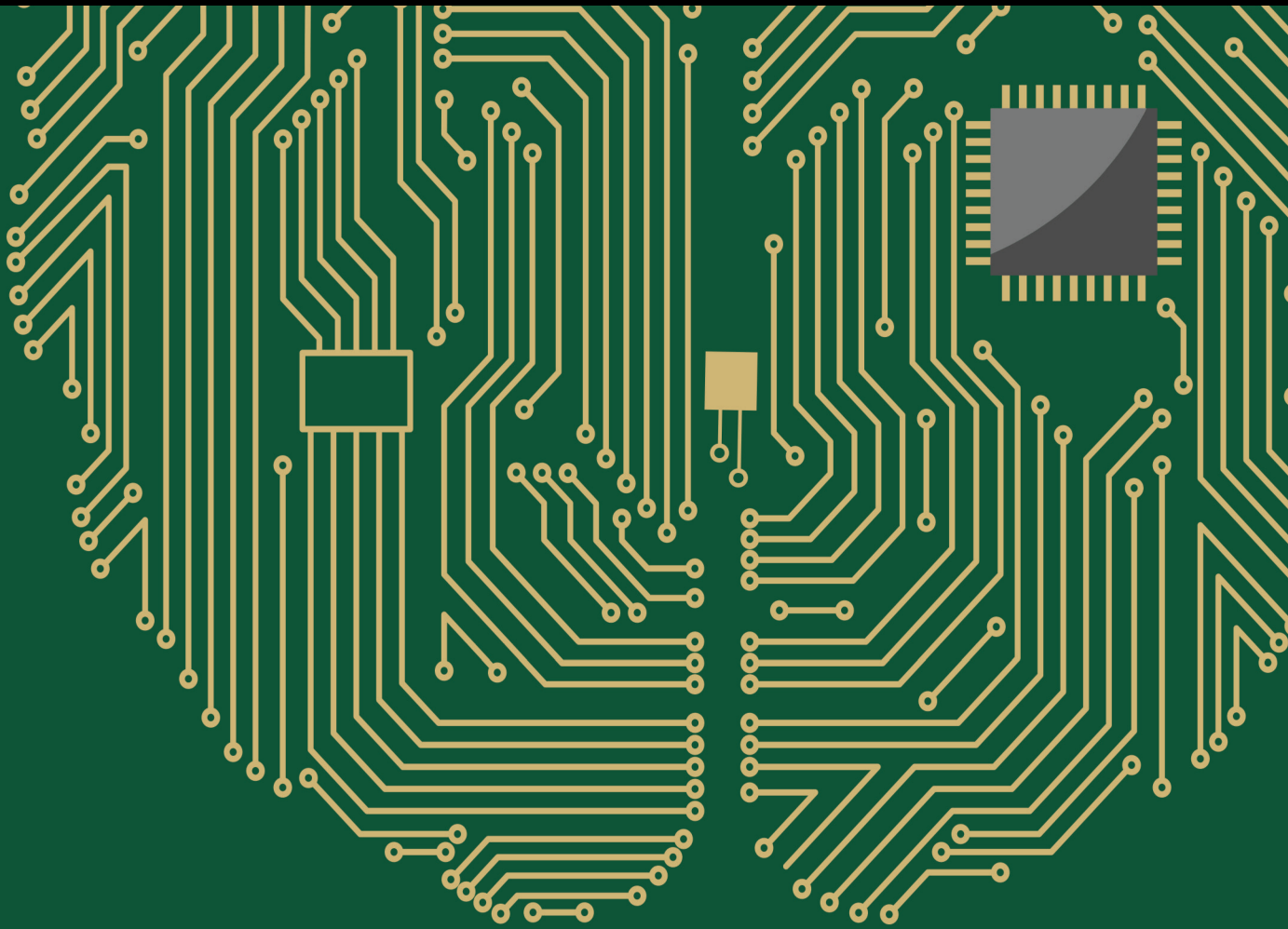


# Interpretation of Machine Learning: Prediction, Representation, Modeling, and Visualization 2021

Lead Guest Editor: Nian Zhang

Guest Editors: Qingshan Liu and Jiang Xiong





---

**Interpretation of Machine Learning:  
Prediction, Representation, Modeling, and  
Visualization 2021**

Computational Intelligence and Neuroscience

---

**Interpretation of Machine Learning:  
Prediction, Representation, Modeling,  
and Visualization 2021**

Lead Guest Editor: Nian Zhang

Guest Editors: Qingshan Liu and Jiang Xiong



---

Copyright © 2021 Hindawi Limited. All rights reserved.

This is a special issue published in "Computational Intelligence and Neuroscience." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

Andrzej Cichocki, Poland

## Associate Editors

Arnaud Delorme, France  
Cheng-Jian Lin , Taiwan  
Saeid Sanei, United Kingdom

## Academic Editors

Mohamed Abd Elaziz , Egypt  
Tariq Ahanger , Saudi Arabia  
Muhammad Ahmad, Pakistan  
Ricardo Aler , Spain  
Nouman Ali, Pakistan  
Pietro Aricò , Italy  
Lerina Aversano , Italy  
Ümit Ağbulut , Turkey  
Najib Ben Aoun , Saudi Arabia  
Surbhi Bhatia , Saudi Arabia  
Daniele Bibbo , Italy  
Vince D. Calhoun , USA  
Francesco Camastra, Italy  
Zhicheng Cao, China  
Hubert Cecotti , USA  
Jyotir Moy Chatterjee , Nepal  
Rupesh Chikara, USA  
Marta Cimitile, Italy  
Silvia Conforto , Italy  
Paolo Crippa , Italy  
Christian W. Dawson, United Kingdom  
Carmen De Maio , Italy  
Thomas DeMarse , USA  
Maria Jose Del Jesus, Spain  
Arnaud Delorme , France  
Anastasios D. Doulamis, Greece  
António Dourado , Portugal  
Sheng Du , China  
Said El Kafhali , Morocco  
Mohammad Reza Feizi Derakhshi , Iran  
Quanxi Feng, China  
Zhong-kai Feng, China  
Steven L. Fernandes, USA  
Agostino Forestiero , Italy  
Piotr Franaszczuk , USA  
Thippa Reddy Gadekallu , India  
Paolo Gastaldo , Italy  
Samanwoy Ghosh-Dastidar, USA

Manuel Graña , Spain  
Alberto Guillén , Spain  
Gaurav Gupta, India  
Rodolfo E. Haber , Spain  
Usman Habib , Pakistan  
Anandakumar Haldorai , India  
José Alfredo Hernández-Pérez , Mexico  
Luis Javier Herrera , Spain  
Alexander Hošovský , Slovakia  
Etienne Hugues, USA  
Nadeem Iqbal , Pakistan  
Sajad Jafari, Iran  
Abdul Rehman Javed , Pakistan  
Jing Jin , China  
Li Jin, United Kingdom  
Kanak Kalita, India  
Ryotaro Kamimura , Japan  
Pasi A. Karjalainen , Finland  
Anitha Karthikeyan, Saint Vincent and the Grenadines  
Elpida Keravnou , Cyprus  
Asif Irshad Khan , Saudi Arabia  
Muhammad Adnan Khan , Republic of Korea  
Abbas Khosravi, Australia  
Tai-hoon Kim, Republic of Korea  
Li-Wei Ko , Taiwan  
Raşit Köker , Turkey  
Deepika Koundal , India  
Sunil Kumar , India  
Fabio La Foresta, Italy  
Kuruva Lakshmana , India  
Maciej Lawrynczuk , Poland  
Jianli Liu , China  
Giosuè Lo Bosco , Italy  
Andrea Loddo , Italy  
Kezhi Mao, Singapore  
Paolo Massobrio , Italy  
Gerard McKee, Nigeria  
Mohit Mittal , France  
Paulo Moura Oliveira , Portugal  
Debajyoti Mukhopadhyay , India  
Xin Ning , China  
Nasimul Noman , Australia  
Fivos Panetsos , Spain



Evgeniya Pankratova , Russia  
Rocío Pérez de Prado , Spain  
Francesco Pistolesi , Italy  
Alessandro Sebastian Podda , Italy  
David M Powers, Australia  
Radu-Emil Precup, Romania  
Lorenzo Putzu, Italy  
S P Raja, India  
Dr.Anand Singh Rajawat , India  
Simone Ranaldi , Italy  
Upaka Rathnayake, Sri Lanka  
Navid Razmjoo, Iran  
Carlo Ricciardi, Italy  
Jatinderkumar R. Saini , India  
Sandhya Samarasinghe , New Zealand  
Friedhelm Schwenker, Germany  
Mijanur Rahaman Seikh, India  
Tapan Senapati , China  
Mohammed Shuaib , Malaysia  
Kamran Siddique , USA  
Gaurav Singal, India  
Akansha Singh , India  
Chiranjibi Sitaula , Australia  
Neelakandan Subramani, India  
Le Sun, China  
Rawia Tahrir , Iraq  
Binhua Tang , China  
Carlos M. Travieso-González , Spain  
Vinh Truong Hoang , Vietnam  
Fath U Min Ullah , Republic of Korea  
Pablo Varona , Spain  
Roberto A. Vazquez , Mexico  
Mario Versaci, Italy  
Gennaro Vessio , Italy  
Ivan Volosyak , Germany  
Leyi Wei , China  
Jianghui Wen, China  
Lingwei Xu , China  
Cornelio Yáñez-Márquez, Mexico  
Zaher Mundher Yaseen, Iraq  
Yugen Yi , China  
Qiangqiang Yuan , China  
Miaolei Zhou , China  
Michal Zochowski, USA  
Rodolfo Zunino, Italy

# Contents




## **Deep Learning Based on Hierarchical Self-Attention for Finance Distress Prediction Incorporating Text**

Sumei Ruan , Xusheng Sun , Ruanxingchen Yao , and Wei Li   
Research Article (11 pages), Article ID 1165296, Volume 2021 (2021)








## **A Hierarchical View Pooling Network for Multichannel Surface Electromyography-Based Gesture Recognition**

Wentao Wei , Hong Hong , and Xiaoli Wu  
Research Article (13 pages), Article ID 6591035, Volume 2021 (2021)



## **Feature Selection Based on a Large-Scale Many-Objective Evolutionary Algorithm**

Yue Li, Zhiheng Sun , Xin Liu , Wei-Tung Chen, Der-Juinn Horng, and Kuei-Kuei Lai   
Research Article (11 pages), Article ID 9961727, Volume 2021 (2021)




## **Discriminative Codebook Hashing for Supervised Video Retrieval**

Xiaoman Bian , Rushi Lan , Xiaoqin Wang , Chen Chen , Zhenbing Liu , Xiaonan Luo , and Kuei-Kuei Lai   
Research Article (11 pages), Article ID 5845094, Volume 2021 (2021)


## **An Improved Stacked Autoencoder for Metabolomic Data Classification**

Xiaojing Fan, Xiye Wang, Mingyang Jiang , Zhili Pei , and Shicheng Qiao  
Research Article (9 pages), Article ID 1051172, Volume 2021 (2021)

## **Automatic Diagnosis of Alzheimer's Disease and Mild Cognitive Impairment Based on CNN + SVM Networks with End-to-End Training**

Zhe Huang , Minglang Sun , and Chengan Guo   
Research Article (13 pages), Article ID 9121770, Volume 2021 (2021)

## **Diversity Evolutionary Policy Deep Reinforcement Learning**

Jian Liu  and Liming Feng  
Research Article (11 pages), Article ID 5300189, Volume 2021 (2021)


## **A Defect Detection Method for Rail Surface and Fasteners Based on Deep Convolutional Neural Network**

Danyang Zheng , Liming Li , Shubin Zheng , Xiaodong Chai , Shuguang Zhao , Qianqian Tong , Ji Wang , and Lizheng Guo  
Research Article (15 pages), Article ID 2565500, Volume 2021 (2021)





## **Indoor Acoustic Signals Enhanced Algorithm and Visualization Analysis**

Suqing Yan , Xiaonan Luo, Xiyang Sun , Jianming Xiao , and Jingyue Jiang  
Research Article (13 pages), Article ID 7592064, Volume 2021 (2021)

## **LSM-SEC: Tongue Segmentation by the Level Set Model with Symmetry and Edge Constraints**



Shanshan Gao , Ningning Guo, and Deqian Mao  
Research Article (14 pages), Article ID 6370526, Volume 2021 (2021)

### **A Multiattention-Based Supervised Feature Selection Method for Multivariate Time Series**

Li Cao , Yanting Chen , Zhiyang Zhang , and Ning Gui 






Research Article (10 pages), Article ID 6911192, Volume 2021 (2021)

### **A Single Target Grasp Detection Network Based on Convolutional Neural Network**

Longzhi Zhang  and Dongmei Wu 

Research Article (12 pages), Article ID 5512728, Volume 2021 (2021)

### **3D M-Net: Object-Specific 3D Segmentation Network Based on a Single Projection**

Xuan Li , Sukai Wang , Xiaodong Niu , Liming Wang , and Ping Chen 





Research Article (13 pages), Article ID 5852595, Volume 2021 (2021)

### **A New Hybrid Forecasting Model Based on SW-LSTM and Wavelet Packet Decomposition: A Case Study of Oil Futures Prices**

Jie Wang  and Jun Wang


Research Article (22 pages), Article ID 7653091, Volume 2021 (2021)

### **A Robust Context-Based Deep Learning Approach for Highly Imbalanced Hyperspectral Classification**

Juan F. Ramirez Rochac , Nian Zhang , Lara A. Thompson , and Tolessa Deksissa 



Research Article (17 pages), Article ID 9923491, Volume 2021 (2021)

### **Detection of Oil Spill Using SAR Imagery Based on AlexNet Model**

Xinzhe Wang, Jiayu Liu, Shuai Zhang, Qiwen Deng, Zhuo Wang, Yunhao Li, and Jianchao Fan 

Research Article (14 pages), Article ID 4812979, Volume 2021 (2021)

### **Quadruplet-Based Deep Cross-Modal Hashing**

Huan Liu, Jiang Xiong , Nian Zhang, Fuming Liu, and Xitao Zou 



Research Article (10 pages), Article ID 9968716, Volume 2021 (2021)

### **Detecting COVID-19 in Chest X-Ray Images via MCFF-Net**

Wei Wang , Yutao Li , Ji Li , Peng Zhang , and Xin Wang 



Research Article (8 pages), Article ID 3604900, Volume 2021 (2021)

### **A Multipulse Radar Signal Recognition Approach via HRF-Net Deep Learning Models**

Ji Li, Huiqiang Zhang, Jianping Ou , and Wei Wang 






Research Article (9 pages), Article ID 9955130, Volume 2021 (2021)

### **Digital Twin-Enabled Online Battlefield Learning with Random Finite Sets**

Peng Wang , Mei Yang, Jiancheng Zhu , Yong Peng, and Ge Li

Research Article (15 pages), Article ID 5582241, Volume 2021 (2021)

### **Hybrid Pyramid Convolutional Network for Multiscale Face Detection**

Shaoqi Hou , Dongdong Fang , Yixi Pan , Ye Li , and Guangqiang Yin 



Research Article (15 pages), Article ID 9963322, Volume 2021 (2021)



# Contents



---

**Improving Loanword Identification in Low-Resource Language with Data Augmentation and Multiple Feature Fusion**

Chenggang Mi , Shaolin Zhu , and Rui Nie



Research Article (9 pages), Article ID 9975078, Volume 2021 (2021)

**A New Random Forest Algorithm Based on Learning Automata**

Mohammad Savargiv , Behrooz Masoumi , and Mohammad Reza Keyvanpour

Research Article (19 pages), Article ID 5572781, Volume 2021 (2021)

**Automatic Detection of Obstructive Sleep Apnea Events Using a Deep CNN-LSTM Model**

Junming Zhang, Zhen Tang, Jinfeng Gao , Li Lin, Zhiliang Liu, Haitao Wu, Fang Liu, and Ruxian Yao 

Research Article (10 pages), Article ID 5594733, Volume 2021 (2021)

## Research Article

# Deep Learning Based on Hierarchical Self-Attention for Finance Distress Prediction Incorporating Text

Sumei Ruan <sup>1</sup>, Xusheng Sun <sup>1</sup>, Ruanxingchen Yao <sup>2</sup>, and Wei Li <sup>1</sup>

<sup>1</sup>School of Finance, Anhui University of Finance and Economics, Bengbu 233030, China

<sup>2</sup>School of Business, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China

Correspondence should be addressed to Wei Li; [liweiaufe@163.com](mailto:liweiaufe@163.com)

Received 5 June 2021; Accepted 22 November 2021; Published 10 December 2021

Academic Editor: Nian Zhang

Copyright © 2021 Sumei Ruan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To detect comprehensive clues and provide more accurate forecasting in the early stage of financial distress, in addition to financial indicators, digitalization of lengthy but indispensable textual disclosure, such as Management Discussion and Analysis (MD&A), has been emphasized by researchers. However, most studies divide the long text into words and count words to treat the text as word count vectors, bringing massive invalid information but ignoring meaningful contexts. Aiming to efficiently represent the text of large size, an end-to-end neural networks model based on hierarchical self-attention is proposed in this study after the state-of-the-art pretrained model is introduced for text embedding including contexts. The proposed model has two notable characteristics. First, the hierarchical self-attention only affords the essential content with high weights in word-level and sentence-level and automatically neglects lots of information that has no business with risk prediction, which is suitable for extracting effective parts of the large-scale text. Second, after fine-tuning, the word embedding adapts the specific contexts of samples and conveys the original text expression more accurately without excessive manual operations. Experiments confirm that the addition of text improves the accuracy of financial distress forecasting and the proposed model outperforms benchmark models better at AUC and  $F2$ -score. For visualization, the elements in the weight matrix of hierarchical self-attention act as scalars to estimate the importance of each word and sentence. In this way, the “red-flag” statement that implies financial risk is figured out and highlighted in the original text, providing effective references for decision-makers.

## 1. Introduction

Financial distress is a global issue of significant concern for all stakeholders. It usually brings a tremendous amount of loss to the related parties [1, 2], which is a severe threat to the stability of global economic systems [3]. Due to loss avoidance, cost saving, and risk management, financial distress prediction is emphasized by potential investors, managers, government officials, and other decision-makers [4]. A scientific and informed prediction model is urgently in need.

Financial distress prediction is a typical binary classification. Most previous researches focused on the application of machine learning methods to gain insights into financial indicators as clues to detect financial risk. For model

construction, on one hand, classic statistical and machine learning methods are applied in feature engineering and classification, such as Naïve Bayesian [5, 6], Support Vector Machine (SVM) [2, 7, 8], and ensemble learning including decision trees based Gradient Boosting Decision Tree (GBDT) [9–12], Random Forest (RF) [13, 14], eXtreme Gradient Boosting (XGB) [13, 15], and Adaptive Boosting (AdaBoost) [16, 17]. On the other hand, various deep learning models are also employed for modeling [18], such as Genetic Algorithm (GA) [6, 19], Convolutional Neural Network (CNN) [20, 21], and Self Organizing Map (SOM) [22]. In short, various models are used to exploit the risk information represented by limited financial ratios to forecast financial distress. This type of research has been quite sufficient.

Financial ratios are calculated in accordance with a specific framework, which provides an opportunity for the company to whitewash the financial situation within a limited range [22]. For example, financially distressed firms tend to undertake more accrual earnings management and less real earnings management [23, 24]. More essentially, forecasting simply covering financial indicators neglects the economic environment and recent business decisions reflected in other disclosure. In summary, the information conveyed by financial data is limited; it is still a challenging task to forecast financial risk accurately.

With the development of artificial intelligence (AI), experts in the field of finance and accounting devote themselves to integrating heterogeneous massive amounts of information by the devices with powerful computing capabilities to predict financial distress more accurately [12, 14, 17]. Relevant research proved that text fusion benefits more accurate identification of financial distress [4, 5, 25]. Since all listed companies obey structural rules to disclose annual reports, the majority of textual information is similar to each other except MD&A. MD&A is closely related to financial distress prediction as it offers investors the review of the company's performance as well as the future potential from the perspective of management [14, 25–27]. Thus, it is reasonable to extract texts from MD&A to represent the nonfinancial information for a supplement. However, the changeable semantic information and unstructured wordy content in MD&A are serious obstacles for text presentation.

There are already some paradigms to quantify text. Most related studies utilize bag-of-words method for text representation [5, 13, 14, 25, 28]. It means that these studies regard the text as a set of scattered segments or isolated words, counting all the terms according to the dictionary to represent text as word count vectors. However, it ignores the contexts hidden inside words and sentences. On the contrary, word embedding through designed neural networks (or pretrained neural networks) preserves the integrity of the article and makes it available to transform the contexts in the corpus into numeric tensors [26, 29, 30]. Compared with training the text embedding neural network based on certain own datasets, the pretraining model with more complicated structures has been trained on a massive standard corpus, with more powerful text representation ability. On a specific natural language processing task, text embedding adaptive for a certain dataset is obtained after fine-tuning the pretrained model. However, in this area, there have been few studies employing advanced pretrained neural networks for end-to-end text representation about financial distress prediction. In this way, Bidirectional Encoder Representations from Transformer (BERT) is introduced for word embedding in the study.

After each word in the text is expressed as a word vector, another major challenge is that the long sequence of information is difficult to remember. In the previous researches on text classification, most researches [31, 32] regard the text as a sequence of words and regard the output from RNN and LSTM as the representation of the text. Generally, multiple hidden layers in RNN and LSTM are

considered to record the contextual information, which is summarized by the output of the last hidden layer. However, for lengthy text information, due to gradient diffusion and gradient explosion, this model tends to forget the previous information in the article. In comparison, attention is better in the classification of long-sequence texts [33, 34]. Only critical information where more weights are assigned is extracted. Although attention does not consider the order of words in the text, it is compensated by the text embedding expressed by the pretrained model, through which the position of each word is recorded.

Aiming to efficiently express the MD&A of large size and provide additional clues to detect financial distress, hierarchical attention neural networks (HAN) are proposed in this study. Since the length of MD&A is usually more than 1000 Chinese words, it is unrealistic to process the entire text as a tedious sentence. We draw on related research on the classification of hierarchical levels, split long texts into sentences, extract the main points of each sentence through attention, and express the sentence vector through the average word vector. On the basis of sentence vectors, the key sentence information is once again refined into text vectors by attention. In this way, the main points of the entire text are effectively expressed in the text vector. This text classification design is especially suitable for the processing of the lengthy MD&A. Based on a combination of original texts and financial ratios, comprehensive experiments have proved that the proposed model outperforms other baseline models trained on word count vectors or financial indicators at AUC and  $F2$ -score.

Our main contributions for financial distress prediction are demonstrated as follows:

For the prediction model, after word embedding, a framework based on hierarchical self-attention neural networks is proposed, competent for the binary classification of texts of large size. Contextual information is embedded as high-dimensional tensors by BERT. Then, attention effectively extracts essential information hierarchically at the word level and the sentence level. Along with financial ratios, as the risk information in MD&A is more effectively and comprehensively extracted, the predictive power of financial distress is enhanced.

For decision support and risk early warning, in consideration of visualization and interpretation, the weights of the attention matrix act as scalars to estimate the importance of linguistic features both at the word and sentence levels. In an article or a sentence belonging to a sample suspected of risk, sentences and words with higher scores will be marked and highlighted as red-flag segments. The parameters learned by the attention network are regarded as the contextual commonality of financially distressed disclosure. For each sample input, this mechanism refines and labels keynotes about risk prediction, providing a direct reference for decision-makers.

## 2. Literature Review

There are different views on the definition of financial distress. Altman [35] first puts forward the multivariate

discriminant analysis to establish a financial distress warning model and proposes the Z-score model to evaluate the possibility of corporate bankruptcy. Beaver [36] defines the default on preferred dividends, and default on debt as financial distress. Altman defines a financial dilemma as a legally bankrupt business. Deakin [37] recognizes only companies that have gone through financial distress, insolvency, or liquidation for the benefit of creditors are in financial distress. Carmichael [38] considers financial distress to be a disruption of obligations in the form of illiquidity, insufficient equity, debt arrears, or insufficient funds. For China's A-share stock market, Shanghai and Shenzhen stock exchanges announced on April 22, 1998, that they would specially treat (ST) stock transactions of listed companies with the abnormal financial state. It mainly refers to two cases: one is the net profit of the listed company audited negative for two consecutive fiscal years, and the other is the net asset per share audited below the face value of the stock in the most recent fiscal year. Usually, a listed company titled ST faces severe financial deterioration, as a sign of financial distress. China's definition of listed companies in financial distress puts weight on profitability before debt defaults, more cautiously.

Based on the indicators covered, the research on financial distress forecasting can be divided into two categories; there are two categories to construct prediction models. On one hand, financial information is simply transformed into financial ratios, and there are intensive studies based on machine learning for feature engineering and classification [10, 16, 20, 39–41]. However, financial statement fraud is frequently committed by cunningly revising financial ratios even legally [24]. Actually, the financial fraudulent activities occurring globally in the past two decades were estimated to amount up to \$5.127 trillion, with associated losses increasing by 56% in the past ten years [26]. It is not convincing enough to adopt financial ratios simply to predict financial distress [23, 24]. On the other hand, more studies begin to focus on nonfinancial information incorporating financial ratios to predict the financial distress to reach higher accuracy. Nonfinancial information, mainly disclosed textual information, has proved to play an important role in financial distress prediction, such as letters to shareholders [28], MD&A [5, 14, 26, 27, 29], or sentiment from annual reports [4, 14, 26], as a supplement to financial numerical information represented by financial ratios only.

There have been methods to accomplish tasks incorporating texts represented by word count vectors. Peng et al. [27] analyze letters to shareholders to build a bag of words (BOW), count word vectors, and propose a scheme for financial distress prediction. Hajek and Henriques [5] deal with counted sentiment words with a random subspace method as an additional feature for financial distress forecasting. Further, word2vec is a comparatively advanced model based on the artificial neural network, which encodes each word as sequential embedded vectors where contexts are included [42]. To

record the sequential information, RNN allows retaining the input sequence as contexts for each segment, which is widely applied for natural language processing (NLP). Long-Short Term Memory (LSTM) [43] is a special type of RNN, comprised of different gates determining corresponding information forgotten or updated and enabling long-term dependencies to be learned. Based on these techniques, Mai et al. [29] employ shallow layers of neural networks for text embedding and apply RNN for text classification. Besides, Du et al. [10] apply pretrained word2vec neural networks for word embeddings and employ models based on bidirectional LSTM (Bi-LSTM) for risk prediction. However, the longer the input sequence is accepted by the RNN, the more likely the training fails to remember the previous part of the article due to gradient vanishment or gradient explosion. Thus, Long-Short Time Memory (LSTM) has made improvements on the basis of RNN, which tries to capture more nonadjacent semantic information through the cell state of a text sequence. Although LSTM introduces a large number of parameters in exchange for more expression length, its expression effect on longer texts is still limited.

Besides, there are two approaches to integrate information derived from the disclosure text and quantitative finance ratios. The first way is to directly combine text and financial indicators in the data set [4, 5, 25, 26]. The latter one is similar to ensemble learning, which reprocesses the separately learned text information and financial information [29], not prevailing for fusing text in financial distress prediction.

### 3. Methodology

The objective of the study is to incorporate text representation and financial ratios to predict financial distress. Generally, financial ratios are structural data and require no excessive preprocessing. Comparatively, unstructured text parsed from annual reports demands to be cleaned and to be transformed into numeric tensors further.

The majority of MD&A exceed 1000 words. It is necessary to disassemble the article into sentences as time distributed series and then encode each part. However, even if the article is split into dozens of sentences, the memory length of convolutional neural networks (CNN, LSTM, etc.) is quite limited. Hence, this article proposed a prediction model based on the hierarchical self-attention after word embedding by the pretrained model, BERT. Composed of 12 encoders and decoders, BERT concludes the word sequences through positional embedding in each component.

The proposed hierarchical framework obtains the final text representation by averaging the sentence-level vectors when each sentence vector is the summary of the word vector. Self-attention treats the fragment most relevant to the other parts as significant information, as a typical efficient approach to deal with long sequences. Subsequently, financial ratios and dense text vectors are combined as final expressions, then identified by the fully connected layer as

positive ones (with financial distress) and negative ones (without financial distress). The flow chart of the proposed method is demonstrated in Figure 1.

**3.1. Hierarchical Attention for Text Representation.** Hierarchical attention (HAN) for multilevel structures is an efficient framework for processing excessively long text information. The framework designed is inspired by Yang et al. [44]. On the one hand, the hierarchical construction divides the text with the large size into small pieces that can be accurately calculated. On the other hand, the model adapts the contexts of the same words or even the same sentences varying in different articles. Further, it endows each word or sentence specific expression according to certain contexts. The architecture of the hierarchical attention is shown in Figure 2.

**3.1.1. Word-Level Self-Attention.** Here is the approach to obtaining sentence-level vectors from the word-level embeddings. The input was scattered isolated Chinese characters without extensive tokenization.  $w_{i\tau}$  denotes the input character  $\tau$  of the sentence  $i$ ,  $\tau \in [1, T]$ , where  $T$  denotes the largest length of a sentence to be encoded.

Scaled dot-product is applied to generate self-attention. Weights in the values ( $V$ ) are obtained by computing scaled dot-products of the query ( $Q$ ) with all keys ( $K$ ). In the word-level attention, the query denotes the embedding result of each word in the sentence  $i$  embedded by BERT,  $Q_i = [e_{i1}, e_{i2}, \dots, e_{iT}]^T$ , and equals the key  $K_i$  and the value  $V_i$ . The weights in square matrices  $W_q, W_k, W_v$  are parameters to be trained in the linear networks.

The element of the dot production matrix  $W_i$  measures the degree of similarity between two words in the word embedding space.  $d_k$  denotes embedding dimensions of words. It is assumed that  $V_i^{\text{attn}}$  is the summary of sentence  $i$ , rewarding the keywords with more weights, while tending to neglect useless words with fewer weights.  $s_i$  is the final sentence-level vector rerepresented by the mean of all word vectors in the word attention  $V_i^{\text{attn}}$ .

$$\begin{aligned} Q_i &= K_i = V_i, \\ W_i &= \text{softmax}\left(\frac{W_q Q_i \cdot W_k K_i^T}{\sqrt{d_k}}\right) = [a_{i1}, a_{i2}, \dots, a_{iT}], \\ V_i^{\text{attn}} &= (Q_i, K_i, V_i) = W_i \cdot W_v V_i = [e'_{i1}, e'_{i2}, \dots, e'_{iT}]^T, \\ s_i &= \frac{\sum_{\tau=1}^T e'_{i\tau}}{T}. \end{aligned} \quad (1)$$

**3.1.2. Sentence-Level Self-Attention.** The way to summarize sentence-level vectors as a final text vector is similar to how to refine word-level inputs from sentence-level input. The text sample  $t$  is composed of sentence queries

$Q_t = [s_1, s_2, \dots, s_L]^T$ , which equals keys  $K_t$  and values  $V_t$ . The weights in square matrices  $U_q, U_k, U_v$  are parameters to be trained in the linear networks. The element in the dot production  $U_t$  measures the similarity between two sentences in the article.  $d_s$  denotes the embedding dimensions of sentences. It is considered that  $V_t^{\text{attn}}$  denotes re-represented information contained in all the sentences of the document  $t$ . In this way, sentence-level attention assigns larger weights to the essential sentences.  $t$  is the final text vector represented by the mean of all the sentence-level vectors in the sentence attention matrix  $V_t^{\text{attn}}$ .

$$\begin{aligned} Q_t &= K_t = V_t, \\ U_t &= \text{softmax}\left(\frac{U_q Q_t \cdot U_k K_t^T}{\sqrt{d_s}}\right) = [a_1, a_2, \dots, a_L], \end{aligned} \quad (2)$$

$$\begin{aligned} V_t^{\text{attn}} &= (Q_t, K_t, V_t) = U_t \cdot U_v V_t = [s'_1, s'_2, \dots, s'_L]^T, \\ t &= \frac{\sum_{l=1}^L e'_l}{L}. \end{aligned}$$

Subsequently, the model takes text vector generated from sentence-level representation as input to concatenate financial ratios.

**3.2. Interpretation.** After normalization by the soft-max function in rows, the element of the dot products in the symmetric matrix  $W_i$  scores the resemblance between word vectors in the sentence  $i$ . If most words in a sentence resemble a certain word  $w_t$ , the word is assumed to be the keyword. The sum of the elements in the column or row  $i$  of the matrix  $W_i$ ,  $\sum_{j \neq i}^T w_{ij}$  ( $j = 1, 2, \dots, T$ ), is regarded as the importance score to evaluate how often the word  $w_t$  is cited in the sentence  $i$ . Notably, the element on the main diagonal is excluded from the evaluation.

$$\text{imp}_{i\tau} = \sum_{j \neq \tau}^T w_{\tau j}. \quad (3)$$

Identically, the evaluation of the importance of each sentence in the article also follows the evaluation above. The sum of the elements in the column  $s$  in the matrix  $U_s$ ,  $\sum_{j \neq s}^L u_{sj}$ , ( $j = 1, 2, \dots, T$ ), is treated as the importance score to measure the frequency of the sentence  $s$  quoted by the other sentences.

$$\text{imp}_s = \sum_{j \neq s}^L u_{sj}. \quad (4)$$

In order to discover significant sentences containing the main idea in one text, the importance score of the sentence is sorted and the top-ranked sentences with high  $\text{imp}_s$  should be concerned by deciders if the sample is labeled with financial distress. If the decision-makers would check the keywords of the red-flagged sentence  $i$ , those words with excessive scores  $\text{imp}_{i\tau}$  should be highlighted.

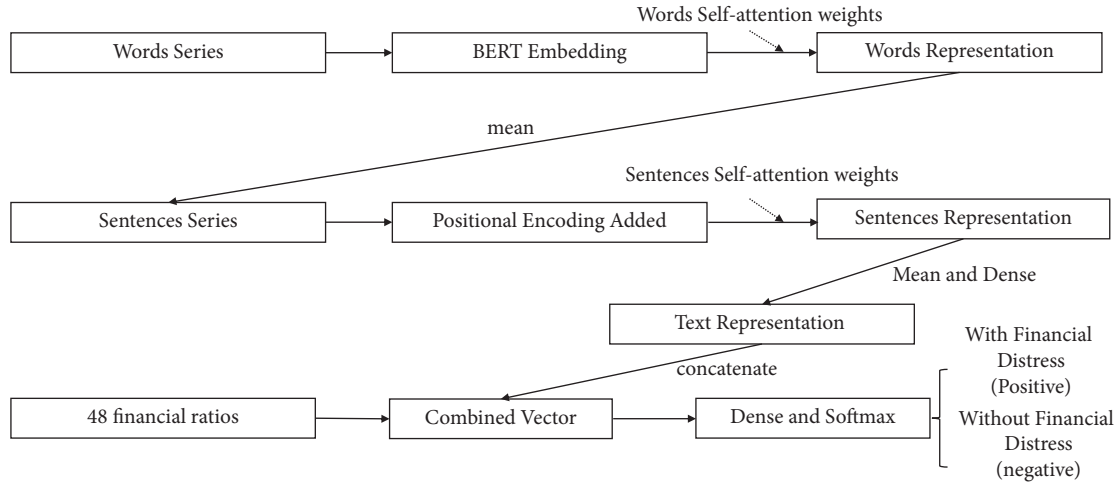


FIGURE 1: Flow chart of the proposed deep learning model.

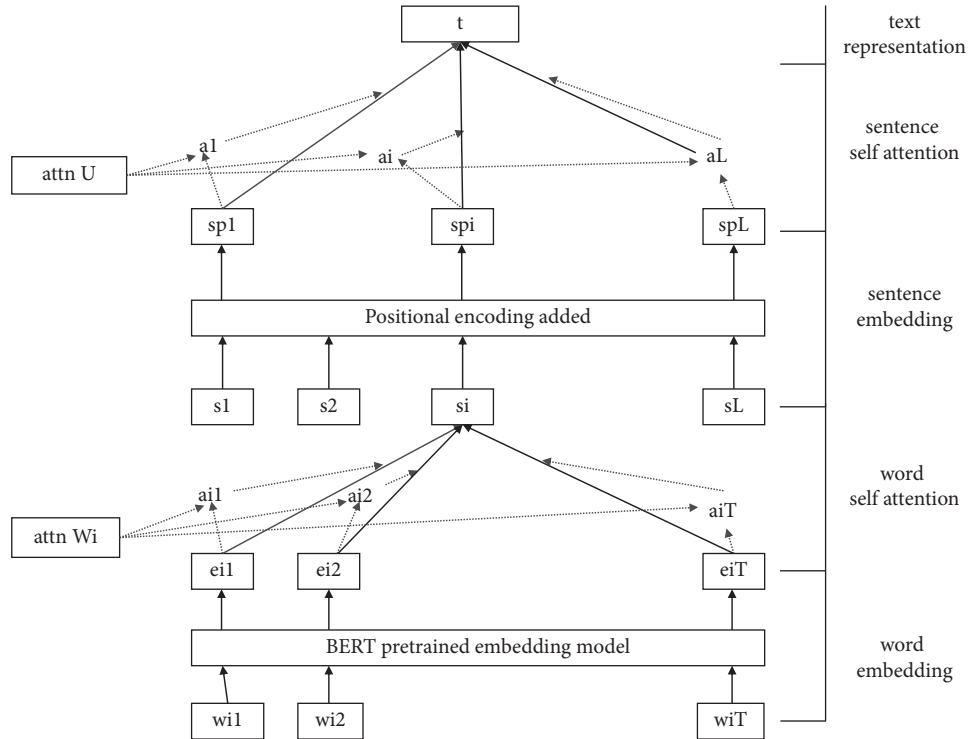


FIGURE 2: The architecture of hierarchical attention networks (HAN).

### 4. Experiment

The data set applied to the proposed model includes both texts of MD&A and financial indicators. Generally, there are two types of listed companies, including companies with special treatment (ST, positive samples) and normal companies (non-ST, negative samples). It is reasonable to mark listed companies to be titled ST or directly delisted as positive samples with financial distress one or two years ahead. Besides, the ratio of positive and negative samples of the original data set is 1:12. Financial distress prediction is challengeable with such a severely imbalanced dataset. Random undersampling is applied in this experiment. By

reducing the number of negative samples, more features derived from positive samples can be noticed by the model.

The core mission is to combine the multisource of information for financial distress forecasting, where one of the difficulties is digitizing text information and combining text representation with financial ratios. The proposed model is compared with the baseline models with word count vector to represent text in the comparative experiments. Besides, in order to present the benefits of information fusion, experiments on financial data simply are also carried out.

Here are details on the implementation of the trial. For the device, the type of graphics processing unit (GPU) applied in this study is NVIDIA TITAN XP. In the process of

processing text, the number of batch training takes a value of 4 with the epoch of 2. For the parameter fine-tuning, the hierarchical learning rate is also adopted,  $2 \times 10^{-5}$  is still proven to be the best learning rate for the pretrained model, and the learning rate of the custom networks is 0.001. With the dropout ratio of text encoding increasing slightly, the recall of positive samples has been effectively improved with acceptable precision.

Besides, 10-fold cross-validation is employed to make sure that there is no violent fluctuation for the generalization performance under the set of hyperparameters. Section 4.3 shows the average of measurements under all the data divisions.

**4.1. Data.** The data in this experiment includes two parts, financial indicators and text MD&A. The text and numeric ratios are directly combined in one data set.

After all, the samples with financial distress are extremely few. In this study, there are 860 positive samples and 11140 negative samples in the original data set listed in Table 1. The ratio of positive samples (with financial distress) to negative samples is 1 : 12. Financial ratios and textual disclosure are included in the research, derived from listed companies in Shanghai and Shenzhen Stock Exchange markets from January 2012 to December 2018.

**4.1.1. Imbalance Treatment.** The effect of learners will decline with the severely unbalanced dataset [7, 10, 45]. It is necessary to preprocess the imbalanced train set. In this study, certain majority samples with negative labels are reduced based on the random undersampling technique (Rus). The final sample distribution is demonstrated in Table 2.

**4.1.2. Text Data.** Annual reports of listed companies are downloaded from Chinese official information query station designated by the China Securities Regulatory Commission information, the earliest securities information professional website, covering more than 3700 listed companies in Shanghai and Shenzhen Stock Exchange markets.

Nonfinancial information, MD&A, is extracted from annual reports. Generally, in addition to the financial indicators calculated by the financial staff, MD&A shows management’s expectations for the company’s prospects. It is assumed that the narrative of the disclosure hints at the company’s governance or development trend [5, 25, 27].

It is worth mentioning that, to prevent overfitting, all company names and geographic locations in documents are filtered by the stop words list. For linear models or decision tree-based models, the BOW is employed to quantify text. For the model proposed in this study, raw text without extensive processing is directly entered as the input. However, the size of the MD&A is excessively large, most of which are beyond 512 words, exceeding the maximum length of the naïve BERT. If all the text in one sample is regarded as a sentence truncated within 512 words, it means

TABLE 1: The sample distribution of the original dataset.

Class	Number
Positive samples (titled “ST” in the next 2 years)	862
Negative samples	11142
Total samples	12004

TABLE 2: The sample distribution of the original dataset.

Class	Number
Positive samples (titled “ST” in the next 2 years)	862
Negative samples	2978
Total samples	3840

that some essential content would be dropped off. Hence, it is necessary to divide the text into hierarchical levels, sentences, and words, to intergrade more information. Due to the limitation of hardware, only 1000 characters or less at the beginning of the document are entered into the proposed model. Each text is staged into 20 sentences within 50 words.

**4.1.3. Quantitative Data.** The quantitative financial indicators are downloaded from the China Stock Market and Accounting Study database (CSMAR). Based on previous researches [5, 10, 12, 24], 48 financial indicators are taken into account, including solvency, ratio structure, operation, profitability, cash flow, risk, development, and the index of per share. Solvency and cash flow describe a company’s ability to repay short-term and long-term debts to prevent bankruptcy. The ratio structure shows the value composition of the company. Operation and profitability evaluate the company’s operating efficiency and performance. Risk measures the multiple that a small change in revenue leads to a huge change in profit due to the existence of fixed costs. Development capability refers to the speed at which a company expands.

**4.2. Metrics.** Financial distress prediction is regarded as a binary classification. There are four predicted results, true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Only TP denotes correct performances to identify samples with financial distress as positive, while FP denotes wrong performances to identify samples without financial distress as positive. Correspondingly, TN indicates correct performances to identify negative samples as negative, and FN denotes wrong performances to mistake positive samples for negative ones.

For the identification of financial distress, the recall of positive samples is crucial. In this study, the model performance is evaluated by a combination of metrics, including the AUC, precision score, recall rate,  $F1$ -score, and  $F2$ -score for positive samples. The  $F$ -score is a combination of precision (the ratio of true positive identified by the classifier to all the positive samples) and recall (the proportion of identified positive samples to all positive samples).

TABLE 3: Evaluation of models on 48 financial ratios.

		AUC	Precision	Recall	F1-score	F2-score
FIN	LR	0.6768	<b>0.8450</b>	0.372	0.5166	0.4189
	SVM	0.7506	0.7768	0.5465	0.6416	0.5809
	XGB	<b>0.8023</b>	0.7222	<b>0.6802</b>	<b>0.7006</b>	<b>0.6882</b>
	RF	0.7829	0.7448	0.6279	0.6814	0.6482
	ANN	0.7337	0.644	0.5581	0.5980	0.5734
	AdaBoost	0.7933	0.7604	0.6453	0.6981	0.6654

TABLE 4: Evaluation of models on both 48 financial ratios and text.

		AUC	Precision	Recall	F1-score	F2-score
FIN + BOW	LR	0.7203	0.8515	0.4826	0.6160	0.5284
	SVM	0.7729	<b>0.8683</b>	0.5258	0.6594	0.5708
	XGB	0.8115	0.7356	0.7035	0.7192	0.7097
	RF	0.7634	0.6357	0.6121	0.6237	0.6167
	ANN	0.7636	0.5720	0.6962	0.6280	0.6672
	AdaBoost	0.8071	0.7214	0.6860	<b>0.7061</b>	0.6986
FIN + TXT	<b>BERT + HAN</b>	<b>0.8218</b>	0.6656	<b>0.7274</b>	0.6951	<b>0.7141</b>

$$\text{precision} = \frac{TP}{TP + FP}, \quad (5)$$

$$\text{recall} = \frac{TP}{TP + FN}.$$

Thus, the  $F$ -score measures how accurate and prudent are those for classifier's performance. Craja et al. [26] estimate the cost of neglecting a positive sample with financial problems to be twice as high as the cost of mistaking a negative sample for a positive one. Effective models should concentrate on the higher recall of positive samples. It is natural to emphasize that recall is more crucial than precision in financial distress prediction. This study employs the  $F2$ -score as a supplement to the  $F1$ -score. Besides, the AUC evaluates the ability to rank positive samples and negative samples in the correct order [10], also serving as an indicator.

$$F1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (6)$$

$$F2 - \text{score} = \frac{(1 + \beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \quad (\beta = 2).$$

**4.3. Comparative Experiment Result.** Multiple sets of comparative experiments are carried out in this part. Generally, there are two groups, models on financial data simply and models on the combination of financial ratios and digitization of texts. The result of experiments on financial data serves as a benchmark to demonstrate the progress of different learners after adding text features. Typical baseline learners, including linear models (LR, SVM), the decision-tree based models (XGB, RF, and AdaBoost), and Multilayer Perceptions (MLP) serve as comparative models.

The evaluation indicators of all learners' performance on different data set divisions are reported. Multiple sets of train

sets and test sets are generated with several random seeds to reduce bias in case of overfitting on the specific splitting.

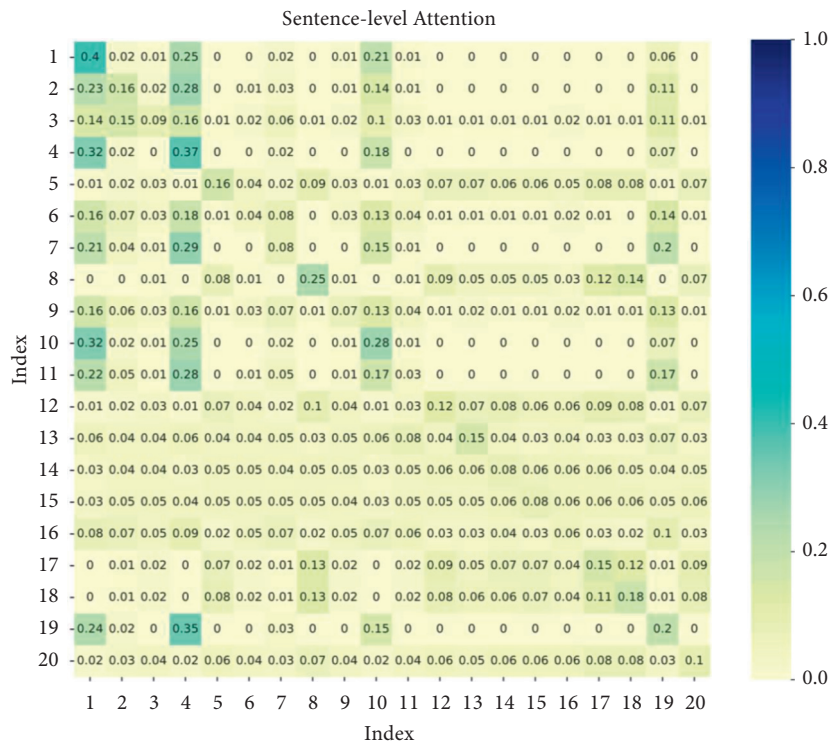
**4.3.1. Modeling on Financial Ratios.** Based on 48 financial indicators, the learning result of control models is shown in Table 3. As mentioned above, in addition to AUC, what should be concentrated on are the indicators of the learner's recognition of positive samples, recall, and  $F2$ -score. For these indicators, decision-tree based models perform well with higher AUC, recall. Especially XGB outperforms the other models in terms of AUC, recall, and  $F2$ -score. Although linear models, LR and SVM, have achieved higher precision, they leave out excessive positive samples, fail to serve as qualified learners in this area. Besides, ANN is composed of two encoders. Each encoder includes two linear layers and a fully connected layer. From the results, the performance of ANN is close to linear learners.

**4.3.2. Modeling on Financial Ratios and Digitalization of Text.** It is the core of this research to intergrade financial indicators and text to predict financial distress. Typical approaches to convert text include BOW and word embedding through neural networks. BOW counts the word frequency in each text according to the dictionary manipulated by chi-square test and pair-words merging. BOW serves as a baseline method. The combined numeric word frequency vector with financial ratios vector is entered into benchmark learners.

As a comparison to BOW, with the pretrained model BERT to represent texts, the result of the comparison experiments is shown in Table 4.

After adding text features, the effects of all models have been improved, with the exception of RF. It is observed that all models have unanimously made progress on the most noteworthy  $F2$ -score. When focusing on the AUC and  $F2$ -score, the proposed model achieved the best results with 82.18% and 71.41%. It can be concluded that when the  $F2$ -





#### I. Overview

During the reporting period, the company's asset restructuring, debt restructuring, share-trading reform and resumption of listing have made substantial progress.

...

The company held the second extraordinary general meeting of shareholders on September 18, 2013, and passed the "Amendment to the Articles of Association" and other proposals. The eighth session of the board of directors was held on September 18, elected the chairman and vice chairman of the new board of directors, and confirmed the appointment of the members of the professional committees of the board and the company's management. The eighth session of the Supervisory Committee was held on September 18, 2013, to elect the chairman of the new Supervisory Committee. On December 31, 2013, the company implemented and completed the equity split reform plan and the debt restructuring and transfer of shares plan. In January 2014, the company completed the registration of new shares and resumed listing on January 10, 2014. The stock abbreviation was changed to "Cobalt Nickel" and the stock code remained unchanged.

In 2013, the company realized operating income of 4,407,771,927.08 yuan, an increase of 249.17% year-on-year, and the total profit was ¥30,634,985.00, attributable to the parent company. The net profit was ¥111,784,706.53. Faced with the unfavorable situation of the long-term low price of non-ferrous metals in 2013 and the continuous increase in the cost of production factors, in order to strive to achieve the company's profitability, the company went all out to do the following work:

1. Carry on researches on market changes, flexibly organize and arrange production, and maintain stable and healthy production and operation. At the same time, increase the trading business of non-ferrous products and strive to achieve the set goals. The company makes every effort to ensure the normal production and ensure that the annual operation rate of its equipment remains above 90%. On the premise of ensuring the quality of nickel sulfate and iron fine powder. We will continue to strengthen market development, stabilize existing customers, actively develop other high-quality customers, and open up new market space. In response to the decline in the price of electrolyzed nickel products, the company deeply analyzed various adverse factors, continuously enhanced the awareness of crises, explored solutions to problems, organized employees, solved practical problems, and worked hard to minimize losses point.

2. Promote the progress of the preliminary work of project construction ...

FIGURE 3: A page from MD&A parsed from a positive sample. In the sentence-level attention, corresponding to the top three total scores of column weights, the three sentences that best summarize the article information are highlighted. Similarly, the keywords in each sentence are also marked according to word-level attention respectively, where word-level attention is not depicted here.

score, which puts weights on the recall rate, is regarded as the core indicator of the financial distress prediction, the proposed model behaves best. When dealing with texts with

intricate internal relationships of intact original documents, deep neural networks (DNN) offer substantial improvement in interpreting the complexity and detect more commonality

shared by positive samples. Our proposed model, BERT + HAN, proves to be a promising alternative method with the performance under a higher recall, which is emphasized by stakeholders.

**4.4. Interpretation Demonstration.** According to assumptions, the documents disclosed by companies facing financial difficulties have a certain contextual commonality instead of the simple frequency of words. These sharing features are summarized, captured by the elaborately designed hierarchical attention mechanism.

Here, the identification of significant sentences and words in a sample facing financial distress is illustrated. In the text-level attention, each row of the matrix has been normalized. The sum of each column is considered to be the total cited score, in other words, the importance of the sentence of the column index. For the example illuminated in Figure 3, sentences with the serial number 1, 4, and 10 are evaluated and marked with the highest scores. In the same way, the keynotes in each sentence are also selected and highlighted with a darker color. The text-level attention and labeled article are displayed in Figure 3. Due to space limitations, the word-level attention matrix is not shown in the picture. Since the text is cleaned, and the sentences with the total number of words less than 50 are merged, the serial number corresponds to the cleaned text and may not correspond to the original sentence one-to-one.

The proposed model not only provides a more powerful financial distress prediction ability, but also the two-step attention mechanism offers an interpretable reference for decision-makers. Visual labeling of suspicious words and sentences offers clues to potential financial distress.

## 5. Discussion

Regarding the textual disclosure of new information as a supplement to financial indicators, a basic prerequisite is that it contains information that is not reflected in the latter, such as management's insights and expectations of the company's outlook. Moreover, companies facing financial distress have potentially similar contextual characteristics in disclosure, difficult to be modified like financial indicators. Our work confirms this, and through the setting of hierarchical attention networks, the exploration of the contextual features mentioned above has been well completed.

Our study introduces the pretrained model BERT with a powerful ability for text representation and employs a hierarchical attention mechanism to disassemble the ultralong text into some shorter sentences for representation and training and, finally, combine the obtained text vector and financial data for financial distress prediction. From the experimental results, our proposed model beats all the benchmark models at the AUC and  $F2$ -score emphasized in the field. Experiments prove that the context of the original text hides clues to financial distress. If these clues are detected, they effectively improve the ability to predict financial distress.

To think further, the plain word2vec based on shallow neural networks and the bag-of-words perhaps have limitations in dealing with the text of large size, and it is difficult for them to capture the intricate and contextual attributes. With the original form of the text remaining, utilizing pretraining models BERT based on deep neural networks with fine-tuning and filtering the key information of long texts hierarchically based on the attention mechanism is a novel idea for analyzing large texts. More importantly, for different samples, attention is targeted to analyze and opt for indispensable features in varying contexts, which is closer to the way people process financial disclosure in reading comprehension. It is more effective than the methods quantifying text with one unified feature scale.

In addition, we have also explored the interpretability of deep neural network models. The attention mechanism provides a way to visualize the key features of all samples. Based on the vector similarity measurement by dot product normalized through soft-max function, we can pick up the key information and encode sentence vectors according to the word-level attention matrix and then refine the text vector through the sentence-level attention, where all the steps are visualized. Illuminating attention to different sentences and words and evaluating importance points, clues of financial distress in the original text can be marked.

We recommend that decision-makers pay more attention to the complex and tedious text disclosures. In particular, we expect that the proposed model can reduce the workload of auditors by filtering out key information. Through tracking and investigation of the clues further, the risk is more likely to be detected in advance.

## 6. Conclusion

Based on heterogeneous information, not only studies in the financial field to predict financial distress are involved, but also artificial intelligence methods to digitize unstructured information are necessary.

The model proposed in this research embeds and expresses the text from the original data at the word and sentence levels and summarizes the final vector representation of the text. Next, the text vector obtained and financial data are entered into the multilayer perceptron and classified. Experiments show that the proposed model beats all the benchmark ones at  $F2$ -score.

Without additional discretion, the potential of the proposed end-to-end deep learning method in information representation and feature engineering has been examined in this study. At the same time, the trained attention mechanism in this study successfully imitates humans to dig keynotes from complex language structures and offers readers with visualization of the "red flag" content as clues of financial distress. Finally, for researchers, research on the time series of corporate disclosure texts and financial indicators based on panel data may still be required. In addition, risk prediction divided by industry segments may be more effective in the application of artificial intelligence in the respective field.

## Data Availability

1. The financial ratios data used to support the findings of this study have been deposited in the CSMAR repository (<https://www.gtarsc.com/>). 2. The annual reports data used to support the findings of this study have been deposited in the CNIFO repository (<http://www.cninfo.com.cn/new/index>).

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This work was funded by Natural Science Foundation of Anhui Province (2008085MG234), University Natural Science Research Project of Anhui Province (KJ2019A0651), and Excellent Young Talents Fund Program of Higher Education Institutions of Anhui Province (gxbjZD2020004).

## References

- [1] A. Mochón, D. Quintana, Y. Sáez, and P. Isasi, "Soft computing techniques applied to finance," *Applied Intelligence*, vol. 29, pp. 111–115, 2008.
- [2] D. Veganzones and E. Séverin, "An investigation of bankruptcy prediction in imbalanced datasets," *Decision Support Systems*, vol. 112, pp. 111–124, 2018.
- [3] V. García, A. I. Marqués, and J. S. Sánchez, "Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction," *Information Fusion*, vol. 47, pp. 88–101, 2019.
- [4] G. Wang, G. Chen, and Y. Chu, "A new random subspace method incorporating sentiment and textual information for financial distress prediction," *Electronic Commerce Research and Applications*, vol. 29, pp. 30–49, 2018.
- [5] P. Hajek and R. Henriques, "Mining corporate annual reports for intelligent detection of financial statement fraud—a comparative study of machine learning methods," *Knowledge-Based Systems*, vol. 128, pp. 139–152, 2017.
- [6] C. H. Cheng, C. P. Chan, and J. H. Yang, "A seasonal time-series model based on gene expression programming for predicting financial distress," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 1067350, 14 pages, 2018.
- [7] J. Sun, H. Fujita, Y. Zheng, and W. Ai, "Multi-class financial distress prediction based on support vector machines integrated with the decomposition and fusion methods," *Information Sciences*, vol. 559, pp. 153–170, 2021.
- [8] Z. Chen, W. Chen, and Y. Shi, "Ensemble learning with label proportions for bankruptcy prediction," *Expert Systems with Applications*, vol. 146, Article ID 113155, 2020.
- [9] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with DTE-SBD: decision tree ensemble based on SMOTE and bagging with differentiated sampling rates," *Information Sciences*, vol. 425, pp. 76–91, 2018.
- [10] X. Du, W. Li, S. Ruan, and L. Li, "CUS-heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection," *Applied Soft Computing*, vol. 97, 2020.
- [11] F. Sigrist and C. Hirnschall, "Grabit: gradient tree-boosted Tobit models for default prediction," *Journal of Banking & Finance*, vol. 102, pp. 177–192, 2019.
- [12] J. Bertomeu, E. Cheynel, E. Floyd, and W. Pan, "Using machine learning to detect misstatements," *Review of Accounting Studies*, vol. 26, no. 2, pp. 468–519, 2021.
- [13] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning," *Electronic Commerce Research and Applications*, vol. 31, pp. 24–39, 2018.
- [14] J. Donovan, J. Jennings, K. Koharki, and J. Lee, "Measuring credit risk using qualitative disclosure," *Review of Accounting Studies*, vol. 26, no. 2, pp. 815–863, 2021.
- [15] B. Lin and R. Bai, "Machine learning approaches for explaining determinants of the debt financing in heavy-polluting enterprises," *Finance Research Letters*, Article ID 102094, 2021.
- [16] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, "Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting," *Information Fusion*, vol. 54, pp. 128–144, 2020.
- [17] Y. Bao, B. Ke, B. Li, Y. J. Yu, and J. Zhang, "Detecting accounting fraud in publicly traded U.S. Firms using a machine learning approach," *Journal of Accounting Research*, vol. 58, no. 1, pp. 199–235, 2020.
- [18] Y. Chen, "BP neural network based on simulated annealing algorithm optimization for financial crisis dynamic early warning model," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 4034903, 11 pages, 2021.
- [19] C.-H. Chou, S.-C. Hsieh, and C.-J. Qiu, "Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction," *Applied Soft Computing*, vol. 56, pp. 298–316, 2017.
- [20] T. Hosaka, "Bankruptcy prediction using imaged financial ratios and convolutional neural networks," *Expert Systems with Applications*, vol. 117, pp. 287–299, 2019.
- [21] K. Shuang, Z. Zhang, J. Loo, and S. Su, "Convolution-deconvolution word embedding: an end-to-end multi-prototype fusion embedding method for natural language processing," *Information Fusion*, vol. 53, pp. 112–122, 2020.
- [22] P. Du Jardin, "A two-stage classification technique for bankruptcy prediction," *European Journal of Operational Research*, vol. 254, no. 1, pp. 236–252, 2016.
- [23] D. Campa and M.-D.-M. Camacho-Miñano, "The impact of SME's pre-bankruptcy financial distress on earnings management tools," *International Review of Financial Analysis*, vol. 42, pp. 222–234, 2015.
- [24] Y. Li, X. Li, E. Xiang, and H. Geri Djajadikerta, "Financial distress, internal control, and earnings management: evidence from China," *Journal of Contemporary Accounting & Economics*, vol. 16, no. 3, Article ID 100210, 2020.
- [25] G. Wang, J. Ma, G. Chen, and Y. Yang, "Financial distress prediction: regularized sparse-based random subspace with ER aggregation rule incorporating textual disclosures," *Applied Soft Computing*, vol. 90, Article ID 106152, 2020.
- [26] P. Craja, A. Kim, and S. Lessmann, "Deep learning for detecting financial statement fraud," *Decision Support Systems*, vol. 139, Article ID 113421, 2020.
- [27] Y. Peng, G. Wang, G. Kou, and Y. Shi, "An empirical study of classification algorithm evaluation for financial risk prediction," *Applied Soft Computing*, vol. 11, no. 2, pp. 2906–2915, 2011.
- [28] Y.-J. Chen and C.-Y. Wu, "Predicting a corporate financial crisis using letters to shareholders," *Soft Computing*, vol. 25, no. 5, pp. 3623–3636, 2021.
- [29] F. Mai, S. Tian, C. Lee, and L. Ma, "Deep learning models for bankruptcy prediction using textual disclosures," *European*

- Journal of Operational Research*, vol. 274, no. 2, pp. 743–758, 2019.
- [30] S. Dong and C. Liu, “Sentiment classification for financial texts based on deep learning,” *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9524705, 9 pages, 2021.
- [31] J. Gu, Z. Wang, J. Kuen et al., “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [32] G. Rao, W. Huang, Z. Feng, and Q. Cong, “LSTM with sentence representations for document-level sentiment classification,” *Neurocomputing*, vol. 308, pp. 49–57, 2018.
- [33] S. Yu, D. Liu, W. Zhu, Y. Zhang, and S. Zhao, “Attention-based LSTM, GRU and CNN for short text classification,” *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 1, pp. 333–340, 2020.
- [34] A. Galassi, M. Lippi, and P. Torrioni, “Attention in natural language processing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, 2021.
- [35] E. I. Altman, “The prediction of corporate bankruptcy: a discriminant analysis,” *The Journal of Finance*, vol. 23, no. 1, p. 193, 1968.
- [36] W. H. Beaver, “Financial ratios as predictors of failure,” *Journal of Accounting Research*, vol. 4, pp. 71–111, 1966.
- [37] E. B. Deakin, “A discriminant analysis of predictors of business failure,” *Journal of Accounting Research*, vol. 10, pp. 167–179, 1972.
- [38] D. R. Carmichael, *The Auditor’s Reporting Obligation: The Meaning and Implementation of the Fourth Standard of Reporting*, American Institute of Certified Public Accountants, Durham, NC, USA, 1972, [https://egrove.olemiss.edu/aicpa\\_guides/](https://egrove.olemiss.edu/aicpa_guides/).
- [39] S. Tian and Y. Yu, “Financial ratios and bankruptcy predictions: an international evidence,” *International Review of Economics & Finance*, vol. 51, pp. 510–526, 2017.
- [40] S. Kim, B. M. Mun, and S. J. Bae, “Data depth based support vector machines for predicting corporate bankruptcy,” *Applied Intelligence*, vol. 48, no. 3, pp. 791–804, 2018.
- [41] S. Daliri, “Using harmony search algorithm in neural networks to improve fraud detection in banking system,” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 6503459, 5 pages, 2020.
- [42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013*, pp. 1–12, Scottsdale, AZ, USA, May 2013.
- [43] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [44] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, San Diego, CA, USA, 2016.
- [45] S. A. Shahee and U. Ananthakumar, “An effective distance based feature selection approach for imbalanced data,” *Applied Intelligence*, vol. 50, no. 3, pp. 717–745, 2020.

## Research Article

# A Hierarchical View Pooling Network for Multichannel Surface Electromyography-Based Gesture Recognition

Wentao Wei <sup>1</sup>, Hong Hong <sup>2</sup>, and Xiaoli Wu<sup>1</sup>

<sup>1</sup>School of Design Arts and Media, Nanjing University of Science and Technology, Nanjing, Jiangsu, China

<sup>2</sup>School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu, China

Correspondence should be addressed to Wentao Wei; [weiwentao@njust.edu.cn](mailto:weiwentao@njust.edu.cn)

Received 10 June 2021; Accepted 29 June 2021; Published 26 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Wentao Wei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hand gesture recognition based on surface electromyography (sEMG) plays an important role in the field of biomedical and rehabilitation engineering. Recently, there is a remarkable progress in gesture recognition using high-density surface electromyography (HD-sEMG) recorded by sensor arrays. On the other hand, robust gesture recognition using multichannel sEMG recorded by sparsely placed sensors remains a major challenge. In the context of multiview deep learning, this paper presents a hierarchical view pooling network (HVPN) framework, which improves multichannel sEMG-based gesture recognition by learning not only view-specific deep features but also view-shared deep features from hierarchically pooled multiview feature spaces. Extensive intrasubject and intersubject evaluations were conducted on the large-scale noninvasive adaptive prosthetics (NinaPro) database to comprehensively evaluate our proposed HVPN framework. Results showed that when using 200 ms sliding windows to segment data, the proposed HVPN framework could achieve the intrasubject gesture recognition accuracy of 88.4%, 85.8%, 68.2%, 72.9%, and 90.3% and the intersubject gesture recognition accuracy of 84.9%, 82.0%, 65.6%, 70.2%, and 88.9% on the first five subdatabases of NinaPro, respectively, which outperformed the state-of-the-art methods.

## 1. Introduction

As a noninvasive approach of establishing links between muscles and devices, the surface electromyography- (sEMG-) based neural interface, also known as the muscle computer interface (MCI), has been widely studied in the past decade. Surface electromyography is a type of biomedical signal recorded by noninvasive electrodes placed on human skin [1]; it is the spatiotemporal superposition of motor unit action potential (MUAP) generated by all active motor units (MU) at different depths within the recording area [2]. sEMG recorded from subject's forearm measures muscular activity of his/her hand movements, thus, can be used for hand gesture recognition. So far, the sEMG-based gesture recognition techniques have been widely applied in rehabilitation engineering [3–5] and human-computer interaction [6–8].

From the perspective of signal recording, there are two types of sEMG signals: (1) high-density sEMG (HD-sEMG)

[9–11] signals which are recorded by electrode arrays that consist of dozens, or even hundreds of electrodes arranged in a grid; (2) multichannel sEMG signals [12, 13] which are recorded by several sparsely located electrodes. For MCIs such as robotic hand prostheses and upper-limb rehabilitation robots, one of the key challenges is to precisely recognize the user's gestures through sEMG signals collected from his/her forearm. Over the past five years, feature learning approaches based on convolutional neural networks (CNNs) have shown promising success in HD-sEMG-based gesture recognition, that is, achieving >90% recognition accuracy in classifying a large set of gestures [11], and almost 100% recognition accuracy in classifying a small set of gestures [14, 15], because HD-sEMG signals contain both spatial and temporal information of muscle activity [16]. Compared to conventional feature engineering approaches based on shallow learning models, a major advantage of feature learning approaches is that the end-to-end learning capability of deep learning models enables them to

automatically learn representative deep features from raw sEMG signals without any hand-crafted feature [17].

On the other hand, achieving high accuracy in multichannel sEMG-based gesture recognition performance remains a challenging task, because multichannel sEMG is noisy, random, nonstationary [18], and vulnerable to electrode shift [16] and contains much less spatial information about muscle activities than HD-sEMG [19]. So far, researchers have tried a variety of strategies to improve the multichannel sEMG-based gesture recognition performance, including extracting more representative features [20], using multimodal gesture data collected from multiple sensors [21], and developing more sophisticated deep learning models [15].

In recent years, there has emerged a trend in combining deep learning models with feature engineering techniques, as well-designed time domain (TD) [22], frequency domain (FD) [23], and time-frequency domain (TFD) [24] features have achieved remarkable success in multichannel sEMG-based gesture recognition systems. For example, Zhai et al. [25] calculated spectrograms of sEMG and used them as features for CNN-based gesture recognition and achieved 78.7% gesture recognition accuracy for recognizing 49 gestures. Hu et al. [26] extracted the Phinyomark feature set [23] from raw sEMG signals and fed them into an attention-based hybrid convolutional neural network and recurrent neural network (CNN-RNN) architecture for gesture recognition; they achieved 87% recognition accuracy for recognizing 52 gestures. Betthausen et al. [27] proposed the encoder-decoder temporal convolutional networks (ED-TCN) for sEMG-based sequential movement prediction; the inputs of their proposed ED-TCN model were composed of mean absolute value (MAV) sequences. Chen et al. [28] used continuous wavelet transform (CWT) to process the data as the input of their proposed CNN model.

In machine learning, multiview learning refers to learning from data described by different view-points or different feature sets [29, 30]. On this basis, Wei et al. [31] proposed a multiview CNN (MV-CNN) framework that constructs images generated from different sEMG features into multiview representations of multichannel sEMG. Compared to prior works that combined deep learning models with feature engineering techniques, one of the key characteristics of MV-CNN is that it adopts a “divide-and-aggregation” strategy that is able to independently learn deep features from each individual view of multichannel sEMG. The MV-CNN framework showed promising success in multichannel sEMG-based gesture recognition, as the gesture recognition accuracy achieved by MV-CNN significantly outperformed the state-of-the-art deep learning approaches.

From the perspective of multiview learning, there are generally two types of features, namely, the “view-specific feature” or “private feature” particular for each individual view and the “view-shared feature” or “public feature” shared by all views [32]. The independent learning under each individual view is able to learn view-specific features [33]; on the other hand, it is unable to learn shared information across different views [34]. The MV-CNN

framework [31] did consider view-shared learning by an early fusion strategy that concatenates the output from the lowest convolutional layers of all view-specific CNN branches. However, from our perspective, the early fusion strategy used in MV-CNN is still a naive approach based on concatenation; it also ignores the original input feature spaces of different views.

Aiming at improving multichannel sEMG-based gesture recognition via better learning of view-shared deep features, in this paper, we proposed a hierarchical view pooling network (HVPN) framework, in which view-shared feature spaces were hierarchically pooled from multiview low-level features for view-shared learning. In order to build up more discriminative view-shared feature spaces, we proposed a CNN-based view pooling technique named the feature-level view pooling (FLVP) layer, which is able to learn a unified view-shared feature space from multiview low-level features. Compared to MV-CNN [31], the application of hierarchical view pooling and FLVP layer results in a wider (i.e., with more CNN branches) and deeper (i.e., with more convolutional layers in the view-shared learning branches) network architecture, respectively, thus enabling the learning of more representative view-shared deep features.

The remainder of this paper is organized as follows. Section 2 formulates the multiview learning problem, describes the databases, and details the proposed HVPN framework. Section 3 introduces the experiments in this paper and provides the experimental setup. Section 4 presents and discusses the experimental results. Finally, Section 5 concludes the paper.

## 2. Materials and Methods

*2.1. Problem Statement.* According to Wei et al. [31], the problem of multiview deep learning-based gesture recognition using multichannel sEMG signals can be formulated as

$$y = H(v_1, v_2, \dots, v_n; \theta), \quad (1)$$

where  $v_1, v_2, \dots, v_n$  denote multiview representations from  $n$  different views of  $C$ -channel sEMG signals  $x \in \mathbb{R}^C$ ,  $H$  denotes a deep neural network with parameters  $\theta$ , and  $y$  denotes the final gesture classification results.

The relationship between  $v_1, v_2, \dots, v_n$  and  $x$  can be formulated as

$$v_i = f_{v_i}(x), \quad (2)$$

where  $f_{v_i}$ ,  $i = 1, 2, \dots, n$  denotes view construction functions that generate multiview representations from raw sEMG signals.

In the field of multiview deep learning, a common approach is to build up  $n$  neural networks  $H_i$ ,  $i = 1, 2, \dots, n$  to learn deep representations from  $n$  views, respectively, and then use a view aggregation network  $H_a$  to fuse the learned multiview deep representations together and obtain the final decisions  $y$ . Thus, equation (1) can be written as

$$y = H_a(H_{l_1}(v_1; \theta_{l_1}), H_{l_2}(v_2; \theta_{l_2}), \dots, H_{l_n}(v_n; \theta_{l_n}); \theta_a). \quad (3)$$

**2.2. Databases.** The evaluations in this work were performed offline using multichannel sEMG signals from the publicly available NinaPro databases [35]. We chose 5 subdatabases of NinaPro, which contain multichannel sEMG signals recorded from intact and transradial amputees through different types of electrodes. Details of these databases are as follows:

The first subdatabase (denoted as NinaProDB1) contains sEMG signals collected from 27 intact subjects; each subject was asked to perform 53 gestures, including 12 finger movements (denoted as Exercise A), 17 wrist movements and hand postures (denoted as Exercise B), 23 grasping and functional movement (denoted as Exercise C), and the rest movement; each gesture was repeated 10 times (i.e., 10 trials per gesture). The sEMG signals in NinaProDB1 were recorded by 10 Otto Bock 13E200-50 electrodes at a sampling rate of 100 Hz [13]. As most of the existing studies on this database excluded the rest movement for gesture recognition [10, 26, 31, 36], in our experiments we also excluded the rest movement for the convenience of performance comparison.

The second subdatabase (denoted as NinaProDB2) contains sEMG signals collected from 40 intact subjects; each subject was asked to perform 50 gestures, including Exercises B and C in NinaProDB1, 9 force patterns (denoted as Exercise D), and the rest movement; each gesture was repeated 6 times (i.e., 6 trials per gesture). The sEMG signals in NinaProDB2 were recorded by 12 Delsys Trigno Wireless electrodes at a sampling rate of 2000 Hz [13].

The third subdatabase (denoted as NinaProDB3) contains sEMG signals collected from 11 transradial amputees; each subject was asked to perform exactly the same 50 gestures as those in NinaProDB2; each gesture was repeated 6 times (i.e., 6 trials per gesture). The sEMG signals in NinaProDB3 were recorded by 12 Delsys Trigno Wireless electrodes at a sampling rate of 2000 Hz [13]. According to the authors of NinaPro database, during the sEMG recording process of NinaProDB3, three amputated subjects performed only a part of gestures due to fatigue or pain, and in two amputated subjects, the number of electrodes was reduced to ten due to insufficient space [13]. To ensure training and testing of the model can be completed, we omitted data from these subjects following the experimental configuration used by Wei et al. [31].

The fourth subdatabase (denoted as NinaProDB4) contains sEMG signals collected from 10 intact subjects; each subject was asked to perform exactly the same 53 gestures as those in NinaProDB1; each gesture was repeated 6 times (i.e., 6 trials per gesture). The sEMG signals in NinaProDB4 were recorded by the Cometa Wave Plus Wireless sEMG system with 12 electrodes, and the sampling rate was 2000 Hz [37]. After checking the data, we found that two subjects (i.e., subject 4 and subject 6) did not complete all hand movements; their data were omitted in our experiments.

The fifth subdatabase (denoted as NinaProDB5) contains sEMG signals collected from 10 intact subjects; each subject was asked to perform exactly the same 53 gestures as those in NinaProDB1; each gesture was repeated 6 times (i.e., 6 trials per gesture). Following the experimental configuration in [37], we chose 41 gestures (i.e., Exercise B and C plus rest movement) from all 53 gestures in NinaProDB5 for classification. The sEMG signals in NinaProDB5 were recorded by two Thalmic Myo armbands at a sampling rate of 200 Hz; each Myo armband contains 8 sEMG electrodes [37].

**2.3. Data Preprocessing and View Construction.** Due to memory limitation of the hardware, for experiments on NinaProDB2-DB4, we downsampled the sEMG signals from 2000 Hz to 100 Hz following the experimental configuration used in [31].

In multiview learning, view construction is usually defined as generation of multiple views from a single view of original data [38]. Considering the fairness of performance comparison, the view construction process in this paper was exactly the same as that in MV-CNN framework [31]. As a result, three different views of multichannel sEMG, denoted as  $v_1$ ,  $v_2$ , and  $v_3$ , are represented by images of discrete wavelet packet transform coefficients (DWPTC), discrete wavelet transform coefficients (DWTC), and the first Phinyomark's feature set (Phin\_FS1) that are extracted from raw sEMG signals, respectively.

For the generation of the feature images, we followed the image generation algorithm proposed by Jiang and Yin [39], which is described in Algorithm 1.

Although the abovementioned three views of multichannel sEMG were proven to be the most discriminative views for gesture recognition in [31], the construction of them still requires a lot of computational time and resources, as well as their high-dimensionality results in the increase of the number of neural network parameters, making us consider the trade-off between gesture recognition accuracy and computational complexity. Thus, in this paper, we also evaluated a "two-view" configuration, which selected the two most discriminative views (i.e.,  $v_1$  and  $v_2$ , represented by images of DWPTC and DWTC, resp.) out of these three views of multichannel sEMG and used them as the input of the proposed HVPN framework. Details of the evaluations on the "two-view" configuration will be presented in the following sections of this paper.

For extraction of sEMG features during view construction, sliding windows were used to segment the multichannel sEMG. Early studies in MCI have pointed out that the response time of a real-time MCI system should be kept below 300 ms to avoid a time delay perceived by the user [40, 41]. For this reason, the sliding window length was set to 200 ms for most of the experiments, and the window increment was set to 10 ms except for experiments on NinaProDB5 using the sliding window length of 200 ms. For experiments on NinaProDB5 using 200 ms sliding windows, we followed the experimental configuration used by Pizzolato et al. [37] and Wei et al. [31], which set the window increment to 100 ms.

Suppose the images that represent the  $i$ th view have an sEMG feature dimension of  $M_i$  and an sEMG channel

**Input:** sEMG features  $z \in \mathbb{R}^{D \times C}$ , which are extracted from a sliding window that is used to segment C-channel sEMG signals.

**Output:** The generated image, denoted as  $v \in \mathbb{R}^{M \times C}$

```

(1) if  $D \% 2 == 0$  then
(2)    $D = D + 1$ ;
(3) end if
(4)  $seq = ['1']; index = [1]$ ;
(5)  $i = 1; j = i + 1$ ;
(6) while  $i \neq j$  do
(7)    $l = "ij"; r = "ji"$ ;
(8)   if  $j > D$  then
(9)      $j = 1$ ;
(10)  else if  $l \notin seq \ \&\& \ r \notin seq$  then
(11)     $seq.append('j')$ ;
(12)     $index.append(j)$ ;
(13)     $i = j; j = i + 1$ ;
(14)  else
(15)     $j = j + 1$ ;
(16)  end if
(17) end while
(18)  $index = index[: -1]$ ;
(19)  $v =$ ;
(20) for  $k = 1; k \leq length(index)$  do
(21)    $v.append(z[:, index[k]])$ 
(22) end for

```

ALGORITHM 1: The image generation algorithm used in this paper [39].

dimension of  $C$ , the  $M_i \times C$  (width, height, respectively, depth = 1) feature space of  $v_i$  is firstly transformed into an  $M_i \times C \times 1$  (depth, width, and height, respectively) feature space before it is input into neural network architecture of HVPN for gesture recognition. The transformation is based on the experimental results presented in [15], where the  $20 \times 10 \times 1$  (depth, width, and height, respectively) sEMG images significantly outperformed the  $1 \times 20 \times 10$  (depth, width, and height, respectively) sEMG images as the input of an end-to-end CNN in gesture recognition using 10-channel sEMG signals segmented by 20-frame sliding window.

**2.4. The HVPN Framework.** A diagram of our proposed HVPN framework with all three views of multichannel sEMG is illustrated in Figure 1. The deep learning architecture of HVPN can be divided into three parts: view-specific CNNs, hierarchical view pooling CNNs, and a view aggregation network. For HVPN with the “two-view” configuration, there are two view-specific CNN branches to learn view-specific deep features from  $v_1$  and  $v_2$ , respectively, and other parts are almost the same as those illustrated in Figure 1. The following sections describe the detailed network architecture and hyperparameter configurations of these parts.

**2.5. View-Specific CNNs.** After view construction, we built up three view-specific CNN branches to learn view-specific deep features from  $v_1$ ,  $v_2$ , and  $v_3$ , respectively. As shown in Figure 1, all view-specific CNN branches share the same network architecture but do not share their weights. The network architecture of each view-specific CNN branch is

based on GengNet [10], which has been extensively used in sEMG-based gesture recognition [15, 31, 42]. Specifically, the images of each view are input into two convolutional layer with  $64 \ 3 \times 3$  filters (stride = 1), followed by two locally connected (LC) layers with  $64 \ 1 \times 1$  filters (stride = 1) and one fully connected (FC) layer with 1024 hidden units. For each CNN branch, we applied batch normalization and the ReLU nonlinearity function after each layer and added dropout layers to the FC layer and the last LC layer to prevent overfitting. The input of each CNN is also normalized through batch normalization.

**2.6. Hierarchical View Pooling CNNs.** The hierarchical view pooling CNNs are composed of two CNN branches, namely, the first-level view pooling CNN (denoted as L1-VPCNN) and the second-level view pooling CNN (denoted as L2-VPCNN); each of them starts with an FLVP layer, which is used to learn a view-shared feature space from multiview low-level features. As illustrated in Figure 2, the FLVP layer firstly concatenates the input feature maps from different views together and then learns a unified feature space from the concatenated feature maps through a  $1 \times 1$  convolutional layer with 64 filters. The FLVP layers in our proposed HVPN framework play two important roles: (1) each of them learns a unified feature space shared by all views from concatenated multiview low-level features for view-shared learning; (2) compared with the extensively used view pooling technique based on simple element-wise maximum [43] or average [44] operation, each FLVP layer can guarantee that its corresponding hierarchical view pooling CNN branch is deep enough to learn representative features.



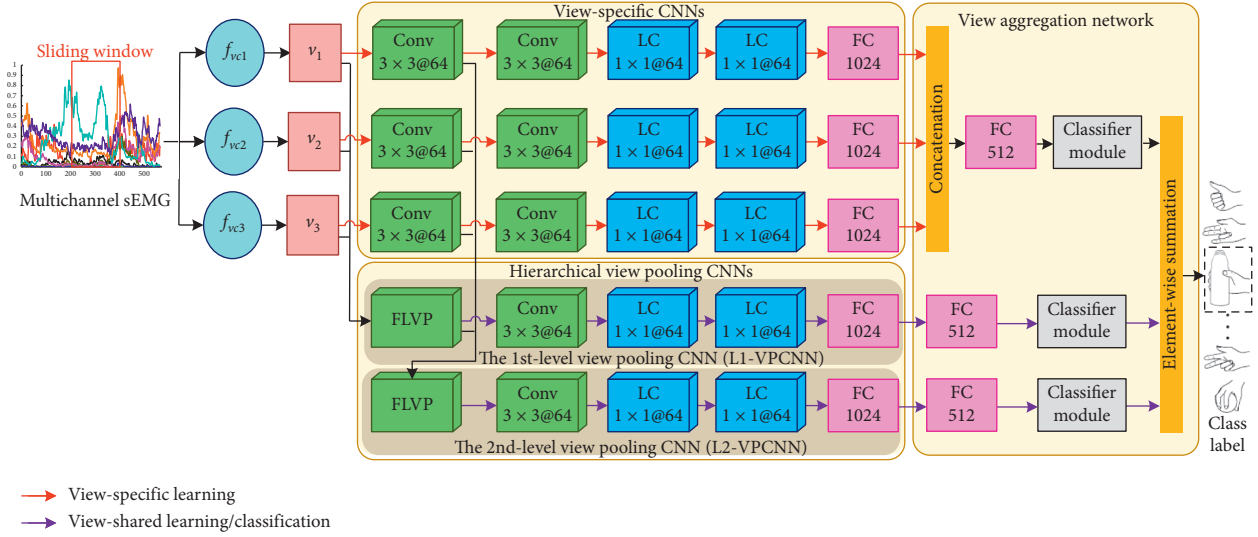


FIGURE 1: A schematic diagram of the proposed HVPN framework. FLVP, Conv, LC, and FC denote the feature-level view pooling layer, convolutional layer, locally connected layer, and fully connected layer, respectively. The numbers after the layer name denote the size and number of the filters or neurons; for example, Conv  $3 \times 3@64$  denotes a CNN with 64  $3 \times 3$  filters, and FC 1024 denotes an FC layer with 1024 hidden units.

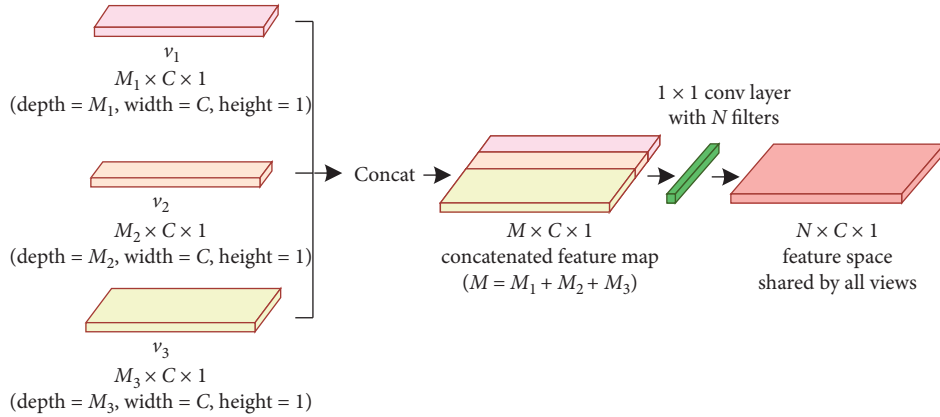


FIGURE 2: Diagram of the FLVP layer.

Suppose we have  $v_1 \in \mathbb{R}^{M_1 \times C \times 1}$ ,  $v_2 \in \mathbb{R}^{M_2 \times C \times 1}$ ,  $v_3 \in \mathbb{R}^{M_3 \times C \times 1}$ , and the multiview low-level features learned by the bottom convolutional layers of three view-specific CNN branches are  $\hat{v}_1, \hat{v}_2, \hat{v}_3 \in \mathbb{R}^{64 \times C \times 1}$ , respectively. The hierarchical view pooling process by FLVP layers can be formulated as follows.

The 1st-level view pooling:

$$\begin{aligned}
 v_{c_1} &= v_1 \parallel v_2 \parallel v_3, \\
 \hat{v}_{l_1} &= H_{f_{v_1}}(v_{c_1}; \theta_{f_{v_1}}), \\
 v_{c_1} &\in \mathbb{R}^{M \times C \times 1}, M = M_1 + M_2 + M_3, \\
 \hat{v}_{l_1} &\in \mathbb{R}^{64 \times C \times 1}.
 \end{aligned} \tag{4}$$

The 2nd-level view pooling:

$$\begin{aligned}
 v_{c_2} &= \hat{v}_1 \parallel \hat{v}_2 \parallel \hat{v}_3 \parallel \hat{v}_{l_1}, \\
 \hat{v}_{l_2} &= H_{f_{v_2}}(v_{c_2}; \theta_{f_{v_2}}), \\
 \hat{v}_{c_2} &\in \mathbb{R}^{256 \times C \times 1}, \\
 \hat{v}_{l_2} &\in \mathbb{R}^{64 \times C \times 1},
 \end{aligned} \tag{5}$$

where  $\parallel$  denotes the feature-level concatenation operation,  $\hat{v}_{l_i}$  denotes the learned feature space after level- $i$  view pooling,  $H_{f_{v_i}}$  denotes the FLVP layer in  $L_i$ -VPCNN, and  $\theta_{f_{v_i}}$  denotes its parameters.

The remaining parts of L1-VPCNN and L2-VPCNN perform view-shared learning from  $\hat{v}_{l_1}$  and  $\hat{v}_{l_2}$ , respectively. They share the same network architecture, which is composed of one convolutional layer with  $64 \times 3 \times 3$  filters (stride = 1), followed by two LC layers with  $64 \times 1 \times 1$  filters (stride = 1) and one FC layer with 1024 hidden units.

**2.7. View Aggregation Network.** The view aggregation network is used for the following: (1) the fusion of all view-specific CNN branches and hierarchical view pooling CNN branches and (2) final gesture classification. As shown in Figure 1, the view aggregation network adopts a two-step view aggregation strategy. Specifically, it concatenates the output view-specific deep features learned by three view-specific CNN branches together at first. Then, the concatenated view-specific deep features and the view-shared deep features learned by L1-VPCNN and L2-VPCNN are input into three branches, respectively. Each branch consists of one FC layer with 512 hidden units and a classifier module, and each classifier module is composed of a G-way FC layer and a softmax classifier for gesture classification. At the top of HVPN, there is an element-wise summation operation that sums up the softmax scores predicted by all three classifier modules together to form the final classification results.

**2.8. Evaluation Metric and Methodology.** For experiments in this study, we calculated the gesture recognition accuracy for each subject as the evaluation metric, which is defined as

$$\text{accuracy} = \frac{\text{number of correct classifications}}{\text{Total number of classifications}} * 100\%. \quad (6)$$

The evaluation methodology in this paper can be categorized into intrasubject evaluation and intersubject evaluation. Generally speaking, in intrasubject evaluation, the deep learning model is trained on a part of the data from one subject and tested on the nonoverlapping part of the data from the same subject, whereas in intersubject evaluation, the deep learning model is usually trained on data from one or a group of subjects and tested on data from another group of subjects.

For fair performance comparison, we adopted the same intrasubject and intersubject evaluation schemes as those were most commonly used in existing studies on NinaPro database [10, 13, 26, 31, 36, 42], which are described as follows.

**Intrasubject Evaluation.** For intrasubject evaluation, we followed the evaluation scheme proposed by the NinaPro team [13]. Specifically, for each subject, approximately 2/3 of the gesture trials are used as the training set; the remaining gesture trials constitute the test set. The final gesture recognition accuracy is obtained by averaging the achieved accuracy over all subjects. The selection of gesture trials for training and testing are based on the literature [13, 37].

**Intersubject Evaluation.** For intersubject evaluation, we followed the leave-one-subject-out cross-validation (LOSOCV) scheme used in the literature [31, 36, 42]. Specifically, in each fold of the cross-validation, data from one subject is used as the test set, and data from the remaining subjects is used as the training set. The final gesture recognition accuracy of the evaluation is obtained by averaging the achieved accuracy over all folds.

Specifications of the evaluation methodology on different sEMG databases are presented in Table 1.

**2.9. Deep Domain Adaptation for Intersubject Evaluation.** In intersubject evaluation, the training (i.e., source domain) and test (i.e., target domain) data comes from two non-overlapping groups of subjects; thus, there exist distribution mismatch and domain shift across the source target domain caused by electrode shifts, changes in arm position, muscle fatigue, skin condition [45], and individual differences among subjects [46], which may dramatically degrade the classification performance of the model [47].

To reduce the negative effect of distribution mismatch and domain shift on classification performance, a number of existing deep learning based approaches [31, 42, 48] in this field have applied a novel unsupervised deep domain adaptation technique named multistream AdaBN (MS-AdaBN) [42]. The MS-AdaBN technique uses a multistream network to incrementally update the batch normalization statistics of the network training process with the calibration data.

In this work, the MS-AdaBN was also implemented for deep domain adaptation in LOSOCV, because our preliminary experiments on NinaProDB1 revealed that the LOSOCV accuracy achieved by our proposed model without deep domain adaptation is far from practical applications (i.e., < 30%). Similar results were achieved by MV-CNN and reported by Wei et al. [31].

For selection of training, calibration, and test data, we followed exactly the same MS-AdaBN configuration as that used in previous works [31, 42]. It should be mentioned that as MS-AdaBN requires a relatively large amount of calibration data, it may not be the best solution for domain adaptation in the context of multichannel sEMG-based gesture recognition. Nevertheless, MS-AdaBN is not a contribution of this work, and we used it in our experiments because we wanted to ensure a fair comparison of LOSOCV accuracy between our proposed method and the previously proposed MV-CNN [31], which is a multiview deep learning framework that also adopted MS-AdaBN for domain adaptation.

### 3. Experiments

All experiments were performed offline (i.e., not real-time) on a DevMax401 workstation with NVIDIA GeForce GTX1080Ti GPU. The proposed HVPN framework was trained using the stochastic gradient descent (SGD) optimizer with 28 epochs. For all experiments, the batch size was set to 1000, and a learning rate decay strategy was adopted during training to improve convergence, which initialized the learning rate at 0.1 and divided it by 10 after 16 and 24 epochs. For all layers with dropout, the dropout rate was set to 0.65 during training.

**3.1. Evaluation of the Hierarchical View Pooling Strategy.** Evaluation of the hierarchical view pooling strategy can be divided into two steps. First, we carried an ablation study to verify the effectiveness of FLVP layer. Second, we carried out an ablation study to validate the effectiveness of the proposed hierarchical view pooling CNNs. For all experiments

TABLE 1: Specifications of the evaluation methodology on different sEMG databases.

Databases	Intrasubject		Intersubject
	Trials for training	Trials for testing	
NinaPro DB1	1st, 3rd, 4th, 6th, 7th, 8th, 9th	2nd, 5th, 10th	LOSOCV
NinaPro DB2	1st, 3rd, 4th, 6th	2nd, 5th	LOSOCV
NinaPro DB3	1st, 3rd, 4th, 6th	2nd, 5th	LOSOCV
NinaPro DB4	1st, 3rd, 4th, 6th	2nd, 5th	LOSOCV
NinaPro DB5	1st, 3rd, 4th, 6th	2nd, 5th	LOSOCV

in these ablation studies, the sliding window length was set to 200 ms.

In the first step of the evaluation, the standard HVPN was firstly compared with its two variants, namely, HVPN-maxpool and HVPN-avgpool, on five databases (i.e., NinaProDB1-DB5). In HVPN-maxpool, the FLVP layer in L2-VPCNN was replaced by view pooling based on element-wise maximum, while in HVPN-avgpool the FLVP layer in L2-VPCNN was replaced by view pooling based on element-wise average. Meanwhile, the FLVP layers in the L1-VPCNN of HVPN-maxpool and HVPN-avgpool were retained, because the input feature spaces of L1-VPCNN have different sizes, which make it impossible for performing element-wise maximum or average operation among them.

In the second step of the evaluation, the proposed HVPN was compared with the following deep neural network architectures:

**VS-L1VP:** a deep network that is equivalent to HVPN without the L2-VPCNN.

**VS-L2VP:** a deep network that is equivalent to HVPN without the L1-VPCNN.

**VS-ONLY:** a deep network that only consists of view-specific CNNs, followed by a concatenation operation that fuses their output together, a FC layer with 512 hidden units and a classifier module.

The schematic illustration of VS-L1VP, VS-L2VP, and VS-ONLY is depicted in Figure 3. Compared to HVPN that contains hierarchical view pooling CNNs, there is only one view pooling CNN in VS-L1VP, as well as VS-L2VP, for view-shared learning.

**3.2. Comparison with Related Works.** The gesture recognition accuracy achieved by the proposed HVPN framework, as well as the gesture recognition accuracy achieved by the proposed HVPN framework with the “two-view” configuration (denoted as HVPN-2-view), was further compared with related works on five databases (i.e., NinaProDB1-DB5). For the aim of fairness in this comparison, among various machine learning methods that were proposed for sEMG-based gesture recognition and tested on NinaPro, we only considered the ones that meet the following requirements: (1) their reported gesture recognition accuracy was achieved using exactly the same intrasubject or intersubject gesture recognition schemes as described in Section 2; (2) the input of their machine learning models were engineered features, not raw sEMG signals.

To prevent overfitting, a pretraining strategy that has been widely used by the compared methods [26, 31] was also adopted in this work. Specifically, for each experiment, a pretrained model was firstly trained using all available training data; then, the gesture recognition model for each subject was initialized by the pretrained model. For all layers with dropout, the dropout rate was set to 0.5 during the pretraining stage.

For comparison of intrasubject gesture recognition accuracy, we evaluated the gesture recognition accuracy achieved with 50 ms, 100 ms, 150 ms, and 200 ms sliding windows. Moreover, the gesture recognition accuracy obtained by majority voting on all 200 ms windows within each trial is also presented in the column labeled “Trial.” For comparison of LOSOCV (i.e., intersubject gesture recognition) accuracy, we only evaluated the gesture recognition accuracy achieved with 200 ms sliding windows.

## 4. Results and Discussion

**4.1. Multichannel sEMG-Based Gesture Recognition Enhanced by Hierarchical View Pooling.** Table 2 presents the intrasubject and LOSOCV accuracy achieved by the standard HVPN, HVPN-maxpool, and HVPN-avgpool on five databases. The proposed HVPN framework achieved the intrasubject gesture recognition accuracy of 86.8%, 84.4%, 68.2%, 70.8%, and 88.6% on NinaProDB1, DB2, DB3, DB4, and DB5, respectively, and achieved the LOSOCV accuracy of 83.1%, 79.0%, 65.6%, 67.0%, and 87.1% on NinaProDB1, DB2, DB3, DB4, and DB5, respectively. The gesture recognition accuracy achieved by HVPN was higher than that achieved by HVPN-maxpool and HVPN-avgpool in all experiments, indicating that the FLVP layer can achieve better gesture recognition accuracy than the conventional view pooling approaches based on element-wise maximum or average operation. However, when evaluated on NinaProDB1, DB2, DB3, and DB4, the performance improvement brought by the FLVP layer was subtle (i.e., from +0.2% to +0.4% over element-wise max or average pooling). This is likely due to the fact that in HVPN-maxpool and HVPN-avgpool we only replaced the FLVP layer in L2-VPCNN with conventional view pooling, making them very similar to the original HVPN.

Table 3 presents the intrasubject and LOSOCV accuracy achieved by HVPN, VS-L1VP, VS-L2VP, and VS-ONLY on five databases (i.e., NinaProDB1-DB5). According to the experimental results in Table 3, the deep neural network architectures with view pooling CNNs (i.e., HVPN,

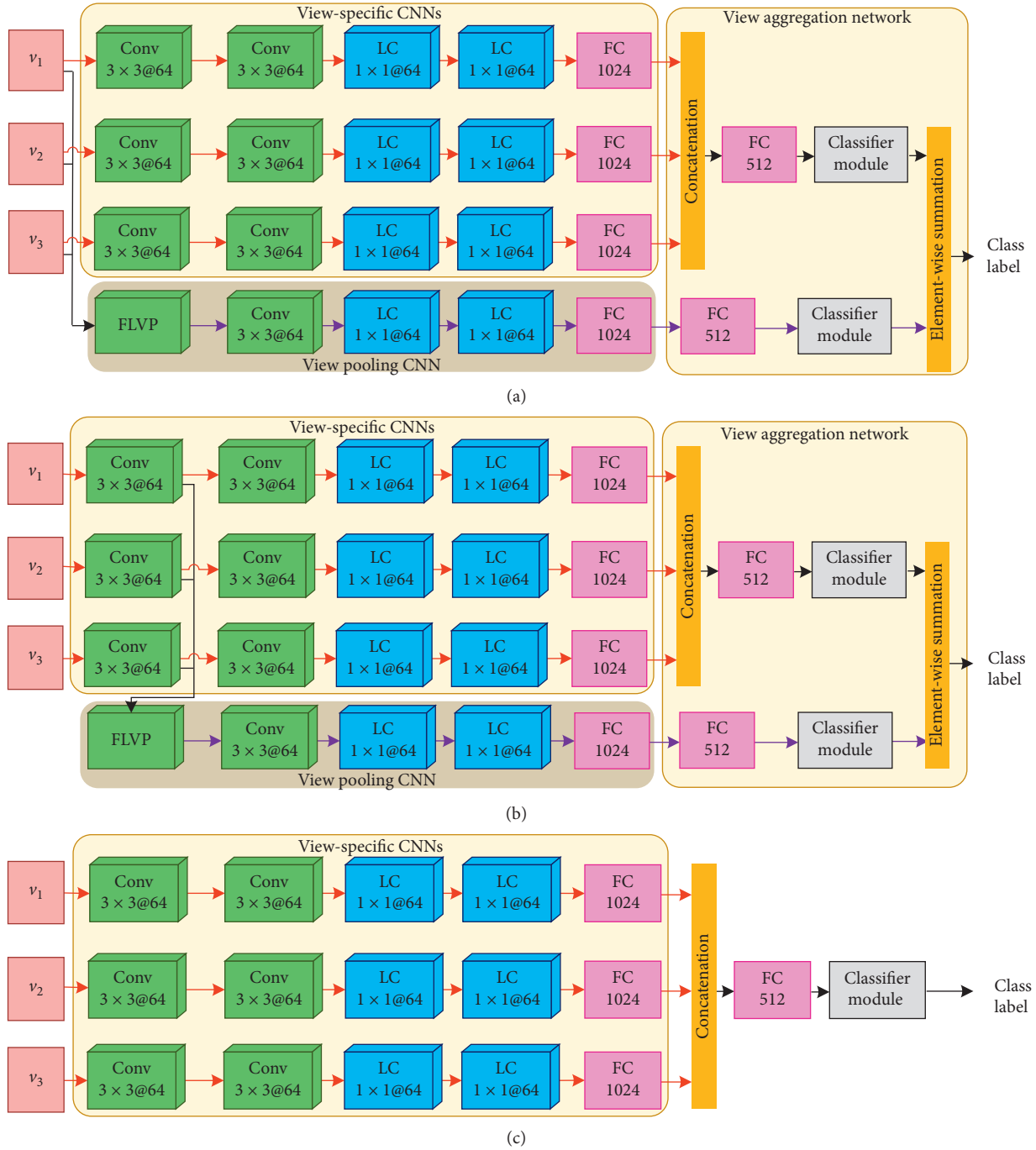


FIGURE 3: Schematic diagrams of (a) VS-L1VP, (b) VS-L2VP, and (c) VS-ONLY.

TABLE 2: Gesture recognition accuracy achieved by the standard HVPN, HVPN-maxpool, and HVPN-avgpool on five databases.

Database	Evaluation methodology	HVPN	HVPN-maxpool	HVPN-avgpool
NinaProDB1	Intrasubject	<b>86.8%</b>	86.4%	86.5%
NinaProDB2	Intrasubject	<b>84.4%</b>	84.1%	84.1%
NinaProDB3	Intrasubject	<b>68.2%</b>	68.0%	67.9%
NinaProDB4	Intrasubject	<b>70.8%</b>	70.5%	70.5%
NinaProDB5	Intrasubject	<b>88.6%</b>	88.1%	88.1%
NinaProDB1	LOSOCV	<b>83.1%</b>	82.7%	82.8%
NinaProDB2	LOSOCV	<b>79.0%</b>	78.8%	78.7%
NinaProDB3	LOSOCV	<b>65.6%</b>	65.4%	65.3%
NinaProDB4	LOSOCV	<b>67.0%</b>	66.6%	66.6%
NinaProDB5	LOSOCV	<b>87.1%</b>	86.4%	86.6%

Results in bold entries indicate best performance.

TABLE 3: Gesture recognition accuracy achieved by HVPN, VS-L1VP, VS-L2VP, and VS-ONLY on five databases.

Database	Evaluation methodology	HVPN	VS-L1VP	VS-L2VP	VS-ONLY
NinaProDB1	Intrasubject	<b>86.8%</b>	86.5%	86.2%	85.8%
NinaProDB2	Intrasubject	<b>84.4%</b>	84.1%	83.9%	83.4%
NinaProDB3	Intrasubject	<b>68.2%</b>	67.7%	67.5%	67.2%
NinaProDB4	Intrasubject	<b>70.8%</b>	69.9%	69.7%	68.5%
NinaProDB5	Intrasubject	<b>88.6%</b>	87.9%	88.3%	87.2%
NinaProDB1	LOSOCV	<b>83.1%</b>	82.6%	82.5%	81.9%
NinaProDB2	LOSOCV	<b>79.0%</b>	78.7%	78.7%	78.1%
NinaProDB3	LOSOCV	<b>65.6%</b>	65.5%	65.0%	64.7%
NinaProDB4	LOSOCV	<b>67.0%</b>	66.3%	65.7%	65.2%
NinaProDB5	LOSOCV	<b>87.1%</b>	86.2%	86.5%	84.7%

Results in bold entries indicate best performance.

VS-L1VP, and VS-L2VP) significantly outperformed VS-ONLY, indicating that combining view-specific learning with view-shared learning is better than performing view-specific learning alone in the context of multiview deep learning for multichannel sEMG-based gesture recognition. Moreover, the intrasubject and LOSOCV accuracy achieved by HVPN was higher than that achieved by VS-L1VP and VS-L2VP on all databases, which proves the effectiveness of our proposed hierarchical view pooling strategy in improving gesture recognition accuracy.

*4.2. Comparison with Related Works Based on Intrasubject Evaluation.* Table 4 presents the intrasubject gesture recognition accuracy achieved by various methods on the first five subdatabases of NinaPro. Among these methods, the methods proposed in [13, 36, 37] are shallow learning frameworks, the methods proposed in [25–27, 49, 50] are single-view deep learning frameworks, and the method proposed in [31] is a multiview deep learning framework (i.e., MV-CNN). All the above-mentioned methods are non-end-to-end methods using engineered sEMG features as their input, and they used exactly the same intrasubject evaluation scheme as that was used in our work.

Experimental results in Table 4 demonstrate that when using all three views of multichannel sEMG as input, the proposed HVPN achieved the intrasubject gesture recognition accuracy of 88.4%, 85.8%, 68.2%, 72.9%, and 90.3% on NinaProDB1, DB2, DB3, DB4, and DB5, respectively, with the sliding window length of 200 ms, which outperformed not only shallow learning frameworks [13, 36, 37] but also deep learning frameworks [25, 26, 31, 49, 50] that were proposed for sEMG-based gesture recognition in recent years.

Compared to MV-CNN, which is also a multiview deep learning framework, experimental results show the following: (1) when using exactly the same input, the gesture accuracy achieved by MV-CNN was significantly inferior to that achieved by HVPN on all databases; (2) when the number of input views of HVPN was reduced to two (i.e., denoted as HVPN-2-view in Table 4), it still outperformed MV-CNN framework on most of the databases (i.e., NinaPro DB2, DB3, DB4, and DB5), and their gesture recognition accuracy on NinaProDB1 was almost the same. For example,

when the sliding window length was set to 200 ms, the HVPN-2-view achieved the intrasubject gesture recognition accuracy of 88.1%, 85.0%, 67.9%, 72.1%, and 90.1% on NinaPro DB1, DB2, DB3, DB4, and DB5, respectively. By comparison, the intrasubject gesture recognition accuracy achieved by MV-CNN on NinaPro DB1, DB2, DB3, DB4, and DB5 was 88.2%, 83.7%, 64.3%, 54.3%, and 90.0%, respectively. These results indicate that compared to MV-CNN, the HVPN framework can achieve better or similar intrasubject gesture recognition accuracy using less input data.

We also found that the intrasubject gesture recognition accuracy achieved by MV-CNN on NinaPro DB4 was much lower than that achieved by a shallow learning method (i.e., random forests [37]). By comparison, our proposed HVPN achieved the intrasubject gesture recognition accuracy of 72.9% on NinaPro DB4, with the sliding window length of 200 ms, which significantly outperformed both MV-CNN [31] and the random forests-based method [37].

*4.3. Comparison with MV-CNN Based on Intersubject Evaluation.* As very few studies in this field have presented the LOSOCV accuracy of recognizing all gestures in any of the NinaPro subdatabases, considering the difference in evaluation methodology and domain adaptation strategy, in this section, we focused on comparison with the MV-CNN framework [31], which used exactly the same intersubject evaluation scheme and domain adaptation technique as our proposed HVPN framework.

The LOSOCV accuracy achieved by MV-CNN and our proposed HVPN framework on five databases is presented in Table 5. The MV-CNN framework achieved the LOSOCV accuracy of 84.3%, 80.1%, 55.5%, 52.6%, and 87.2% on NinaProDB1, DB2, DB3, DB4, and DB5, respectively, with the sliding window length of 200 ms. By comparison, the HVPN framework achieved the LOSOCV accuracy of 84.9%, 82.0%, 65.6%, 70.2%, and 88.9% on NinaPro DB1, DB2, DB3, DB4, and DB5, respectively, with the sliding window length of 200 ms, which significantly outperformed MV-CNN. Similar to the results of intrasubject evaluation, the LOSOCV accuracy achieved by HVPN framework with the “two-view” configuration (i.e., denoted as HVPN-2-view in Table 5) also outperformed that achieved by MV-CNN

TABLE 4: Intrasubject gesture recognition accuracy in comparison with related works on five databases.

Machine learning (ML) model	Type of ML model	Input of ML model	Database	Num. of gestures for classification	Window length				Trial
					50 ms	100 ms	150 ms	200 ms	
Random forests [13]	Shallow learning	5 hand-crafted features	NinaProDB1	50	N.A.	N.A.	N.A.	75.3%	N.A.
Dictionary learning [36]	Shallow learning	MLSVD-based features	NinaProDB1	52	N.A.	N.A.	N.A.	N.A.	97.4%
HuNet [26]	CNN-RNN	Phinyomark feature set	NinaProDB1	52	N.A.	N.A.	86.8%	87.0%	97.3%
MV-CNN [31]	Multiview CNN	3 views of sEMG	NinaProDB1	52	85.8%	86.8%	87.4%	88.2%	N.A.
ChengNet [49]	CNN	Multi-sEMG-features image	NinaProDB1	52	N.A.	N.A.	N.A.	82.5%	N.A.
<b>HVPN-2-view</b>	<b>Multi-view CNN</b>	<b>2 views of sEMG</b>	<b>NinaProDB1</b>	<b>52</b>	<b>85.4%</b>	<b>86.5%</b>	<b>87.2%</b>	<b>88.1%</b>	<b>97.8%</b>
<b>HVPN</b>	<b>Multi-view CNN</b>	<b>Same as [31]</b>	<b>NinaProDB1</b>	<b>52</b>	<b>86.0%</b>	<b>86.9%</b>	<b>87.7%</b>	<b>88.4%</b>	<b>98.0%</b>
Random forests [13]	Shallow learning	Hand-crafted features	NinaProDB2	50	N.A.	N.A.	N.A.	75.3%	N.A.
ZhaiNet [25]	CNN	sEMG spectrogram	NinaProDB2	50	N.A.	N.A.	N.A.	78.7%	N.A.
HuNet [26]	CNN-RNN	Phinyomark feature set	NinaProDB2	50	N.A.	N.A.	N.A.	82.2%	97.6%
MV-CNN [31]	Multiview CNN	3 views of sEMG	NinaProDB2	50	80.6%	81.1%	82.7%	83.7%	N.A.
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>2 views of sEMG</b>	<b>NinaProDB2</b>	<b>50</b>	<b>82.7%</b>	<b>83.8%</b>	<b>83.3%</b>	<b>85.0%</b>	<b>97.8%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>Same as [31]</b>	<b>NinaProDB2</b>	<b>50</b>	<b>82.3%</b>	<b>84.1%</b>	<b>84.8%</b>	<b>85.8%</b>	<b>98.1%</b>
Support vector machine (SVM) [13]	Shallow learning	5 hand-crafted features	NinaProDB3	50	N.A.	N.A.	N.A.	46.3%	N.A.
MV-CNN [31]	Multiview CNN	3 views of sEMG	NinaProDB3	50	N.A.	N.A.	N.A.	64.3%	N.A.
ED-TCN [27]	TCN	MAV sequences	NinaProDB3	41	N.A.	N.A.	63.5%	N.A.	N.A.
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>2 views of sEMG</b>	<b>NinaProDB3</b>	<b>50</b>	<b>64.4%</b>	<b>65.7%</b>	<b>66.8%</b>	<b>67.9%</b>	<b>80.3%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>Same as [31]</b>	<b>NinaProDB3</b>	<b>50</b>	<b>64.5%</b>	<b>65.9%</b>	<b>66.9%</b>	<b>68.2%</b>	<b>80.7%</b>
Random forests [37]	Shallow learning	mDWT features	NinaProDB4	53	N.A.	N.A.	N.A.	69.1%	N.A.
MV-CNN [31]	Multiview CNN	3 views of sEMG	NinaProDB4	53	N.A.	N.A.	N.A.	54.3%	N.A.
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>2 views of sEMG</b>	<b>NinaProDB4</b>	<b>53</b>	<b>60.1%</b>	<b>63.2%</b>	<b>67.6%</b>	<b>72.1%</b>	<b>81.1%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>Same as [31]</b>	<b>NinaProDB4</b>	<b>53</b>	<b>58.3%</b>	<b>67.1%</b>	<b>70.5%</b>	<b>72.9%</b>	<b>81.7%</b>
SVM [37]	Shallow learning	mDWT features	NinaProDB5	41	N.A.	N.A.	N.A.	69.0%	N.A.
ShenNet [50]	Stacking-based CNN	TD, FD and TFD features	NinaProDB5	40	N.A.	N.A.	N.A.	72.1%	N.A.
MV-CNN [31]	Multiview CNN	3 views of sEMG	NinaProDB5	41	N.A.	N.A.	N.A.	90.0%	N.A.
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>2 views of sEMG</b>	<b>NinaProDB5</b>	<b>41</b>	<b>88.7%</b>	<b>89.1%</b>	<b>89.9%</b>	<b>90.1%</b>	<b>98.8%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>Same as [31]</b>	<b>NinaProDB5</b>	<b>41</b>	<b>88.7%</b>	<b>89.3%</b>	<b>90.0%</b>	<b>90.3%</b>	<b>98.4%</b>

N.A. denotes not applicable, and bold entries indicate our proposed method. HVPN-2-view refers to the proposed HVPN framework with the “two-view” configuration (i.e., using  $v_1$  and  $v_2$  as its input). †It should be mentioned that existing MCIs seldom segment raw sEMG signals by trial due to the constraint that the maximal response time of an MCI should be kept below 300 ms [40, 41]. ‡For experiments on HVPN, the predicted class label of each gesture trial is obtained by majority voting on all 200 ms sliding windows within it.

TABLE 5: LOSOCV accuracy in comparison with MV-CNN on five databases.

ML model	Type of ML model	Domain adaptation method	Database	Num. of gestures for classification	LOSOCV accuracy (achieved with 200 ms window)
MV-CNN [31]	Multiview CNN	MS-AdaBN	NinaProDB1	52	84.3%
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB1</b>	<b>52</b>	<b>84.5%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB1</b>	<b>52</b>	<b>84.9%</b>
MV-CNN [31]	Multiview CNN	MS-AdaBN	NinaProDB2	50	80.1%
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB2</b>	<b>50</b>	<b>81.8%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB2</b>	<b>50</b>	<b>82.0%</b>
MV-CNN [31]	Multiview CNN	MS-AdaBN	NinaProDB3	50	55.5%
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB3</b>	<b>50</b>	<b>65.4%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB3</b>	<b>50</b>	<b>65.6%</b>
MV-CNN [31]	Multiview CNN	MS-AdaBN	NinaProDB4	53	52.6%
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB4</b>	<b>53</b>	<b>69.9%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB4</b>	<b>53</b>	<b>70.2%</b>
MV-CNN [31]	Multiview CNN	MS-AdaBN	NinaProDB5	41	87.2%
<b>HVPN-2-view</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB5</b>	<b>41</b>	<b>88.8%</b>
<b>HVPN</b>	<b>Multiview CNN</b>	<b>MS-AdaBN</b>	<b>NinaProDB5</b>	<b>41</b>	<b>88.9%</b>

N.A. denotes not applicable, and bold entries indicate our proposed method. HVPN-2-view refers to the proposed HVPN framework with the “two-view” configuration (i.e., using  $v_1$  and  $v_2$  as its input).

framework on all databases, indicating that HVPN framework can achieve better LOSOCV accuracy than MV-CNN using less input data.

## 5. Conclusions

This paper proposed and implemented a hierarchical view pooling network (HVPN) framework, which improves multichannel sEMG-based gesture recognition by not only view-specific learning under each individual view but also view-shared learning in feature spaces that are hierarchically pooled from multiview low-level features.

Ablation studies were conducted on five multichannel sEMG databases (i.e., NinaPro DB1–DB5) to validate the effectiveness of the proposed framework. Results show the following: (1) when the FLVP layer in L2-VPCNN was replaced by conventional view pooling based on element-wise max pooling or average pooling, both intrasubject and LOSOCV accuracy degraded; (2) the proposed HVPN outperformed its two simplified variants that have only one view pooling CNN, as well as a deep neural network architecture that only consists of view-specific CNNs, in both intrasubject evaluation and LOSOCV. According to the above results, the effectiveness of the proposed hierarchical view pooling strategy can be proven.

Furthermore, we carried out performance comparison with the state-of-the-art methods on five databases (i.e.,

NinaPro DB1–DB5). Experimental results have demonstrated the superiority of the proposed HVPN framework over other deep learning and shallow learning-based methods. When using sliding windows of 200 ms, the proposed HVPN achieved the intrasubject gesture recognition accuracy of 88.4%, 85.8%, 68.2%, 72.9%, and 90.3% on NinaPro DB1, DB2, DB3, DB4, and DB5, respectively. The LOSOCV accuracy achieved on NinaPro DB1, DB2, DB3, DB4, and DB5 using 200 ms sliding windows was 84.9%, 82.0%, 65.6%, 70.2%, and 88.9%, respectively.

Limited by experimental conditions, we only considered offline experiments to verify our proposed HVPN framework. Our future work will focus on online evaluation of the proposed multiview deep learning framework. Moreover, in the future, we will investigate the integration of our proposed framework with hardware systems, such as upper-limb prostheses [51, 52] and space robots [53, 54] that are driven by multichannel sEMG signals.

## Data Availability

The multichannel sEMG data supporting the findings of this study are from the NinaPro dataset, which is publicly available at <http://ninapro.hevs.ch>. Papers describing the NinaPro dataset are cited at relevant places within the text as references [13, 37]. The processed data and trained deep

learning models used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors thank the NinaPro team for providing the publicity available sEMG databases. This work was supported in part by the National Natural Science Foundation of China under Grant nos. 62002171 and 61871224, the Natural Science Foundation of Jiangsu Province under Grant BK20200464, the National Key Research and Development Program of China under Grant 2020YFC2005302, and the Jiangsu Provincial Key Research and Development Program under Grant BE2018729.

## References

- [1] R. Beanbonyka, S. Nak-Jun, M. Sedong, and H. Min, "Deep learning in physiological signal data: a survey," *Sensors*, vol. 20, no. 4, p. 969, 2020.
- [2] X. Chen, S. Wang, C. Huang, S. Cao, and X. Zhang, "ICA-based muscle-tendon units localization and activation analysis during dynamic motion tasks," *Medical & Biological Engineering & Computing*, vol. 56, no. 3, pp. 341–353, 2018.
- [3] C. Li, G. Li, G. Jiang, D. Chen, and H. Liu, "Surface EMG data aggregation processing for intelligent prosthetic action recognition," *Neural Computing and Applications*, vol. 32, no. 22, pp. 16795–16806, 2018.
- [4] G. Shi, G. Xu, H. Wang, N. Duan, and S. Zhang, "Fuzzy-adaptive impedance control of upper limb rehabilitation robot based on sEMG," in *Proceedings of International Conference on Ubiquitous Robots*, pp. 745–749, Jeju, Korea, June 2019.
- [5] R. Ma, L. Zhang, G. Li, D. Jiang, S. Xu, and D. Chen, "Grasping force prediction based on sEMG signals," *Alexandria Engineering Journal*, vol. 59, no. 3, pp. 1135–1147, 2020.
- [6] U. Côté-Allard, G. Gagnon-Turcotte, F. Laviolette, and B. Gosselin, "A low-cost, wireless, 3-D-printed custom armband for sEMG hand gesture recognition," *Sensors*, vol. 19, no. 12, p. 2811, 2019.
- [7] Y. Yu, X. Chen, S. Cao, X. Zhang, and X. Chen, "Exploration of Chinese sign language recognition using wearable sensors based on deep belief net," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 5, pp. 1310–1320, 2020.
- [8] Y. Sun, C. Xu, G. Li et al., "Intelligent human computer interaction based on non redundant EMG signal," *Alexandria Engineering Journal*, vol. 59, no. 3, pp. 1149–1157, 2020.
- [9] C. Amma, T. Krings, J. Böer, and T. Schultz, "Advancing muscle-computer interfaces with high-density electromyography," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 929–938, Seoul, Republic of Korea, April 2015.
- [10] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Scientific Reports*, vol. 6, no. 1, p. 36571, 2016.
- [11] X. Chen, Y. Li, R. Hu, X. Zhang, and X. Chen, "Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, 2020.
- [12] R. N. Khushaba, S. Kodagoda, M. Takruri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10731–10738, 2012.
- [13] M. Atzori, A. Gijsberts, C. Castellini et al., "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific Data*, vol. 1, 2014.
- [14] A. Bahador, M. Yousefi, M. Marashi, and O. Bahador, "High accurate lightweight deep learning method for gesture recognition based on surface electromyography," *Computer Methods and Programs in Biomedicine*, vol. 195, Article ID 105643, 2020.
- [15] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface," *Pattern Recognition Letters*, vol. 119, pp. 131–138, 2019.
- [16] J. Chen, B. Sheng, G. Zhang, and G. Cao, "High-density surface EMG-based gesture recognition using a 3D convolutional neural network," *Sensors*, vol. 20, no. 4, p. 1201, 2020.
- [17] A. Phinyomark and E. Scheme, "EMG pattern recognition in the era of big data and deep learning," *Big Data and Cognitive Computing*, vol. 2, no. 3, p. 21, 2018.
- [18] D. Farina and R. Merletti, "Comparison of algorithms for estimation of EMG variables during voluntary isometric contractions," *Journal of Electromyography and Kinesiology*, vol. 10, no. 5, pp. 337–349, 2000.
- [19] L. Wu, X. Zhang, K. Wang, X. Chen, and X. Chen, "Improved high-density myoelectric pattern recognition control against electrode shift using data augmentation and dilated convolutional neural network," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2637–2646, 2020.
- [20] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, "A Hilbert curve based representation of semg signals for gesture recognition," in *Proceedings of 2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 201–206, Osijek, Croatia, June 2019.
- [21] T. Y. Pan, W. L. Tsai, C. Y. Chang, C. W. Yeh, and M. C. Hu, "A hierarchical hand gesture recognition framework for sports referee training-based EMG and accelerometer sensors," *IEEE Transactions on Cybernetics*, pp. 1–12, 2020, inpress.
- [22] Y.-C. Du, C.-H. Lin, L.-Y. Shyu, and T. Chen, "Portable hand motion classifier for multi-channel surface electromyography recognition using grey relational analysis," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4283–4291, 2010.
- [23] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.
- [24] F. Duan, L. Dai, W. Chang, Z. Chen, C. Zhu, and W. Li, "sEMG-based identification of hand motion commands using wavelet neural network combined with discrete wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1923–1934, 2016.
- [25] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers in Neuroscience*, vol. 11, p. 379, 2017.



- [26] Y. Hu, Y. Wong, W. Wei et al., “A novel attention-based hybrid CNN-RNN architecture for SEMG-based gesture recognition,” *PLoS One*, vol. 13, no. 10, pp. 1–18, Article ID e0206049, 2018.
- [27] J. L. Betthausen, J. T. Krall, S. G. Bannowsky et al., “Stable responsive EMG sequence prediction and adaptive reinforcement with temporal convolutional networks,” *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 6, pp. 1707–1717, 2020.
- [28] L. Chen, J. Fu, Y. Wu, H. Li, and B. Zheng, “Hand gesture recognition using compact CNN via surface electromyography signals,” *Sensors*, vol. 20, no. 3, p. 672, 2020.
- [29] S. Sun, “A survey of multi-view machine learning,” *Neural Computing and Applications*, vol. 23, no. 7, pp. 2031–2038, 2013.
- [30] J. Zhao, X. Xie, X. Xu, and S. Sun, “Multi-view learning overview: recent progress and new challenges,” *Information Fusion*, vol. 38, pp. 43–54, 2017.
- [31] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, “Surface-electromyography-based gesture recognition by multi-view deep learning,” *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 10, pp. 2964–2973, 2019.
- [32] D. Wang, W. Ouyang, W. Li, and D. Xu, “Dividing and aggregating network for multi-view action recognition,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.
- [33] Z. Shao, Y. Li, and H. Zhang, “Learning representations from skeletal self-similarities for cross-view action recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 160–174, 2021.
- [34] Y. Kong, Z. Ding, J. Li, and Y. Fu, “Deeply learned view-invariant features for cross-view action recognition,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 3028–3037, 2017.
- [35] M. Atzori, A. Gijsberts, S. Heynen et al., “Building the Ninapro database: a resource for the biorobotics community,” in *Proceedings of the IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1258–1265, Pisa, Italy, February 2012.
- [36] S. Padhy, “A tensor-based approach using multilinear SVD for hand gesture recognition from SEMG signals,” *IEEE Sensors Journal*, vol. 21, 2020.
- [37] S. Pizzolato, L. Tagliapietra, M. Cognolato et al., “Comparison of six electromyography acquisition setups on hand movement classification tasks,” *PLoS One*, vol. 12, no. 10, pp. 1–17, Article ID e0186132, 2017.
- [38] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” 2013, <http://arxiv.org/abs/1304.5634>.
- [39] W. Jiang and Z. Yin, “Human activity recognition using wearable sensors by deep convolutional neural networks,” in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1307–1310, Brisbane Australia, October 2015.
- [40] B. Hudgins, P. Parker, and R. N. Scott, “A new strategy for multifunction myoelectric control,” *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
- [41] K. Englehart and B. Hudgins, “A robust, real-time control scheme for multifunction myoelectric control,” *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, 2003.
- [42] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, “Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation,” *Sensors*, vol. 17, no. 3, p. 458, 2017.
- [43] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3D shape recognition,” in *Proceedings of IEEE International Conference on Computer Vision*, pp. 945–953, Santiago, Chile, December 2015.
- [44] T. He, H. Mao, and Z. Yi, “Moving object recognition using multi-view three-dimensional convolutional neural networks,” *Neural Computing and Applications*, vol. 28, no. 12, pp. 3827–3835, 2017.
- [45] S. Shin, R. Tafreshi, and R. Langari, “Robustness of using dynamic motions and template matching to the limb position effect in myoelectric classification,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 11, 2016.
- [46] A. Phinyomark, E. Campbell, and E. Scheme, “Surface electromyography (EMG) signal processing, classification, and practical considerations,” in *Biomedical Signal Processing*, pp. 3–29, Springer, Berlin, Germany, 2020.
- [47] L. Zhang, “Transfer adaptation learning: a decade survey,” 2019, <http://arxiv.org/abs/1903.04687>.
- [48] U. Côté-Allard, C. L. Fall, A. Campeau-Lecours et al., “Transfer learning for SEMG hand gestures recognition using convolutional neural networks,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1663–1668, Banff, Canada, October 2017.
- [49] Y. Cheng, G. Li, M. Yu et al., “Gesture recognition based on surface electromyography-feature image,” *Concurrency and Computation: Practice and Experience*, vol. 33, no. 6, Article ID e6051, 2021.
- [50] S. Shen, K. Gu, X.-R. Chen, M. Yang, and R.-C. Wang, “Movements classification of multi-channel sEMG based on CNN and stacking ensemble learning,” *IEEE Access*, vol. 7, pp. 137489–137500, 2019.
- [51] J. Fajardo, V. Ferman, D. Cardona, G. Maldonado, A. Lemus, and E. Rohmer, “Galileo hand: an anthropomorphic and affordable upper-limb prosthesis,” *IEEE Access*, vol. 8, pp. 81365–81377, 2020.
- [52] A. Prakash and S. Sharma, “A low-cost transradial prosthesis controlled by the intention of muscular contraction,” *Physical and Engineering Sciences in Medicine*, vol. 44, no. 1, pp. 229–241, 2021.
- [53] X. Zhang, J. Liu, Q. Gao, and Z. Ju, “Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique,” *Nonlinear Dynamics*, vol. 100, no. 3, pp. 2449–2467, 2020.
- [54] X. Zhang, J. Liu, J. Feng, Y. Liu, and Z. Ju, “Effective capture of nongrasable objects for space robots using geometric cage pairs,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 1, pp. 95–107, 2020.

## Research Article

# Feature Selection Based on a Large-Scale Many-Objective Evolutionary Algorithm

Yue Li,<sup>1</sup> Ziheng Sun ,<sup>1</sup> Xin Liu ,<sup>2</sup> Wei-Tung Chen,<sup>3</sup> Der-Juinn Horng,<sup>3</sup>  
and Kuei-Kuei Lai <sup>4</sup>

<sup>1</sup>State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin, China

<sup>2</sup>School of Economics and Management, Hebei University of Technology, Tianjin, China

<sup>3</sup>Department of Business Administration, NCU, Taoyuan, China

<sup>4</sup>Department of Business Administration of Chaoyang University of Technology, Taichung, China

Correspondence should be addressed to Xin Liu; [xinliu10@163.com](mailto:xinliu10@163.com) and Kuei-Kuei Lai; [laikk.tw@gmail.com](mailto:laikk.tw@gmail.com)

Received 3 April 2021; Revised 2 June 2021; Accepted 26 June 2021; Published 25 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Yue Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The feature selection problem is a fundamental issue in many research fields. In this paper, the feature selection problem is regarded as an optimization problem and addressed by utilizing a large-scale many-objective evolutionary algorithm. Considering the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance, a large-scale many-objective feature selection model is constructed. It is difficult to optimize the large-scale many-objective feature selection optimization problem by using the traditional evolutionary algorithms. Therefore, this paper proposes a modified vector angle-based large-scale many-objective evolutionary algorithm (MALSMEA). The proposed algorithm uses polynomial mutation based on variable grouping instead of naive polynomial mutation to improve the efficiency of solving large-scale problems. And a novel worst-case solution replacement strategy using shift-based density estimation is used to replace the poor solution of two individuals with similar search directions to enhance convergence. The experimental results show that MALSMEA is competitive and can effectively optimize the proposed model.

## 1. Introduction

Feature selection involves the selection of a specific number of features from existing features to optimize specific objectives [1]. Feature selection can be regarded as a multi-objective optimization problem that can be solved using evolutionary algorithms. Feature selection has attracted the attention of scholars and has been widely used in gene expression analysis [2], face recognition [3], and drug discovery [4]. For example, a two-stage heuristic algorithm minimal redundancy maximal relevance (mRMR) [5] is used to optimize relevance and redundancy simultaneously. A filter-based algorithm [6] is used to consider the entropy-based correlation measure and the combination measure of the redundancy and cardinality of a selected subset. A decomposition algorithm based on a weighted method is utilized to optimize interclass and intraclass distances [7].

Gulsah et al. [8] proposed two algorithms, W-QEISS and F-QEISS, that use nondominated sorting based on classification accuracy, feature number, relevance, and redundancy. Li et al. [9] established a model with feature number, classification performance, interclass distance, and intraclass distance as objectives and proposed a decomposition-based large-scale algorithm (DMEA-FS).

However, some unsolved problems still exist in feature selection using traditional evolutionary algorithms. The first problem is that the selection of a large number of features can be regarded as the optimization of the large-scale optimization problem [1] or the large-scale multiobjective optimization problem (LSMOP) [10], but the traditional evolutionary algorithms cannot effectively solve such problems. The second problem is that feature number and accuracy are two basic objectives, and other objectives are needed to explore the potential information to guide the

evolution in feature selection [1]. Correspondingly, more objectives result in many-objective optimization problems (MaOPs) [11, 12].

There are three main types of current algorithms, which are mainly used to solve LSMOPs or MaOPs, but they perform poorly on large-scale many-objective problems (LSMaOPs) [13], which include more than 3 objectives and over 100 decision variables [14, 15].

The first kind of algorithms is based on the Pareto dominance, which improves the convergence pressure by modifying the Pareto dominance relation. The new dominance relations are  $\varepsilon$ -dominance [16],  $\theta$ -dominance [17],  $L$ -optimality [18], simplex dominance [19], grid dominance [20, 21], etc. The algorithm using shift-based density estimation (SDE) was proposed in the work of [22], which allows individuals with poor convergence to obtain higher density.

The second is based on performance indicators, such as the hypervolume (HV) adaptive grid algorithm (HAGA) [23], the evolutionary algorithm (MaOEA/IGD) using inverted generational distance (IGD) [24], indicator-based algorithm with boundary protection (MaOEA-IBP) [25], and R2 indicator and weight vector-based method (R2-WVEA) [26]. Most of these algorithms are many-objective evolutionary algorithms (MaOEAs), but their computational costs are large.

The third category is composed of decomposition-based methods. The most classic ones are the multiobjective evolutionary algorithm based on decomposition (MOEA/D) [27] and its variants [28–30]. The algorithm based on nondominated sorting approach (NSGA-III) [31] uses evenly distributed reference points to assist the environmental selection. Based on NSGA-III, Gu and Wang [10] introduced an information feedback model to solve LSMaOPs. The reference vector-guided evolutionary algorithm (RVEA) [32] uses reference vectors to guide the optimization.

To more comprehensively describe and better solve the large-scale feature selection problem, this paper studies the existing multiobjective models based on the evolutionary algorithm, combines the existing objectives, constructs the feature selection problem as an LSMaOP, and uses an improved large-scale many-objective evolutionary algorithm (LSMaOEA) for optimization.

The main contributions of this paper are summarized as follows:

- (1) A novel worst-case solution replacement strategy based on SDE is proposed. This strategy allows conditional replacement of poor solutions in terms of convergence and diversity compared to other solutions, thereby maintaining a balance between convergence and diversity.
- (2) A modified vector angle-based large-scale many-objective evolutionary algorithm (MALSMEA) is proposed, which uses variable grouping-based polynomial mutation instead of naive polynomial mutation to improve the efficiency of solving large-scale problems. In the environmental selection

process, the proposed worst solution replacement strategy is used to improve diversity.

- (3) A large-scale many-objective feature selection optimization model is constructed, and MALSMEA is used to optimize it. The optimization objectives of this model are the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance.

The remainder of this paper is arranged as follows. Section 2 introduces the related works. Section 3 describes the proposed model and MALSMEA in detail. In Section 4, we compare and analyze the experimental results of MALSMEA and four advanced algorithms in solving benchmark LS-MaOPs, as well as the performance of MALSMEA and three feature selection algorithms in optimizing the proposed feature selection model. Section 5 provides a summary of the full paper and prospects of future research.

## 2. Related Works

*2.1. Large-Scale Many-Objective Optimization Problem.* An LSMaOP can be described as

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \quad (1)$$

where  $\Omega = \prod_{i=1}^D [l_i, u_i] \subseteq R^D$  is the decision space,  $D$  is the number of decision variables ( $D \geq 100$ ), and  $l_i$  and  $u_i$  are the lower and upper bounds of decision variables in the  $i$ th dimension, respectively.  $x$  is the  $D$ -dimensional decision vector in  $\Omega$ ,  $m$  is the objective number ( $m > 3$ ), and  $F(x) \in R^m$  is the objective vector of  $x$ . If no other solution dominates  $x$ , then  $x$  is a Pareto optimal solution [33]. The objective vectors corresponding to all Pareto optimal solutions constitute the Pareto optimal front (PF) [34, 35].

*2.2. Shift-Based Density Estimation.* We use the SDE [22] with the  $k$ th nearest neighbor [36] to estimate the density of all individuals. For an individual  $x_i$ , the following method is used to calculate the density value  $\text{SDE}(x_i)$ .

- (i) First, the standardized objective vectors of other individuals in population  $P$  are shifted.
- (ii) Then, the Euclidean distances between other shifted normalized objective vectors and the considered individual are calculated, expressed as  $d(x_i, x_k)$ .
- (iii) Next, the  $k$ th minimum value  $\lambda(x_i)$  in the set  $\{d(x_i, x_k), x_k \in P \cap x_k \neq x_i\}$  is found, where  $k = \sqrt{N}$  and  $N$  is the size of the population.
- (iv) Finally,  $\text{SDE}(x_i)$  is calculated as follows:

$$\text{SDE}(x_i) = \frac{1}{\lambda(x_i) + 2}. \quad (2)$$

Through the above process of estimating the individual density, we can observe that the smaller the individual density is, the better the performance of the individual.

Therefore, this paper uses this strategy, considering both diversity and convergence, to judge a pair of individuals with similar search direction, so as to delete the individual with poor performance.

**2.3. Information Theory Criterion Based on Entropy.** The feature selection model uses an entropy-based information theory criterion [8] to measure correlation and redundancy. For a given discrete random variable  $A$ , its entropy  $E(A)$  is determined as follows:

$$E(A) = - \sum_{a \in A} p(a) \log p(a), \quad (3)$$

where  $p(a) = \Pr(A = a)$ ,  $A$  is the set of all possible values of  $A$ ,  $a \in A$ . Then, the joint entropy of  $A$  and  $B$  is determined as follows:

$$E(A, B) = - \sum_{a \in A} \sum_{b \in B} p(a, b) \log p(a, b), \quad (4)$$

where  $B$  is a discrete random variable,  $p(a, b) = \Pr(A = a, B = b)$ ,  $a \in A$ , and  $b \in B$ . Then, the mutual information between  $A$  and  $B$  is determined as follows:

$$M(A, B) = E(A) + E(B) - E(A, B). \quad (5)$$

Symmetric uncertainty is used to scale the value range of mutual information to  $[0, 1]$  [37], which is defined as follows:

$$\text{SU}(A, B) = \frac{2M(A, B)}{E(A) + E(B)}. \quad (6)$$

### 3. Proposed Model and Algorithm

**3.1. Model Design.** The optimization objectives of the feature selection model include the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance, which are described as follows:

- (1) *The Number of Selected Features.* It is minimized to ensure the simplification of feature selection:

$$F_1(S) = |S|, \quad (7)$$

where  $|S|$  represents the cardinality of feature set  $S$ .

- (2) *Accuracy.* The accuracy of the learning algorithm is measured by the classification performance. The higher the classification performance is, the greater the accuracy. In this paper, the extreme learning machine (ELM) classifier [8] is used to calculate the accuracy:

$$F_2(S) = \frac{\text{tn} + \text{tp}}{\text{fn} + \text{fp} + \text{tn} + \text{tp}}, \quad (8)$$

where tn, tp, fn, and fp represent the true negative, true positive, false negative, and false positive, respectively.

- (3) *Relevance.* The relevance between features and categorical variables reflects the recognition ability of the selected features. The greater the correlation is, the stronger the recognition ability is:

$$F_3(S) = \sum_{x_i \in S} \text{SU}(x_i, y), \quad (9)$$

where  $x_i$  represents the  $i$ th feature and  $y$  represents the target categorical variable. This objective is normalized according to  $F_3(S) = F_3(S)/\max F_3(S)$ .

- (4) *Redundancy.* The redundancy is used to quantify the level of similarity between selected features. The smaller the redundancy is, the smaller the similarity:

$$F_4(S) = \sum_{x_i, x_j \in S, i < j} \text{SU}(x_i, x_j), \quad (10)$$

where  $x_j$  represents the  $j$ th feature. This objective is normalized according to  $F_4(S) = F_4(S)/\max F_4(S)$ .

- (5) *Interclass Distance.* The interclass distance represents the distance between the mean sample of each class and the average of mean samples of all classes, which reflects the recognition ability of samples of different classes. In the evolutionary process, a better sample distribution is obtained by maximizing the distance between classes:

$$F_5(S) = \sum_{i=1}^L \left( m_i - \frac{1}{L} \sum_{i=1}^L m_i \right)^2, \quad (11)$$

where  $L$  is the total number of classes and  $m_i$  is the average value of all samples with feature  $S$  in class  $i$ . This objective is normalized according to  $F_5(S) = F_5(S)/\max F_5(S)$ .

- (6) *Intraclass Distance.* By calculating the distances between the samples with the selected feature and the mean of all samples of the same kind, this value reflects the cohesion of the same kind of samples and can improve the accuracy to a certain extent:

$$F_6(S) = \sum_{i=1}^L \sum_{a_{ij} \in L_i} (a_{ij} - m_i)^2, \quad (12)$$

where  $a_{ij}$  is the  $j$ th sample in class  $i$ . This objective is normalized according to  $F_6(S) = F_6(S)/\max F_6(S)$ .

Therefore, the definition of the feature selection optimization model in this paper is as follows:

$$\min(F_1(S), -F_2(S), -F_3(S), F_4(S), -F_5(S), F_6(S)). \quad (13)$$

**3.2. The Proposed Algorithm: MALSMEA.** In this paper, a modified vector angle-based large-scale many-objective evolutionary algorithm is proposed, termed as MALSMEA. MALSMEA mainly uses a mutation operator based on variable grouping and the environment selection method of VaEA [38]. Figure 1 shows the program flowchart of MALSMEA. The main process of MALSMEA is as follows:

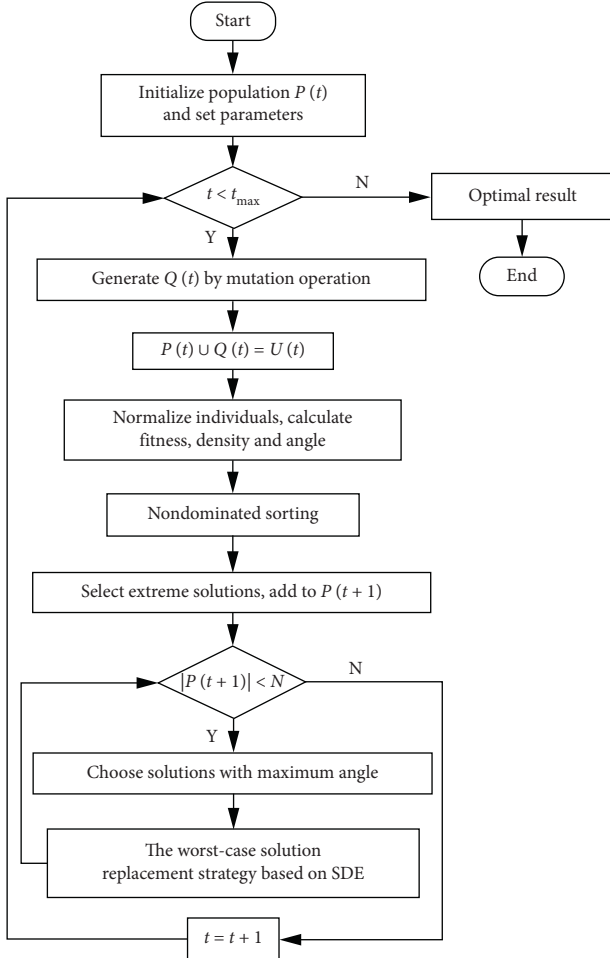


FIGURE 1: Program flowchart of MALSMEA.

- (i) *Step 1.* Initialize a population  $P(t)$  with  $N$  individuals randomly in the whole decision space  $\Omega$ , and set parameters.
- (ii) *Step 2.* The mutation operator based on variable grouping is used to mutate the population  $P(t)$ , in which the grouping method is ordered grouping, to generate the offspring population  $Q(t)$ .
- (iii) *Step 3.* Combine the offspring population  $Q(t)$  with the parent population  $P(t)$  and obtain the joint population  $U(t)$ . Then, the environmental selection in steps 4–9 is adopted to select  $N$  promising individuals from  $U(t)$ .
- (iv) *Step 4.* Normalize the individuals in the population  $U(t)$ , and calculate the fitness and density values of each individual as well as the vector angle between every two individuals.
- (v) *Step 5.* Use the nondominated sorting method to rank, and determine the last layer  $F(l)$ .
- (vi) *Step 6.* According to the vector angle between any two individuals in layer  $F(l)$  and the fitness value of each individual,  $m$  individuals with the largest vector angle and  $m$  individuals with the smallest

fitness value are selected to join  $P(t+1)$  to ensure the diversity.

- (vii) *Step 7.* If  $|P(t+1)| < N$ , select the individual with the largest vector angle in  $F(l)$  to join the new population  $P(t+1)$  by calculating the vector angles between the individuals in  $F(l)$  and the individuals in  $P(t+1)$ ; otherwise, go to step 9.
- (viii) *Step 8.* To maintain the balance between convergence and diversity, the worst individual replacement strategy is used to replace the poor individual with other individuals. Repeat from step 7 if  $|P(t+1)| < N$ .
- (ix) *Step 9.* Obtain the new population  $P(t+1)$ .
- (x) *Step 10.* Repeat from step 2, and stop when the maximum number of generations  $t_{\max}$  is reached.

**3.3. The Worst-Case Solution Replacement Strategy Based on SDE.** As the extreme individuals have been selected according to the vector angle and fitness value, for the worst individual replacement strategy in the process of environmental selection, we use the SDE strategy to calculate the density of individuals. The SDE strategy can consider the convergence and diversity of individuals simultaneously. Using this method, we can replace the poor individuals with similar search directions. The specific process is as follows: if the angle between an individual  $a$  in  $F(l)$  and an individual  $b$  in  $P(t+1)$  is less than the angle between two solutions of  $N$  ideal solutions, that is,  $\theta = ((\pi/2)/N + 1)$ , where  $N$  is the population size, then they have similar search directions. In this case, if  $SDE(a) < SDE(b)$ , then individual  $b$  is replaced by  $a$ . After replacement, the angle between each individual  $a \in F(l)$  and the new population  $P(t+1)$  is updated.

**3.4. The Wrapper Structure of MALSMEA.** MALSMEA is applied to the feature selection model, and the pseudocode of the wrapper structure of MALSMEA is shown in Algorithm 1. The main steps are as follows:

- (i) First, the input dataset DS is divided into training and test datasets.
- (ii) Then, in the initialization process, MALSMEA allocates the random feature vector  $W_S$  selected from the data feature matrix  $W$ . The selected feature vector  $W_S$  is encoded as solutions by using the coding technology of [9] to reduce the amount of computation in the evolutionary process, and the mask of  $W_S$  is regarded as the decision variables, and the population  $P$  is formed.
- (iii) Then, in the wrapper structure, the population  $P$  is evaluated via six objective functions to obtain objective vectors and obtain the evaluated population  $P(t)$ . The feature number is calculated according to the decision variables of the solutions. The accuracy can be obtained from the decoded feature subset and the corresponding ELM classifier [8], and other objectives can be calculated according to the corresponding equations.

- (iv) Then, the population is optimized by MALSMEA.
- (v) Finally, the optimal set  $P_S$  is obtained.

**3.5. Time Complexity Analysis.** The time complexity of MALSMEA is composed mainly of the following parts: the time complexity of the mutation operation in MALSMEA is  $O(D^2N/K)$ , where  $K$  is the number of groups, the time complexity of nondominated sorting is  $O(N \log^{m-2} N)$  [31], the worst-case solution replacement strategy based on SDE has the time complexity of  $O(mN^2)$ , and the time complexity of other operations is  $O(mN^2)$ . Therefore, the time complexity of MALSMEA is  $\max\{O(D^2N/K), O(N \log^{m-2} N), O(mN^2)\}$ . Compared with the four algorithms, the time complexity of the grouped and linked polynomial mutation operator (GLMO) is  $\max\{O(D^2N/K), O(mN^2)\}$  [39], linear combination-based search algorithm (LCSA) is  $O(mN^2)$  [40], vector angle-based evolutionary algorithm (VaEA) is  $\max\{O(N \log^{m-2} N), O(mN^2)\}$  [38], and RVEA is  $O(mN^2)$  [32]. Thus, the time complexity of MALSMEA is similar to that of GLMO but greater than that of the other three algorithms.

## 4. Experimental Studies

In this section, DTLZ1-DTLZ6 in the Deb, Thiele, Laumanns, and Zitzler (DTLZ) test suite [41] and LSMOP1-LSMOP9 in the Large-Scale Multi- and Many-Objective Problems (LSMOP) test suite [42] are selected to evaluate the performance of MALSMEA, and four datasets in the University of California at Irvine (UCI) machine learning library [43] are selected to evaluate the ability of MALSMEA to optimize the proposed feature selection model, among which Heart is a two-class dataset, Zoo and Iris are two multiclass datasets, and Musk1 is a high-dimensional dataset. For LSMaOPs, MALSMEA is compared with GLMO [39], LCSA [40], VaEA [38], and RVEA [32]. GLMO and LCSA are large-scale multiobjective evolutionary algorithms. GLMO uses mutation operators based on variable grouping, and LCSA uses a linear combination to reduce dimensionality. VaEA and RVEA are many-objective evolutionary algorithms that use vector angles and reference vectors, respectively. For the proposed six-objective feature selection model, MALSMEA is compared with W-MOSS [44], W-QEISS, and F-QEISS [8].

In the next sections, we introduce the performance indicators and set the parameters in the experiments. Then, for all algorithms, when the objective numbers  $m$  are 5 and 10, the population sizes  $N$  are 126 and 275, and the numbers of decision variables  $D$  are 500 and 1000, respectively. Each algorithm runs 20 times independently and stops when the number of function evaluations (FEs) reaches 90,000. The performance of MALSMEA is verified by comparing the average IGD values obtained by five algorithms. In each test instance, the best average IGD value is highlighted in bold. Finally, in four datasets, MALSMEA and three feature selection algorithms are utilized to deal with the proposed six-objective feature selection optimization model, for which

$N = 100$ , the maximum number of FEs is 100, and each algorithm runs independently for 10 times. The optimization ability of MALSMEA is verified by comparing the HV indicator and optimization results.

### 4.1. Experimental Settings

- (1) *Performance Indicator.* In the experiment, IGD [45] and HV [46] are used as evaluation indicators. The smaller (larger) the IGD (HV) indicator value is, the better the performance of the algorithm. The IGD indicator evaluates the algorithm by calculating the average of minimum distances between all sampled individuals on the actual PF and the obtained solution set. The HV indicator quantifies the algorithm performance by calculating the volume between the obtained nondominated solution set and the reference point.
- (2) *Parameter Settings for the Crossover and Mutation Operators.* In the performance verification experiment of MALSMEA, MALSMEA and GLMO use the mutation operator based on variable grouping to generate offspring. Other algorithms use simulated binary crossover (SBX) [32] and polynomial mutation [47]. The crossover probability is  $p_c = 1.0$ , the mutation probability is  $p_m = 1/D$ , and the distribution indicator is  $\eta_m = 20$ , where  $D$  is the number of decision variables. In the experiment to verify the superiority of MALSMEA with respect to the proposed model, according to [9],  $p_c = 0.8$ ,  $p_m = 0.2$ .
- (3) *Other Parameter Settings for Algorithms.* In MALSMEA and GLMO [39], the number of groups  $K$  is set to 4, and the ordered grouping method is adopted. For RVEA [32], the index  $\alpha$  and the frequency  $f_r$  are set to 2 and 0.1, respectively. The parameters in W-QEISS and F-QEISS are set according to [8], and the searching method is based on r-NSGA-II [48]. The parameters in W-MOSS are set according to [44].
- (4) *Datasets.* The details of 4 UCI datasets utilized are shown in Table 1.
- (5) *ELM Classifier.* For the proposed model, the ELM classifier [8] is utilized to evaluate the accuracy of the current solution, which follows the criterion given in [46]: the activation function is  $g(x) = 1/(1 + e^{-x})$  in the hidden layer, and the number of neurons is set to  $n_h = 10$ . The target classification variable and the (input) features are normalized into ranges  $[0, 1]$  and  $[-1, 1]$  in each dataset, respectively. To minimize the accuracy deviation, the  $k$ -fold cross validation approach is utilized with  $k = 10$ , and the average accuracy is used for comparison [9].

**4.2. Performance Comparison of Algorithms on DTLZ.** Table 2 describes the IGD indicator values obtained by the five algorithms on the 5- and 10-objective DTLZ1-DTLZ6 with 500 and 1000 decision variables. As shown in Table 2,

<p><b>Input:</b> Datasets with labels, DS; the maximal number of generations, <math>t_{\max}</math>; the population size, <math>N</math>;</p> <p><b>Output:</b> The Pareto subset, <math>P_S</math>;</p> <p>(1) divide DS into training and test datasets;</p> <p>(2) <math>[W, Y] = \text{Segment}(\text{training datasets})</math>;</p> <p>(3) <math>S = \text{Encoding}(W_S)</math>; <math>W_S = \text{Feature Select}(W)</math>;</p> <p>(4) <math>P(t) = \text{Evaluate Six Objectives}(P)</math>; <math>P = \text{Initialize}(N, S)</math>;</p> <p>(5) <math>P(t) = \text{MALSMEA}(P(t))</math>;</p> <p>(6) <math>P_S \leftarrow P(t)</math>;</p>
--

ALGORITHM 1: The wrapper structure of MALSMEA.

TABLE 1: The information of four UCI datasets.

Dataset	Classes	Features	Instance
Heart	2	13	270
Zoo	7	16	101
Iris	3	4	150
Musk1	2	166	476

MALSMEA is competitive with the other four algorithms. Specifically, MALSMEA produces 18 best results out of 24 test instances, and its performance on the 10-objective DTLZ is significantly better than that of the other algorithms. The experimental results are analyzed in detail as below.

DTLZ1 reflects the convergence of the algorithm. MALSMEA outperforms the other algorithms on the 5- and 10-objective DTLZ1. These results demonstrate that MALSMEA has better convergence on the large-scale high-dimensional DTLZ1. DTLZ2 is generally used to test the scalability of algorithms with respect to the number of objectives. The performance of MALSMEA on the 5-objective DTLZ2 is better than that of LCSA but slightly inferior to that of GLMO, VaEA, and RVEA. The performance of MALSMEA on the 10-objective DTLZ2 is better than that of the other four algorithms. Thus, MALSMEA has better scalability to the objective number.

DTLZ3 is a highly multimodal problem similar to DTLZ1. MALSMEA obtains the smallest IGD indicator value on DTLZ3 with 500 and 1000 decision variables. DTLZ4 is used to test the ability of the algorithm to ensure the diversity of the population. MALSMEA obtains the smallest IGD indicator value on the 10-objective DTLZ4 with 500 and 1000 decision variables. For the 5-objective DTLZ4, VaEA outperforms other algorithms on DTLZ4 with 500 and 1000 decision variables. MALSMEA exhibits greater diversity on the large-scale 10-objective DTLZ4.

For the 5-objective DTLZ5, MALSMEA outperforms LCSA on DTLZ5 with 500 and 1000 decision variables, but inferior to GLMO, VaEA, and RVEA. For the 10-objective DTLZ5, MALSMEA outperforms its counterparts. For DTLZ6, the overall performance of MALSMEA is optimal on instances with up to 1000 decision variables.

To further test the performance of MALSMEA, the nonparametric Friedman test [49] is employed. According to the average IGD indicator values of the five algorithms on DTLZ, Table 3 indicates the average ranking of the five algorithms. The average ranking of MALSMEA is the

smallest, which indicates that MALSMEA performs the best. The average ranking of LCSA is the largest, so its performance is the worst.

To verify the efficiency of MALSMEA, Table 4 presents the running time of MALSMEA and the four other algorithms on the 10-objective DTLZ1 with 1000 decision variables. The running times of MALSMEA and GLMO are quite similar but greater than those of other algorithms.

#### 4.3. Performance Comparison of Algorithms on LSMOP.

LSMOP is proposed to test the performance of the algorithm in LSMaOPs. Table 5 lists the IGD indicator values obtained by five algorithms on 5- and 10-objective LSMOP1-LSMOP9 with 500 and 1000 decision variables. MALSMEA produces 26 best results out of 36 test instances. Therefore, compared with the other four algorithms, MALSMEA has better performance in solving LSMaOPs.

Specifically, for the LSMOP test suite with 500 decision variables, MALSMEA outperforms the other algorithms on the 5- and 10-objective LSMOP2, LSMOP4, LSMOP5, LSMOP8, and LSMOP9. MALSMEA is inferior to LCSA on LSMOP3. MALSMEA outperforms the other algorithms on the 10-objective LSMOP1 and LSMOP7, but LCSA obtains the smallest IGD indicator value on the 5-objective LSMOP1 and LSMOP7. MALSMEA obtains the smallest IGD indicator value on the 5-objective LSMOP6, while RVEA performs better on the 10-objective LSMOP6.

For the LSMOP test suite with 1000 decision variables, MALSMEA outperforms the other algorithms on the 5- and 10-objective LSMOP2, LSMOP4, LSMOP5, LSMOP8, and LSMOP9. MALSMEA is inferior to LCSA on LSMOP3. LCSA obtains the best performance on the 5-objective LSMOP1 and LSMOP7, and MALSMEA outperforms the other algorithms on the 10-objective LSMOP1 and LSMOP7. The performance of MALSMEA on the 5-objective LSMOP6 is better than that of the other algorithms, but it is slightly inferior to that of LCSA and RVEA on the 10-objective LSMOP6.

#### 4.4. Comparison of the Optimization Results on the Proposed Model.

Table 6 shows the HV indicator values and objective values of the four algorithms after optimization on four datasets. The results demonstrate that MALSMEA obtains the maximum HV indicator values, showing that MALSMEA has certain advantages in feature selection. As noted in

TABLE 2: Performance comparison between MALSMEA and four algorithms with respect to the average IGD values on the DTLZ1-DTLZ6 (gray values represent the best values in each row).

Problem	$m$	$D$	MALSMEA	GLMO	LCSA	VaEA	RVEA
DTLZ1	5	500	1.1079e+3 (4.24e+2)	9.9478e+3 (1.61e+3)	3.9526e+3 (2.52e+2)	4.5327e+3 (2.97e+2)	7.8347e+3 (1.99e+2)
		1000	3.6284e+3 (1.03e+3)	1.8810e+4 (2.97e+3)	7.7836e+3 (4.34e+2)	1.3520e+4 (5.87e+2)	1.8532e+4 (3.84e+2)
	10	500	2.2202e+3 (3.47e+2)	9.4305e+3 (5.82e+2)	4.5825e+3 (3.67e+2)	8.4640e+3 (3.85e+2)	7.2316e+3 (7.61e+2)
		1000	4.7828e+3 (9.45e+2)	1.8648e+4 (9.68e+2)	9.2419e+3 (4.31e+2)	1.8151e+4 (5.38e+2)	1.6042e+4 (3.96e+2)
DTLZ2	5	500	2.8988e+1 (1.27e+0)	3.0185e+1 (4.19e+0)	2.9554e+1 (2.79e+0)	4.0643e+0 (2.88e-1)	2.5720e+0 (1.80e-1)
		1000	6.6661e+1 (2.14e+0)	6.2634e+1 (6.02e+0)	7.7019e+1 (5.98e+0)	1.8169e+1 (8.34e-1)	1.4208e+1 (6.65e-1)
	10	500	2.1635e+1 (3.68e+0)	3.7764e+1 (4.64e+0)	4.2687e+1 (1.30e+1)	2.4451e+1 (1.05e+0)	2.6761e+1 (7.91e+0)
		1000	4.4200e+1 (7.77e+0)	7.7862e+1 (7.30e+0)	7.9824e+1 (2.29e+0)	5.8910e+1 (1.32e+0)	5.0172e+1 (1.07e+0)
DTLZ3	5	500	4.3020e+3 (1.57e+3)	2.3235e+4 (6.13e+3)	1.2346e+4 (7.98e+0)	1.8602e+4 (7.53e+2)	3.1291e+4 (6.31e+2)
		1000	1.0670e+4 (2.63e+3)	4.3783e+4 (8.98e+3)	2.4844e+4 (1.47e+1)	6.0220e+4 (1.81e+3)	7.6232e+4 (9.67e+3)
	10	500	1.3306e+4 (1.60e+3)	4.1894e+4 (2.65e+3)	1.4213e+4 (1.03e+1)	3.8615e+4 (7.37e+2)	3.9350e+4 (7.75e+2)
		1000	2.5625e+4 (4.16e+3)	8.4954e+4 (6.42e+3)	2.7420e+4 (9.93e+0)	8.5528e+4 (1.17e+3)	8.8290e+4 (1.15e+3)
DTLZ4	5	500	2.6523e+1 (1.65e+0)	3.5059e+1 (5.11e+0)	2.6238e+1 (3.03e+0)	5.6372e+0 (3.94e-1)	5.9172e+0 (7.12e-1)
		1000	5.8758e+1 (3.00e+0)	6.6782e+1 (1.20e+1)	6.9454e+1 (3.04e+0)	2.2961e+1 (9.24e-1)	2.9794e+1 (2.77e+0)
	10	500	2.3290e+1 (1.42e+0)	3.3557e+1 (9.70e+0)	4.0018e+1 (1.84e+0)	2.4352e+1 (8.56e-1)	2.5171e+1 (5.22e-1)
		1000	4.9795e+1 (3.90e+0)	7.0596e+1 (1.42e+1)	8.0600e+1 (2.05e+0)	5.8910e+1 (1.22e+0)	5.6250e+1 (1.07e+0)
DTLZ5	5	500	2.8696e+1 (1.44e+0)	2.4279e+1 (7.19e+0)	3.5655e+1 (1.04e+0)	7.8308e+0 (6.53e-1)	2.9302e+0 (2.18e-1)
		1000	6.3241e+1 (2.51e+0)	3.7272e+1 (1.12e+1)	7.4941e+1 (1.88e+0)	2.7873e+1 (1.40e+0)	1.6365e+1 (4.94e-1)
	10	500	2.2663e+1 (4.07e+0)	2.2874e+1 (7.97e+0)	4.0439e+1 (4.57e+0)	2.7748e+1 (1.08e+0)	2.6317e+1 (8.24e+0)
		1000	4.8397e+1 (7.02e+0)	4.8756e+1 (1.63e+1)	8.1209e+1 (2.71e+0)	6.3840e+1 (1.28e+0)	4.9904e+1 (1.13e+0)
DTLZ6	5	500	8.8879e+0 (1.39e+0)	4.2732e+2 (2.36e+1)	9.4574e+0 (8.51e+0)	3.8495e+2 (4.90e+0)	3.6416e+2 (2.58e+0)
		1000	1.9706e+1 (3.29e+0)	8.9090e+2 (2.39e+1)	2.9199e+1 (1.29e+1)	8.1717e+2 (6.01e+0)	8.0078e+2 (3.00e+0)
	10	500	5.4188e+1 (1.00e+1)	4.2710e+2 (1.53e+1)	7.0212e+1 (1.23e+1)	4.1523e+2 (2.44e+0)	4.1207e+2 (2.66e+0)
		1000	1.0773e+2 (3.16e+1)	8.6234e+2 (4.55e+1)	1.1227e+2 (8.94e+1)	8.5524e+2 (2.99e+0)	8.5757e+2 (2.39e+0)

TABLE 3: Average rankings of the Friedman test.

Algorithm	Ranking
MALSMEA	2.1667
GLMO	3.4583
LCSA	3.6667
VaEA	2.9583
RVEA	2.75

TABLE 4: Comparison of running time between MALSMEA and the other four algorithms.

Algorithm	Time
MALSMEA	2.3113e+2
GLMO	2.0182e+2
LCSA	4.3017e+1
VaEA	1.2587e+2
RVEA	6.8803e+1

TABLE 5: Performance comparison between MALSMEA and four algorithms with respect to the average IGD values on the LSMOP1-LSMOP9 (gray values represent the best values in each row).

Problem	$m$	$D$	MALSMEA	GLMO	LCSA	VaEA	RVEA
LSMOP1	5	500	1.3173e+0 (1.55e-1)	9.9913e-1 (1.05e-1)	9.3999e-1 (5.30e-3)	1.6687e+0 (2.66e-1)	1.2713e+0 (1.54e-1)
		1000	1.3109e+0 (1.61e-1)	1.2099e+0 (5.21e-1)	9.3942e-1 (2.67e-3)	3.6704e+0 (4.00e-1)	2.6898e+0 (2.09e-1)
	10	500	1.2008e+0 (1.89e-1)	5.9934e+0 (2.79e+0)	1.2010e+0 (1.16e-3)	4.1745e+0 (1.28e+0)	1.6742e+0 (3.51e-1)
		1000	1.1728e+0 (1.53e-1)	7.9449e+0 (3.19e+0)	1.1938e+0 (2.75e-3)	7.0153e+0 (6.50e-1)	4.0353e+0 (9.27e-1)



TABLE 5: Continued.

Problem	$m$	$D$	MALSMEA	GLMO	LCSA	VaEA	RVEA
LSMOP2	5	500	1.5237e-1 (1.77e-3)	1.8423e-1 (5.16e-3)	1.9821e-1 (6.56e-3)	1.6390e-1 (1.71e-3)	1.6594e-1 (9.99e-4)
		1000	1.3444e-1 (1.08e-3)	1.6139e-1 (4.75e-3)	1.7402e-1 (3.87e-3)	1.4188e-1 (1.73e-3)	1.4299e-1 (8.72e-4)
	10	500	2.8094e-1 (6.69e-3)	3.3525e-1 (7.25e-3)	3.6322e-1 (8.55e-3)	3.1995e-1 (3.89e-3)	2.8197e-1 (3.56e-3)
		1000	2.3979e-1 (2.71e-3)	2.8301e-1 (5.22e-3)	3.0751e-1 (7.90e-3)	2.6900e-1 (1.85e-3)	2.3980e-1 (3.04e-3)
LSMOP3	5	500	1.1955e+1 (3.86e+0)	1.3626e+0 (6.23e-1)	9.5883e-1 (0.00e+0)	1.6636e+1 (4.85e+0)	4.7605e+0 (1.27e+0)
		1000	1.3419e+1 (4.38e+0)	1.4773e+0 (5.34e-1)	9.5883e-1 (0.00e+0)	1.6875e+1 (5.62e+0)	8.7885e+0 (1.03e+0)
	10	500	1.2546e+1 (1.59e+0)	2.1075e+2 (3.43e+2)	1.8733e+0 (1.57e-3)	1.7999e+1 (3.05e+0)	2.4510e+0 (4.99e-1)
		1000	1.3071e+1 (1.29e+0)	1.1423e+4 (1.26e+2)	1.9179e+0 (8.35e-4)	1.9379e+1 (2.80e+0)	4.3816e+1(1.40e+0)
LSMOP4	5	500	2.8356e-1 (8.13e-3)	3.3698e-1 (1.31e-2)	3.2856e-1 (9.98e-3)	3.0856e-1 (5.78e-3)	2.8894e-1 (2.96e-3)
		1000	2.1150e-1 (5.31e-3)	2.4674e-1 (7.40e-3)	2.5458e-1 (6.51e-3)	2.1842e-1 (3.10e-3)	2.1661e-1 (1.51e-3)
	10	500	3.3748e-1 (5.61e-3)	3.9190e-1 (1.04e-2)	4.3146e-1 (1.52e-2)	3.7828e-1 (3.79e-3)	3.4044e-1 (3.98e-3)
		1000	2.7003e-1 (2.36e-3)	3.1838e-1 (8.76e-3)	3.5483e-1 (6.41e-3)	3.0457e-1 (3.65e-3)	2.7902e-1 (3.82e-3)
LSMOP5	5	500	4.5817e-1 (5.45e-3)	3.3566e+0 (3.16e+0)	4.6074e-1 (3.81e-2)	4.5633e+0 (3.26e-1)	1.8603e+0 (3.83e-1)
		1000	4.5647e-1 (2.97e-2)	8.3782e+0 (6.28e+0)	4.5874e-1 (1.99e-2)	7.4372e+0 (7.67e-1)	3.3211e+0 (5.08e-1)
	10	500	6.5504e-1 (4.37e-2)	1.6148e+1 (8.45e+0)	1.1132e+0 (8.69e-2)	8.4930e+0 (1.21e+0)	3.0758e+0 (5.69e-1)
		1000	6.6973e-1 (6.22e-2)	1.4246e+1 (6.02e+0)	1.1087e+0 (9.32e-2)	1.0274e+1 (1.04e+0)	6.1324e+0 (5.95e-1)
LSMOP6	5	500	1.2094e+0 (1.33e-1)	5.3807e+2 (1.68e+3)	1.2106e+0 (3.67e-2)	1.1135e+1 (5.75e+0)	8.3040e+0 (1.66e+1)
		1000	1.2188e+0 (8.52e-2)	2.5183e+3 (4.35e+3)	1.2549e+0 (5.34e-2)	1.4415e+2 (3.65e+1)	5.3053e+1 (2.99e+1)
	10	500	1.4348e+0 (1.42e-1)	6.0471e+1 (1.88e+2)	1.4179e+0 (8.13e-2)	1.3763e+2 (2.90e+2)	1.2580e+0 (1.09e-1)
		1000	1.4961e+0 (1.46e-1)	7.6272e+2 (3.01e+3)	1.3573e+0(7.95e-2)	1.5136e+0 (8.68e-3)	1.2743e+0 (9.00e-2)
LSMOP7	5	500	1.3323e+0 (6.63e-2)	2.4841e+0 (3.00e-1)	1.0912e+0 (1.46e-2)	2.9317e+0 (1.47e-1)	1.2645e+0 (1.88e-1)
		1000	1.3577e+0 (6.26e-2)	1.7911e+0 (1.01e-1)	1.0321e+0 (1.40e-2)	1.9182e+0 (5.40e-2)	1.1214e+0 (8.68e-2)
	10	500	1.3995e+0 (7.97e-2)	3.5137e+4 (1.36e+4)	1.5578e+0 (5.12e-2)	1.0739e+3 (7.45e+2)	2.6040e+1 (6.95e+0)
		1000	1.4663e+0 (1.11e-1)	3.7805e+4 (1.15e+4)	1.5933e+0 (5.63e-2)	2.7102e+3 (1.09e+3)	1.4501e+2 (3.01e+1)
LSMOP8	5	500	3.8850e-1 (2.43e-2)	1.1661e+0 (7.11e-2)	3.8922e-1 (1.02e-2)	1.1767e+0 (9.67e-3)	9.3066e-1 (1.19e-1)
		1000	3.9206e-1 (3.27e-2)	1.0697e+0 (9.46e-2)	3.9962e-1 (8.72e-3)	1.1544e+0 (1.25e-3)	8.9791e-1 (1.45e-1)
	10	500	6.4152e-1 (4.00e-2)	1.2619e+1 (4.49e+0)	9.6995e-1 (9.27e-2)	2.8446e+0 (5.01e-1)	1.4025e+0 (1.12e-1)
		1000	6.2434e-1 (3.37e-2)	1.1402e+1 (4.19e+0)	1.0886e+0 (1.06e-1)	4.0270e+0 (6.02e-1)	2.6957e+0 (3.85e-1)

TABLE 5: Continued.

Problem	$m$	$D$	MALSMEA	GLMO	LCSA	VaEA	RVEA
LSMOP9	5	500	2.8005e+0 (2.91e-8)	2.9775e+0 (9.23e-2)	2.9985e+0 (8.77e-3)	1.2971e+1 (2.27e+0)	2.5483e+1 (6.20e+0)
		1000	2.9801e+0 (9.11e-2)	2.9976e+0 (9.44e-2)	3.0005e+0 (0.00e+0)	3.5883e+1 (3.99e+0)	5.5544e+1 (1.95e+1)
	10	500	6.4182e+0 (1.93-1)	6.5037e+0 (7.63e-1)	6.5321e+0 (3.65e-15)	3.6094e+2 (2.89e+1)	2.7313e+2 (9.11e+1)
		1000	6.3652e+0 (2.05e-1)	6.3891e+0 (1.06e+0)	6.5321e+0 (3.65e-15)	5.0223e+2 (2.77e+1)	3.4370e+2 (9.37e+1)

TABLE 6: HV values and optimized results of four algorithms (values in bold represent better results).

Dataset	Algorithm	HV	Feature	Accuracy	Relevance	Redundancy	Interclass distance	Intraclass distance
Heart	MALSMEA	<b>0.9972</b>	<b>6</b>	<b>0.7979</b>	0.4615	0.1923	<b>0.0802</b>	<b>0.0123</b>
	W-MOSS	0.9962	7	0.7667	0.5385	0.2692	0.0769	0.0128
	W-QEISS	0.9943	8	0.7604	<b>0.6154</b>	0.3590	0.0764	0.0130
	F-QEISS	0.9980	7	0.7811	0.5385	<b>0.0256</b>	0.0798	0.0125
Zoo	MALSMEA	<b>0.9979</b>	<b>5</b>	<b>0.9842</b>	0.3125	0.0833	<b>0.0637</b>	<b>0.0074</b>
	W-MOSS	0.9975	7	0.9816	0.4375	0.1750	0.0622	0.0085
	W-QEISS	0.9972	7	0.9697	<b>0.5000</b>	0.2333	0.0615	0.0076
	F-QEISS	0.9977	6	0.9556	0.3750	<b>0.0167</b>	0.0609	0.0083
Iris	MALSMEA	<b>0.9351</b>	<b>2</b>	<b>0.9387</b>	0.5000	<b>0.1667</b>	<b>0.2574</b>	<b>0.1667</b>
	W-MOSS	0.9234	3	0.9071	<b>0.7500</b>	0.5000	0.2566	0.1673
	W-QEISS	0.9236	3	0.9049	0.5655	0.1765	0.2571	0.1670
	F-QEISS	0.9247	3	0.9187	<b>0.7500</b>	<b>0.1667</b>	0.2569	0.1668
Musk1	MALSMEA	<b>0.9697</b>	<b>11</b>	<b>0.6173</b>	0.0663	<b>0.0045</b>	<b>7.3102e-5</b>	<b>0.0060</b>
	W-MOSS	0.9693	12	0.6130	0.0723	0.0048	7.3023e-5	<b>0.0060</b>
	W-QEISS	0.9603	13	0.5956	<b>0.0783</b>	0.0057	7.3037e-5	0.0067
	F-QEISS	0.9627	13	0.6069	<b>0.0783</b>	0.0057	7.3026e-5	0.0062

Table 6, for the four datasets, the optimization performance of MALSMEA is better on Iris and Musk1. MALSMEA is slightly inferior to the other three algorithms in relevance and redundancy but exhibits better performance in the other four objectives. In addition, W-QEISS and F-QEISS are relatively better than the other algorithms in terms of relevance and redundancy, but they are worse in other objectives.

## 5. Conclusion

In this paper, a modified vector angle-based large-scale many-objective evolutionary algorithm called MALSMEA is proposed. In MALSMEA, the polynomial mutation based on variable grouping is used to replace the polynomial mutation to improve the efficiency of solving large-scale optimization problems. A novel worst-case solution replacement strategy based on SDE is proposed to replace the worse one of two individuals with similar search directions to increase diversity. In addition, MALSMEA is compared with four typical algorithms to solve the optimization problem with up to 10 objectives and 1000 decision variables. Experimental results indicate that MALSMEA outperforms the four algorithms on the DTLZ and LSMOP test suites. By studying the existing feature selection models, taking the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance as the optimization objectives, a six-objective optimization model is constructed

and solved by using MALSMEA. Compared with the other three feature selection algorithms, MALSMEA has some advantages in solving this model.

Future studies will proceed in two directions. The first direction is to add a parallel strategy to MALSMEA to improve efficiency or to further modify its environmental selection method. Another research direction is to solve LSMaOPs in other fields using MALSMEA.

## Data Availability

The details of the four UCI datasets utilized are shown in Table 1.

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant no. 61976242, in part by the Fundamental Scientific Research Funds for Interdisciplinary Team of Hebei University of Technology under Grant no. JBKYTD2002, and in part by the Guangdong Provincial Key Laboratory under Grant no. 2020B121201001.

## References

- [1] M. Komeili, W. Louis, N. Armanfard, and D. Hatzinakos, "Feature selection for nonstationary data: application to human recognition using medical biometrics," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1446–1459, 2018.
- [2] P. García-Díaz, I. Sánchez-Berriel, J. A. Martínez-Rojas, and A. M. Díez-Pascual, "Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-seq data," *Genomics*, vol. 112, no. 2, pp. 1916–1925, 2020.
- [3] S. L. Marie-Sainte and S. Ghouzali, "Multi-objective particle swarm optimization-based feature selection for face recognition," *Studies in Informatics and Control*, vol. 29, no. 1, pp. 99–109, 2020.
- [4] Z.-Z. Liu, J.-W. Huang, Y. Wang, and D.-S. Cao, "ECoFFeS: a software using evolutionary computation for feature selection in drug discovery," *IEEE Access*, vol. 6, pp. 20950–20963, 2018.
- [5] H. Hanchuan Peng, F. Fuhui Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [6] H. Xia, J. Zhuang, and D. Yu, "Multi-objective unsupervised feature selection algorithm utilizing redundancy measure and negative epsilon-dominance for fault diagnosis," *Neurocomputing*, vol. 146, no. 25, pp. 113–124, 2014.
- [7] S. Paul and S. Das, "Simultaneous feature selection and weighting—an evolutionary multi-objective optimization approach," *Pattern Recognition Letters*, vol. 65, no. 1, pp. 51–59, 2015.
- [8] K. Gulsah, G. Stefano, S. A. Damla, and T. Riccardo, "Identifying (quasi) equally informative subsets in feature selection problems for classification: a max-relevance min-redundancy approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1424–1437, 2016.
- [9] H. Li, F. He, Y. Liang, and Q. Quan, "A dividing-based many-objective evolutionary algorithm for large-scale feature selection," *Soft Computing*, vol. 24, no. 9, pp. 1–31, 2019.
- [10] Z.-M. Gu and G.-G. Wang, "Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization," *Future Generation Computer Systems*, vol. 107, pp. 49–69, 2020.
- [11] Q. Lin, S. Liu, K.-C. Wong et al., "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, 2019.
- [12] W. L. Wang, W. Li, and Y. L. Wang, "An opposition-based evolutionary algorithm for many-objective optimization with adaptive clustering mechanism," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 5126239, 27 pages, 2019.
- [13] Q. Wang, L. Zhang, S. Wei, and B. Li, "Tensor decomposition-based alternate sub-population evolution for large-scale many-objective optimization," *Information Sciences*, vol. 569, pp. 376–399, 2021.
- [14] X. Y. Zhang, Y. Tian, R. Cheng, and Y. C. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 99, pp. 97–112, 2018.
- [15] Z. A. Yin, G. G. Wang, K. Q. Li, W. C. Yeh, M. W. Jian, and J. Y. Dong, "Enhancing MOEA/D with information feedback models for large-scale many-objective optimization," *Information Sciences*, vol. 522, pp. 1–16, 2020.
- [16] D. Hadka and P. Reed, "Borg: an auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.
- [17] C. Zhou, G. M. Dai, and M. C. Wang, "Enhanced  $\theta$  dominance and density selection based evolutionary algorithm for many-objective optimization problems," *Applied Intelligence*, vol. 48, no. 1, pp. 992–1012, 2018.
- [18] X. F. Xiufen Zou, Y. Yu Chen, M. Z. Minzhong Liu, and L. S. Lishan Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1402–1412, 2008.
- [19] J. C. Yuan and H. L. Liu, "A new dominance relation based on simplex for many objective optimization problems," in *Proceedings of the 2016 12th International Conference on Computational Intelligence and Security (CIS)*, pp. 175–178, Wuxi, China, December 2016.
- [20] J. K. Chong and K. C. Tan, "A novel grid-based differential evolution (DE) algorithm for many-objective optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2776–2783, Vancouver, Canada, July 2016.
- [21] X. Cai, Y. Xiao, M. Li, H. Hu, H. Ishibuchi, and X. Li, "A grid-based inverted generational distance for multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 21–34, 2021.
- [22] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.
- [23] S. Rostami and F. Neri, "A fast hypervolume driven selection mechanism for many-objective optimisation problems," *Swarm and Evolutionary Computation*, vol. 34, no. 1, pp. 50–67, 2016.
- [24] Y. N. Sun, C. C. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2018.
- [25] Z. Liang, T. Luo, K. Hu, X. Ma, and Z. Zhu, "An indicator-based many-objective evolutionary algorithm with boundary protection," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–14, 2020.
- [26] Y. Liu, J. Liu, T. Li, and Q. Li, "An R2 indicator and weight vector-based evolutionary algorithm for multi-objective optimization," *Soft Computing*, vol. 24, no. 7, pp. 5079–5100, 2019.
- [27] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [28] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016.
- [29] C. Zhao, Y. Zhou, and Z. Chen, "Decomposition-based evolutionary algorithm with automatic estimation to handle many-objective optimization problem," *Information Sciences*, vol. 546, pp. 1030–1046, 2021.
- [30] C. Dai, X. Lei, and X. Q. He, "A decomposition-based evolutionary algorithm with adaptive weight adjustment for many-objective problems," *Soft Computing*, vol. 24, no. 1, pp. 10587–10609, 2020.
- [31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with

- box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [32] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
- [33] F. Gu and Y.-M. Cheung, “Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 211–225, 2018.
- [34] M.-G. Dong, B. Liu, and C. Jing, “A many-objective evolutionary algorithm based on decomposition with dynamic resource allocation for irregular optimization,” *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 8, pp. 1171–1190, 2020.
- [35] L. Li, G. G. Yen, A. Sahoo, L. Chang, and T. Gu, “On the estimation of pareto front and dimensional similarity in many-objective evolutionary algorithm,” *Information Sciences*, vol. 563, pp. 375–400, 2021.
- [36] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, “AnD: a many-objective evolutionary algorithm with angle-based selection and shift-based density estimation,” *Information Sciences*, vol. 509, pp. 400–419, 2020.
- [37] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Vol. 31, Morgan Kaufmann, Burlington, MA, USA, 2011.
- [38] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, “A vector angle-based evolutionary algorithm for unconstrained many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.
- [39] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, “Mutation operators based on variable grouping for multi-objective large-scale optimization,” in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece, December 2016.
- [40] H. Zille, *Large-scale multi-objective optimisation: new approaches and a classification of the state-of-the-art*, Ph.D thesis, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany, 2019.
- [41] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [42] C. Ran, Y. C. Jin, M. Olhofer, and B. Sendhoff, “Test problems for large-scale multiobjective and many-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2017.
- [43] K. Bache and M. Lichman, *UCI Machine Learning Repository*, <http://archive.ics.uci.edu/ml/>, 2019.
- [44] T. M. Hamdani, J. M. Won, and A. M. Alimi, “Multi-objective feature selection with NSGA-II,” in *Adaptive and Natural Computing Algorithms. ICANNGA 2007*, vol. 4431, pp. 240–247, Springer, Berlin, Germany, 2009.
- [45] Y. Zhou, Y. Xiang, Z. Chen, J. He, and J. Wang, “A scalar projection and angle-based evolutionary algorithm for many-objective optimization problems,” *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2073–2084, 2019.
- [46] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [47] G. Chen and J. Li, “A diversity ranking based evolutionary algorithm for multi-objective and many-objective optimization,” *Swarm and Evolutionary Computation*, vol. 48, pp. 274–287, 2019.
- [48] L. Ben Said, S. Bechikh, and K. Ghedira, “The r-dominance: a new dominance relation for interactive evolutionary multi-criteria decision making,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 801–818, 2010.
- [49] J. Alcalá-Fdez, L. Sánchez, S. García et al., “KEEL: a software tool to assess evolutionary algorithms for data mining problems,” *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2008.

## Research Article

# Discriminative Codebook Hashing for Supervised Video Retrieval

Xiaoman Bian <sup>1</sup>, Rushi Lan <sup>1</sup>, Xiaoqin Wang <sup>1</sup>, Chen Chen <sup>1</sup>, Zhenbing Liu <sup>1</sup>,  
Xiaonan Luo <sup>1</sup> and Kuei-Kuei Lai <sup>2</sup>

<sup>1</sup>Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China

<sup>2</sup>Department of Business Administration, Chaoyang University of Technology, Taichung 413310, Taiwan, China

Correspondence should be addressed to Xiaoqin Wang; xqwang@guet.edu.cn and Kuei-Kuei Lai; laik.tw@gmail.com

Received 18 May 2021; Accepted 12 August 2021; Published 25 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Xiaoman Bian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, hashing learning has received increasing attention in supervised video retrieval. However, most existing supervised video hashing approaches design hash functions based on pairwise similarity or triple relationships and focus on local information, which results in low retrieval accuracy. In this work, we propose a novel supervised framework called discriminative codebook hashing (DCH) for large-scale video retrieval. The proposed DCH encourages samples within the same category to converge to the same code word and maximizes the mutual distances among different categories. Specifically, we first propose the discriminative codebook via a predefined distance among intercode words and Bernoulli distributions to handle each hash bit. Then, we use the composite Kullback–Leibler (KL) divergence to align the neighborhood structures between the high-dimensional space and the Hamming space. The proposed DCH is optimized via the gradient descent algorithm. Experimental results on three widely used video datasets verify that our proposed DCH performs better than several state-of-the-art methods.

## 1. Introduction

Under the condition of the increase in smartphones, the amount of video data has shown an explosive growth trend [1–3]. For example, TikTok has over 400 million daily active users who upload approximately 2,000 videos every minute. YouTube receives a total of 100 hours of videos per minute [4–6]. Due to the economic storage and efficiency of binary codes, hash-based methods have been widely applied to visual retrieval tasks [7–13].

Previous hash-related work [14] mainly focused on image hashing and can be divided into data-independent and data-dependent methods. Data-independent approaches learn binary codes without data information but through random space projection. The most representative algorithm is local sensitive hashing (LSH) [15], which generates huge redundant information using random mapping and obtains satisfactory performance with long hash codes. Data-dependent hash methods [16–18], which can also be divided into unsupervised hashing and supervised hashing, are proposed to generate more efficient hash

codes by maintaining the neighborhood structure between data. For example, Gong et al. [19] proposed iterative quantization hashing (ITQ), which minimizes quantization error by rotating principal component analysis (PCA) projection data. Spectral hashing (SH) [20] assumes that data obey a uniform distribution and divides the data according to the main direction of the data stream. Density sensitive hashing (DSH) [21] extends LSH by studying structural information. Zhang et al. [22] developed a convergence-preserving parametric learning algorithm, called latent factor hashing (LFH), to learn similarity-preserving binary codes based on latent factor models. Liu et al. [23] proposed kernel supervised hashing (KSH) by applying kernel-based formulas to accommodate linearly inseparable data and designed a greedy algorithm to solve the hash function optimization problem.

In recent years, hashing methods proposed for video retrieval have also received extensive attention [24–31] and are composed of two categories: machine learning methods and deep hashing. Machine learning methods, resembling image hashing approaches, learn binary codes of video

keyframes based on the low-level manual features and then calculate video hashing codes via averaging. Wu et al. [4] employed video hashing via using color histograms to obtain global features. This is the first application of hash learning in the video field. Multiple-feature hashing (MFH) [32] adopts the weight-based method to combine different features. Ye et al. [33] used video structural information in the supervised learning paradigm to obtain the optimal binary codes. Stochastic multiview hashing (SMVH) [34] attempts to separately calculate the probability similarity matrices of video frames in the feature space and the Hamming space, and then, the difference between the above two probability matrices is minimized using the KL divergence. Nie et al. [35] defined joint multiview hashing (JMVH) by maximizing the interclass distance and minimizing the innerclass distance to preserve the global structure and local structure with multiple features. Boosting temporal video hashing (BTVH) [36] studies the multitask learning problem to boost the performance and captures the inherent similarity of video from both visual and temporal perspectives. In addition, some researchers in recent years have used deep networks to obtain the temporal and spatial information between keyframes. For instance, central similarity quantization (CSQ) [37] learns the temporal information by using 3D convolutional neural networks and proposes a view point called hash center to enhance the central similarity.

However, most existing video hashing approaches may lead to the following problems. (1) Low discriminability among different categories: functions based on pairwise similarity or triple relationships only consider local information, which results in good maintenance of the information of similar samples but shows poor performance in distinguishing samples from different categories. (2) Poor performance in real-world scenarios: in real application scenarios, similar data often accounts for only a small proportion, and most samples are not similar, which leads to low efficiency when the data are imbalanced [37]. (3) Greater time costs on deep learning: deep learning frameworks are time-consuming when training models and have no significant performance based on the spatiotemporal information extracted by the network. Hence, these video hashing functions cannot learn discriminative hash codes to enhance the performance.

To solve the above problems, in this work, we propose a novel framework for supervised video retrieval, called discriminative codebook hashing, which considers the global structure to construct the hash function. DCH encourages samples within the same category to converge to the identical codeword and maximizes the mutual distances between different categories. Specifically, the discriminative codebook is first generated based on two characters: the predefined distance between intercode words and Bernoulli distributions for ensuring that each hash bit stores more information. Then, to keep the similarity matrix between the feature space and the Hamming space, the composite KL divergence is proposed to solve this problem. Finally, the gradient descent algorithm is utilized to optimize the algorithm. In this way, we can obtain discriminative binary codes for video retrieval. Figure 1 shows the framework of

DCH, and the method we proposed has the following innovations:

- (i) We proposed the discriminative codebook based on the predefined distance between intercode words and Bernoulli distributions for ensuring each hash bit to store more information
- (ii) The DCH method, which can maximize the distance of the intercode words generated by the predefined codebook to learn discriminative binary codes for supervised video retrieval, is proposed
- (iii) We verify our proposed method by experimenting on three widely used datasets, which shows that DCH has a significant improvement in contrast with several state-of-the-art methods

The other sections are organized as follows. Section 2 introduces some preliminary works. Section 3 introduces the proposed discriminative codebook hashing in detail. The experimental work is presented in Section 4, and the conclusion of DCH is shown in Section 5.

## 2. Preliminary Work

In this section, we briefly introduce the preliminary work, namely, stochastic multiview hashing [34]. It is a supervised video retrieval method that aims to preserve the similarity structure from the original space to the Hamming space.

Let  $V = \{v_i\}_{i=1}^{n_v}$  be the video set, where  $v_i$  indicates the  $i$ th video of  $V$  and  $n_v$  is the number of videos.  $H = \{h_i\}_{i=1}^{n_v}$  is hash code of the video set, where  $h_i \in \{0, 1\}$  is  $l$ -bit length binary codes transformed by  $v_i$ . The video features are extracted based on the set of keyframe features  $X = \{x_i\}_{i=1}^n$ , where  $x_i \in \mathbb{R}^{1 \times d}$ ,  $n$  is the number of keyframes, and  $d$  is the dimension of each keyframe.  $Z = \{z_i\}_{i=1}^n$  represents the corresponding binary codes of the keyframes, where  $z_i \in \mathbb{R}^{1 \times l}$ . The conversion relationships between the above variables are formulated as

$$\tilde{Z} = XW + b, \quad (1)$$

$$Z = \text{sigmoid}(\tilde{Z}), \quad (2)$$

$$h_i = T\left(\frac{1}{|\text{Ind}_i|} \sum_{j \in \text{Ind}_i} z_j\right), \quad (3)$$

where  $\tilde{Z} \in \mathbb{R}^{n \times l}$  is the temporal result of linear projection,  $b \in \mathbb{R}^l$  is a bias parameter,  $W \in \mathbb{R}^{d \times l}$  is the projection matrix,  $\text{Ind}_i$  is the set of frames, and  $|\text{Ind}_i|$  is the sum of samples in the set. The high-dimensional keyframe feature matrix  $X$  is first projected into the lower matrix  $\tilde{Z}$ . Then, the sigmoid function is used to map the variable between 0 and 1. Finally, a thresholding function is used to change the data into a binary code with  $T(y) = 0$  if  $y < 0.5$  and  $T(y) = 1$ , otherwise.

SMVH keeps the similarity matrix between the feature space and the Hamming space using a composite KL divergence measure. In particular, it separately calculated the similarity probability matrix  $P$  in the original space and the

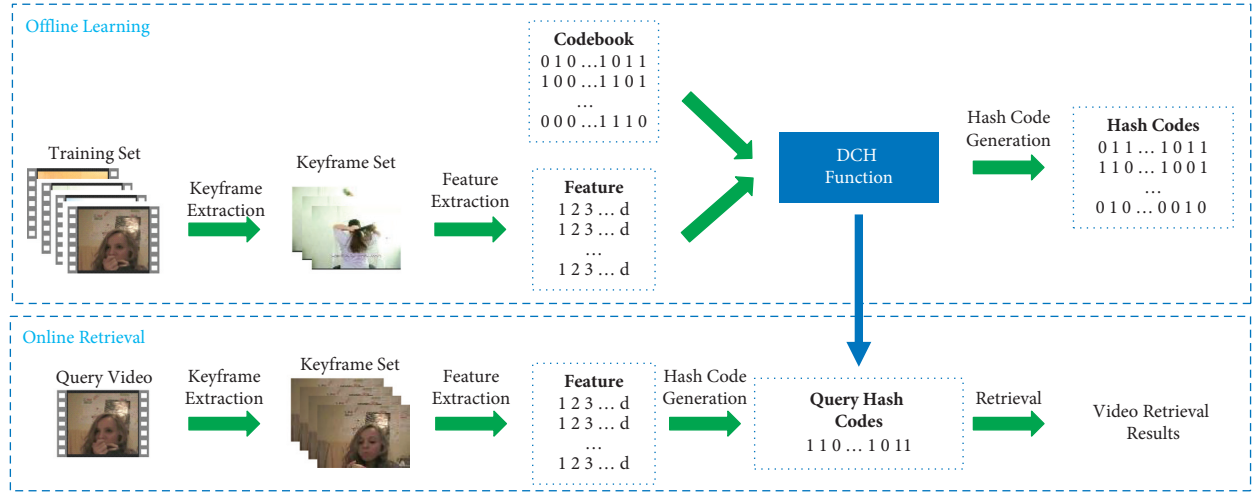


FIGURE 1: The framework of DCH. We divide the entire experiment into two steps, namely, offline learning and online retrieval. In the offline phase, we join keyframe features and predefined codebook to learn hash functions. In the online phase, we map the query video into a set of binary codes through hash functions. Next, we use the exclusive or (XOR) operation to obtain the Hamming distance between the query video and samples in the database. Finally, we take videos with the shortest Hamming distance as the video retrieval results.

pairwise similarity matrix  $Q$  among samples in the Hamming space. Then, the KL divergence is used to examine how well the above two probability matrices  $P$  and  $Q$  match. Therefore, the objective function of SMVH is defined as follows:

$$\min_{W,b} S_{\text{KL}}(W, b) + \frac{\mu}{2} \|W\|_F^2, \quad (4)$$

where  $\mu > 0$  controls the weight of the regular term to prevent overfitting and  $S_{\text{KL}}(W, b)$  is the composite KL divergence. The latter can be represented as

$$S_{\text{KL}}(W, b) = \lambda \text{KL}(P \| Q) + (1 - \lambda) \text{KL}(Q \| P), \quad (5)$$

where  $0 \leq \lambda \leq 1$  controls the influence of the composite KL divergence,  $P = \{p_{ij}\}_{i=1}^n \in \mathbb{R}^{n \times n}$  is the similarity structure based on  $X$ , and  $Q = \{q_{ij}\}_{i=1}^n \in \mathbb{R}^{n \times n}$  is another probability matrix preserving the similarity information of  $Z$  in the Hamming space. In addition, the KL divergence is defined as follows:

$$\text{KL}(P \| Q) = \sum_{i=1}^n \sum_{j \neq i} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}, \quad (6)$$

where  $p_{j|i}$  is a conditional probability that reflects the similarity between  $x_i$  and  $x_j$ , and another conditional probability  $q_{j|i}$  represents the probability of returning  $z_j$  given the query  $z_i$ .

### 3. Discriminative Codebook Hashing

In this section, we present the proposed DCH in detail through four parts, including the proposed discriminative codebook, the objective function, algorithmic optimization, and complexity analysis.

**3.1. Discriminative Codebook.** Motivated by CSQ [37], we propose a novel and discriminative codebook  $C = \{c_i\}_{i=1}^m$  for supervised video retrieval, where  $c_i \in \{0, 1\}^{1 \times l}$  is the code word of the  $i$ th category. The proposed codebook is defined according to two characters. The first is that the value in the same bit of different code words obeys a Bernoulli distribution. Specifically, the proportions of 0 and 1 of the same bit in different categories are both 50%, that is,  $c_i$  has a 50% probability of being 0 or 1, which will maximize the entropy and store more information in each bit. The other is that the mutual distances among intercode words are defined as follows:

$$D_H(c_i, c_j) \geq \frac{l}{2} - f, \quad (7)$$

where  $D_H$  is the Hamming distance between code words  $c_i$  and  $c_j$ ,  $l$  is the length of binary codes, and  $f$  represents the fault tolerance. The mutual distance between intercode words will be the largest constrained by equation (7).

Overall, the proposed codebook encourages samples within the same category to converge to the same codeword and maximizes the mutual distance between different categories. Therefore, the proposed codebook can preserve global structures and help generate discriminative binary codes for video retrieval. The scheme of the proposed discriminative codebook is presented in Algorithm 1.

**3.2. Objective Function.** According to the proposed discriminative codebook  $C$ , we expand each row of the codebook matrix  $C$  into  $R = \{r_i\}_{i=1}^n$  according to the number of samples, where  $r_i \in \mathbb{R}^{1 \times l}$ . The detailed generation process of  $R$  is shown in Algorithm 2. We minimize the error between the binary codes and the predefined codebook as

**Input:** the number of categories  $m$ ; the number of samples per category  $n_i$ ; code length  $l$ ; maximum number of iterations  $T_c$ ; fault tolerance rate  $f$ .

**Output:** codebook  $C \in \mathbb{R}^{m \times l}$

- (1) **for** iteration  $t_c = 1 : T_c$
- (2)   **for** category  $i = 1 : m$
- (3)      $c_{.i}[\text{random half coordinate}] = 1$
- (4)      $c_{.i}[\text{the rest coordinate}] = 0$
- (5)   **end**
- (6)   **if** any two rows of  $C$  satisfy equation (7)
- (7)     **break**
- (8)   **end**
- (9) **end**

ALGORITHM 1: Discriminative codebook.

**Input:** training data  $X \in \mathbb{R}^{n \times d}$ ; codebook  $C \in \mathbb{R}^{m \times l}$ ; maximum number of iterations  $T$ ; code length  $l$ ; parameters  $\lambda, \mu, \gamma$ ; learning rate  $\alpha$ ;

**Output:** hash codes  $H \in \{0, 1\}^{n \times l}$ .

- (1) **Initialization:** initialize the projection matrix  $W$  and bias matrix  $b$  as a random matrix and vector.
- (2) **Generating  $R$  according to the number of samples:**
- (3) **for** category  $i = 1 : m$
- (4)    $R = [R; \text{repmat}(C(i, :), n_i, 1)]$
- (5) **end**
- (6) **Gradient descent:**
- (7) **for** iteration  $i = 1 : T$
- (8)   **W-Step:**  $W^{(i+1)} = W^{(i)} + \alpha dW$
- (9)   **b-Step:**  $b^{(i+1)} = b^{(i)} + \alpha db$
- (10) **end**
- (11) **Video binary code computation:** video hash codes are obtained by equations (1)–(3).

ALGORITHM 2: Discriminative codebook hashing.

$$\min_{W, b} \|Z - R\|_F^2. \quad (8)$$

Specifically, for each  $z_i \in Z$ , we take  $r_i$  as the codebook of  $z_i \in Z$  to make samples in the same category share the same codebook and samples in different categories have discriminative binary codes.

To keep the similarity matrix between the feature space and the Hamming space, we join the composite KL divergence and our proposed codebook to construct the overall objective function of DCH as follows:

$$\min_{W, b} S_{\text{KL}}(W, b) + \frac{\gamma}{2} \|Z - R\|_F^2 + \frac{\mu}{2} \|W\|_F^2, \quad (9)$$

where  $\gamma$  controls the weight of the error loss between the codebook and the learned hash codes, and the second term of equation (9) aligns values between binary codes and their corresponding code word.

In this way, our proposed DCH can solve the problem that other algorithms only consider the pairwise relationships and ensure that samples in the same category share the

same code word. Furthermore, DCH maximizes the mutual distances between different categories and then obtains discriminative binary codes.

**3.3. Algorithmic Optimization.** The optimization problem has two main variables:  $W$  and  $b$ . Our solution is to use the gradient descent algorithm to find good solutions. To facilitate the writing, we split the objective function equation (9) into three parts:

$$\begin{aligned} \Phi_1(W, b) &= S_{\text{KL}}(W, b), \\ \Phi_2(W, b) &= \frac{\gamma}{2} \|Z - R\|_F^2, \\ \Phi_3(W) &= \frac{\mu}{2} \|W\|_F^2. \end{aligned} \quad (10)$$

The detailed optimization procedure is presented as follows.

**W-Step:** the corresponding problem is to minimize the following loss function:



$$\min_W S_{KL}(W, b) + \frac{\gamma}{2} \|Z - R\|_F^2 + \frac{\mu}{2} \|W\|_F^2. \quad (11)$$

To compute the optimal  $W$ , the relevant deviation formula can be expressed as

$$dW = \frac{\partial\Phi_1(W, b)}{\partial W} + \frac{\partial\Phi_2(W, b)}{\partial W} + \frac{\partial\Phi_3(W)}{\partial W}. \quad (12)$$

The derivative of  $\partial\Phi_1(W, b)$  w.r.t.  $W$  can be computed as follows:

$$\frac{\partial\Phi_1(W, b)}{\partial W} = \left[ \frac{\partial\Phi_1(W, b)}{\partial z_{ik}} \frac{\partial z_{ik}}{\partial w_{kj}} \right]_{d \times l}, \quad (13)$$

where  $\partial\Phi_1(W, b)/\partial z_{ik}$  and  $\partial z_{ik}/\partial w_{kj}$  are represented as

$$\begin{aligned} \frac{\partial\Phi_1(W, b)}{\partial z_{ik}} &= 2(\lambda(p_{it} - q_{it} + p_{ti} - q_{ti}) + (1 - \lambda)) * \left( q_{ti} \sum_{g \neq i} q_{gli} \log \frac{q_{gli}}{p_{gli}} + q_{it} \sum_{g \neq t} q_{glt} \log \frac{q_{glt}}{p_{glt}} - \log \frac{q_{ti}}{p_{ti}} - \log \frac{q_{it}}{p_{it}} \right) (z_{ik} - z_{tk}), \\ \frac{\partial z_{ik}}{\partial w_{kj}} &= z_{ik} (1 - z_{ik}) x_{ji}. \end{aligned} \quad (14)$$

Following the norm derivation law,  $\partial\Phi_2(W, b)/\partial W$  can be optimized as follows:

$$\frac{\partial\Phi_2(W, b)}{\partial W} = \frac{\partial\Phi_2(W, b)}{\partial Z} \frac{\partial Z}{\partial W} = X^T ((Z - R) \odot (Z \odot (1 - Z))), \quad (15)$$

where  $\odot$  indicates that the elements in the same position of two matrices are multiplied.

For  $\partial\Phi_3(W)/\partial W$ , we have the derivative that

$$\frac{\partial\Phi_3(W)}{\partial W} = \mu W. \quad (16)$$

**b-Step:** the subproblem of  $b$  is given by

$$\min_b S_{KL}(W, b) + \frac{\gamma}{2} \|Z - R\|_F^2. \quad (17)$$

The deviation w.r.t.  $b$  can be expressed as

$$db = \frac{\partial\Phi_1(W, b)}{\partial b} + \frac{\partial\Phi_2(W, b)}{\partial b}. \quad (18)$$

The derivative of  $\partial\Phi_1(W, b)/\partial b$  is described as follows:

$$\frac{\partial\Phi_1(W, b)}{\partial b} = \left[ \frac{\partial\Phi_1(W, b)}{\partial z_{ik}} \frac{\partial z_{ik}}{\partial b_k} \right]_{1 \times l}, \quad (19)$$

where

$$\frac{\partial z_{ik}}{\partial b_k} = z_{ik} (1 - z_{ik}). \quad (20)$$

The second term of equation (18) is described as follows:

$$\frac{\partial\Phi_2(W, b)}{\partial b} = \frac{\partial\Phi_2(W, b)}{\partial Z} \frac{\partial Z}{\partial b} = (Z - R) \odot (Z \odot (1 - Z)). \quad (21)$$

Algorithm 2 describes the overall algorithm optimization process of the proposed DCH.

**3.4. Complexity Analysis.** The time complexity of the entire training process of SMVH [34] is approximately  $O(Tn^3 + n^2)$ , and the proposed DCH algorithm adds two parts time-consuming on this basis. The first part is the learning process of  $C$ , and the time complexity is  $O(T_c l)$ . The second part is that the time complexity of optimizing equations (15) and (21) together is  $O(dnl)$  in each iteration. Therefore, the overall time complexity of DCH is  $O(n^2 + T_c l + T(n^3 + dnl))$ . In this work, time complexities  $O(T_c l)$  and  $O(dnl)$  can be ignored due to  $T_c, l, d \ll n$  so that our complexity is nearly  $O(Tn^3 + n^2)$ . Additionally, the calculation of the hash codes is a linear projection with a time complexity of approximately  $O(1)$ , and the online search can be performed by XOR operations. Although the algorithm proposed in this paper adds a constraint on SMVH, the maximum number of iterations  $T$  directly affects the time complexity of the algorithm. It can be proven in subsequent experiments that DCH can converge in fewer iterations. Thus, the time complexity of DCH is in a reasonable range.

## 4. Experiments

In this section, we first introduce the datasets used in this paper, and then, the baselines and some experimental details will be introduced. Finally, we present the experimental results.

**4.1. Datasets.** CC\_WEB\_VIDEO [4] is the most useful dataset in near-duplicate video retrieval (NDVR) research, which contains data from YouTube, Google, and Yahoo. There are 12,877 videos that are divided into 24 sets, and keyframes are extracted by a uniform sampling method to represent the video. Since some videos do not have label information, we take 3,482 videos with labels as the experimental dataset. In each category, we select 70% of the video data as the training set and the remainder as the testing set. We extract 10 keyframes for each video uniformly and

extract 4096-dimensional features to represent keyframes by using the pretrained VGG-19 network.

**HMDB51** [38] contains 6,766 human action videos selected from movies and some other public sources such as YouTube. The dataset is divided into 51 categories, and each of them includes approximately 100 clips. In each category, we randomly select 45 video samples. Of these, 25 videos are added to the training set and the rest are select to the testing set. We uniformly extract 10 keyframes for each video, and the VGG-19 pretraining network is used to extract the 4096-dimensional deep features.

**UCF101** [39] contains 13,320 videos which has been divided into 101 human behavior categories, such as sports, instruments, character interactions, and others used for action recognition. We randomly select 70 videos in each category to join the training set, and 30 videos to join the testing set. For each video, 10 keyframes are uniformly selected to represent the video. We use VGG-19 to extract the 4096-dimensional features for each keyframe.

## 4.2. Experimental Setting

**4.2.1. Baselines.** Several state-of-the-art hash functions, including ITQ [19], SH [20], DSH [21], LFH [22], KSH [23], JMVH [35], and SMVH [34], are used for comparison. Among these methods, ITQ, SH, and DSH are unsupervised hashing methods, while LFH, KSH, JMVH, and SMVH are supervised hashing methods. For the comparative test, we use the source codes published to conduct the experiment. JMVH and SMVH can also be used for multiview video retrieval, but in this paper, we only test these methods as a single view method. It is worth noting that all the experimental results are obtained in MATLAB R2016a on the same computer with an Intel Core i7-6700 CPU @ 3.40 GHz, 72 GB RAM and the 64 bit Windows 10 operating system.

**4.2.2. Evaluation Metrics.** We use four popular evaluation metrics to comprehensively evaluate experimental results. The mean average precision (mAP) is widely used in the retrieval field. The higher the mAP score is, the better the retrieval performance of the method is. The precision@K curve represents the precision accuracy versus the first  $K$  retrieved samples, where precision represents the proportion of the number of retrieved correct videos to the total number of retrieved videos. The recall@K curve represents the average recall rate versus the first  $K$  retrieved samples, where recall represents the proportion of the correct video volume retrieved in all near-duplicate video samples. The precision-recall (PR) curve is an index used to evaluate reliability and is widely used in the fields of medicine and machine learning.

**4.2.3. Parameter Selection.** We have three model parameters, including  $\lambda$ ,  $\mu$ , and  $\gamma$ , and the number of iterations  $T$ . According to SMVH [34], we set  $\lambda = 0.9$  and  $\mu = 0.01$ .

As shown in Figure 2(a), when  $\gamma$  is in the range of 0.05 to 1, the results are stable across three different datasets. Therefore, we empirically choose  $\gamma = 1$  in our proposed model. The maximum number of iterations  $T$  determines the training time cost and the performance, so it is worth discussing. Figure 2(b) shows the effect of the maximum iterations  $T$  in the range of 100 to 1400 on mAP performance. For HMDB51, it can be seen that the best mAP is generated with  $T = 800$  before decreasing. However, in the other two datasets,  $T = 800$  is not an optimal experimental result. Therefore, after comprehensive consideration,  $T = 1000$  is set as the final parameter setting.

**4.3. Results and Discussion.** Table 1 shows the mAP results for different lengths of hash codes on the three datasets, and the results of other evaluation metrics are shown in Figures 3–5. We will give the detailed analysis of all results of the three datasets in the following parts.

According to Table 1, for the CC\_WEB\_VIDEO dataset, the mAPs are very high because the dataset is movie clips, and videos of the same category are near-duplicate videos. As shown in Table 1, the performance of the proposed DCH is at least 1.85% better than that of the other methods from 32 to 64 bits. When the code length is 96 bits, the mAP of DCH is slightly lower than that of LFH. As shown in Figure 3, the experimental results of our method in precision@K and recall@K are equal to or slightly higher than those of most other methods. Besides, as the code length increases, the performance of our proposed DCH gradually surpasses that of other methods. Figures 3(i)–3(l) show that the area surrounded by DCH is gradually increasing.

Table 1 shows that our proposed DCH performs better than other hash methods in most cases in the HMDB51 dataset. Although the mAP performance of the JMVH method surpasses 2.39% over that of DCH with 32 bits, the mAPs of our proposed DCH are better than those of the other comparison methods in the subsequent experiments. Figure 4 shows that when the length of hash codes is larger than 32 bits, regardless of whether precision@K curve, recall@K curve, or PR curve is used, DCH has excellent performance compared with other methods in all metrics for the precision@K curve, recall@K curve, and PR curve.

For the UCF101 dataset, DCH obtained the optimal experimental results in the range of [32, 48, 64] bits. It is worth noting that the size of the UCF101 dataset is relatively large, and SMVH cannot obtain discriminative video hash when the hash code length is very small. Therefore, SMVH has no experimental results available for  $l = 32$  and  $l = 48$ . As shown in Figure 5, the performance of DCH is much higher than those of some of the methods except JMVH. We can see that the recall rate of DCH for positive samples is slightly lower than that of JMVH based on Figures 5(e)–5(h). Figures 5(i)–5(k) show that the performance of DCH for 32 to 48 bits is better than those of all other methods for the PR curve.

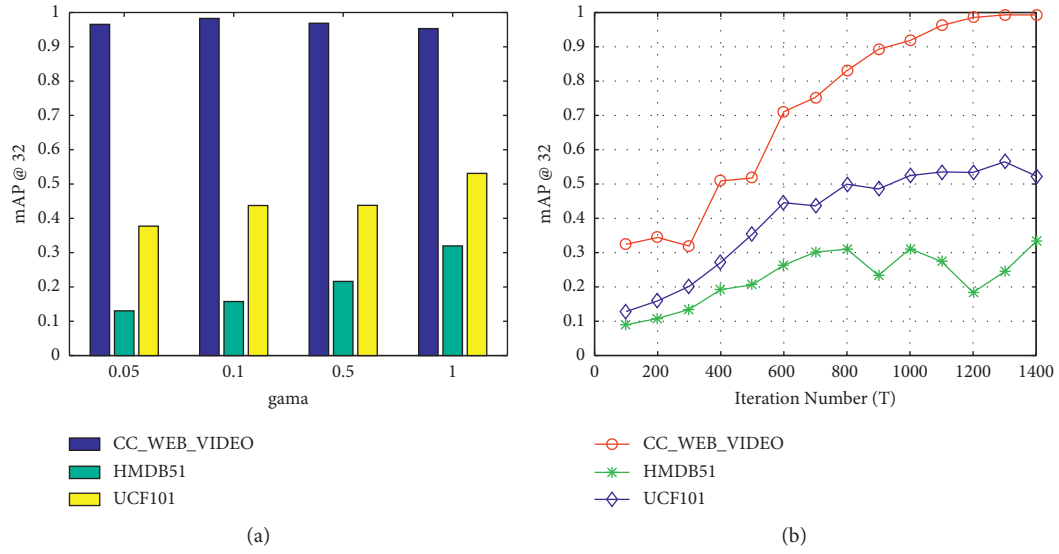


FIGURE 2: Parameter analysis on the CC\_WEB\_VIDEO, HMDB51, and UCF101 datasets. (a) mAP vs.  $\gamma$  (weight parameter  $\gamma$ ) and (b) mAP vs.  $T$  (iteration parameter  $T$ ).

TABLE 1: The mAP of different hash code lengths on three datasets, where the best experimental results are given in bold.

Method	CC_WEB_VIDEO				HMDB51				UCF101			
	32 bits	48 bits	64 bits	96 bits	32 bits	48 bits	64 bits	96 bits	32 bits	48 bits	64 bits	96 bits
ITQ [19]	0.6877	0.7725	0.8099	0.7700	0.0697	0.0749	0.0793	0.0885	0.1383	0.1620	0.1801	0.2119
SH [20]	0.6729	0.7026	0.6994	0.6708	0.0662	0.0657	0.0642	0.0653	0.1033	0.1138	0.1244	0.1395
DSH [21]	0.6510	0.7060	0.6929	0.8158	0.0505	0.0628	0.0671	0.0750	0.0720	0.0667	0.0815	0.1082
LFH [22]	0.8327	0.8088	0.9854	<b>0.9912</b>	0.0141	0.0208	0.0148	0.0225	0.0032	0.0038	0.0078	0.0113
KSH [23]	0.9368	0.9030	0.9477	0.8761	0.2470	0.2811	0.3054	0.3144	0.3222	0.3598	0.3972	0.4075
JMVH [35]	0.7842	0.5576	0.4335	0.3745	<b>0.2807</b>	0.3015	0.2418	0.1295	0.3941	0.5166	0.6007	<b>0.6875</b>
SMVH [34]	0.9346	0.9411	0.9543	0.7490	0.1212	0.1399	0.1374	0.0319	—	—	0.0094	0.0304
DCH	<b>0.9531</b>	<b>0.9763</b>	<b>0.9886</b>	0.9858	0.2568	<b>0.3819</b>	<b>0.3600</b>	<b>0.4150</b>	<b>0.5310</b>	<b>0.6137</b>	<b>0.6609</b>	0.6458

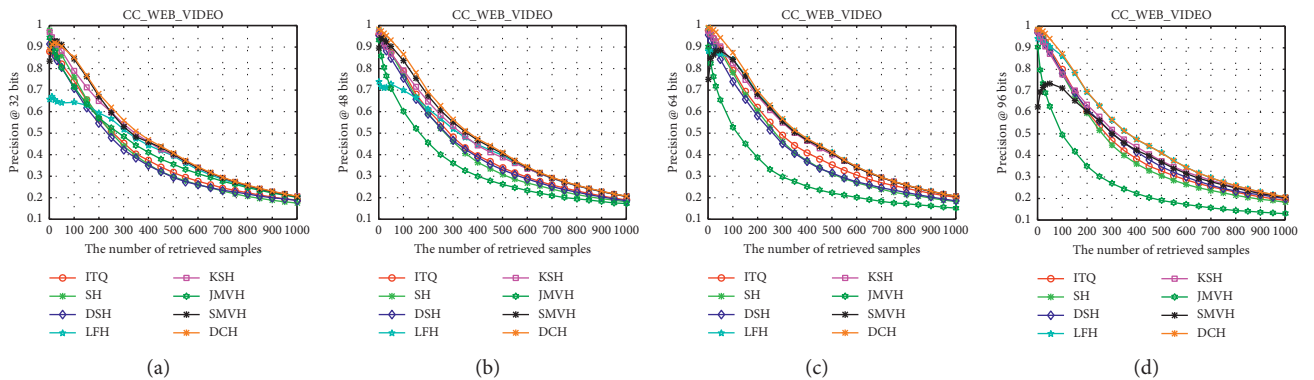


FIGURE 3: Continued.

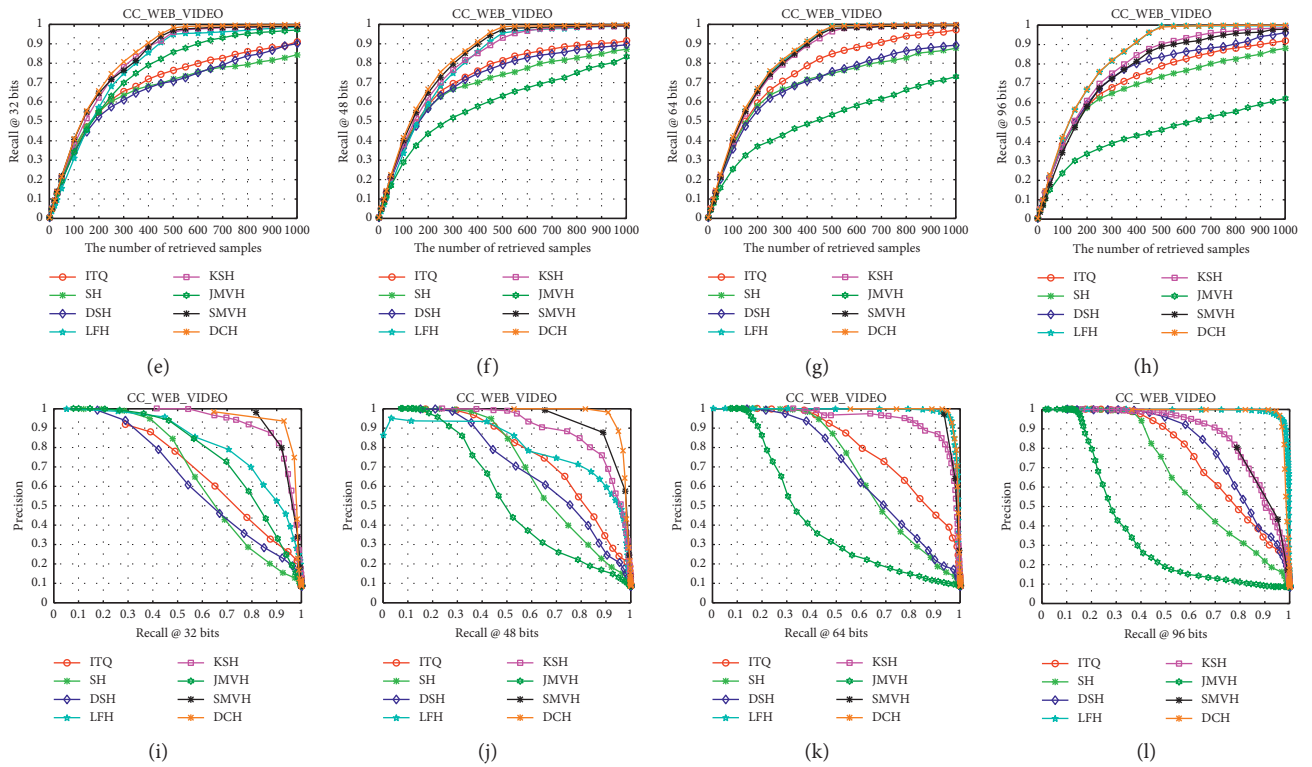


FIGURE 3: Precision@K (a-d), recall@K (e-h), and PR (i-l) curves on the CC\_WEB\_VIDEO dataset.

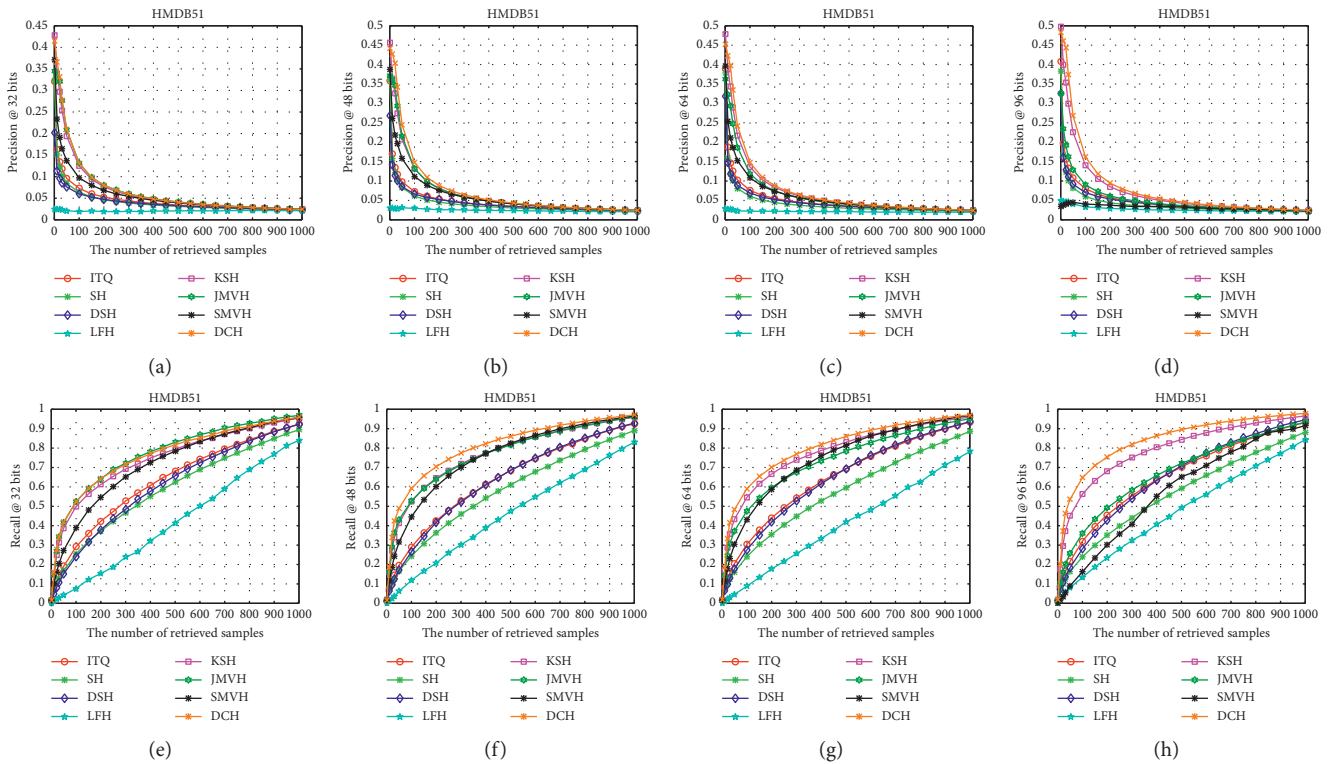


FIGURE 4: Continued.

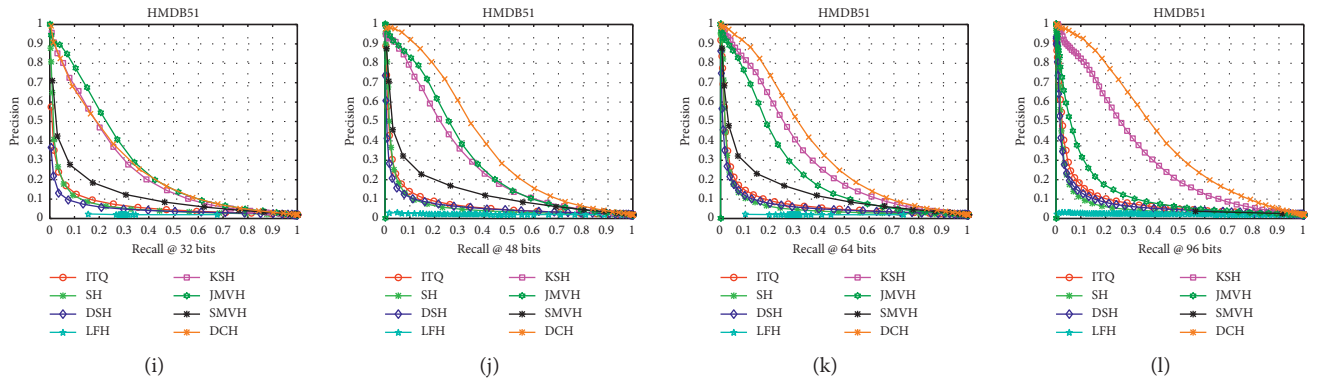


FIGURE 4: Precision@K (a–d), recall@K (e–h), and PR (i)–(l) curves on the HMDB51 dataset.

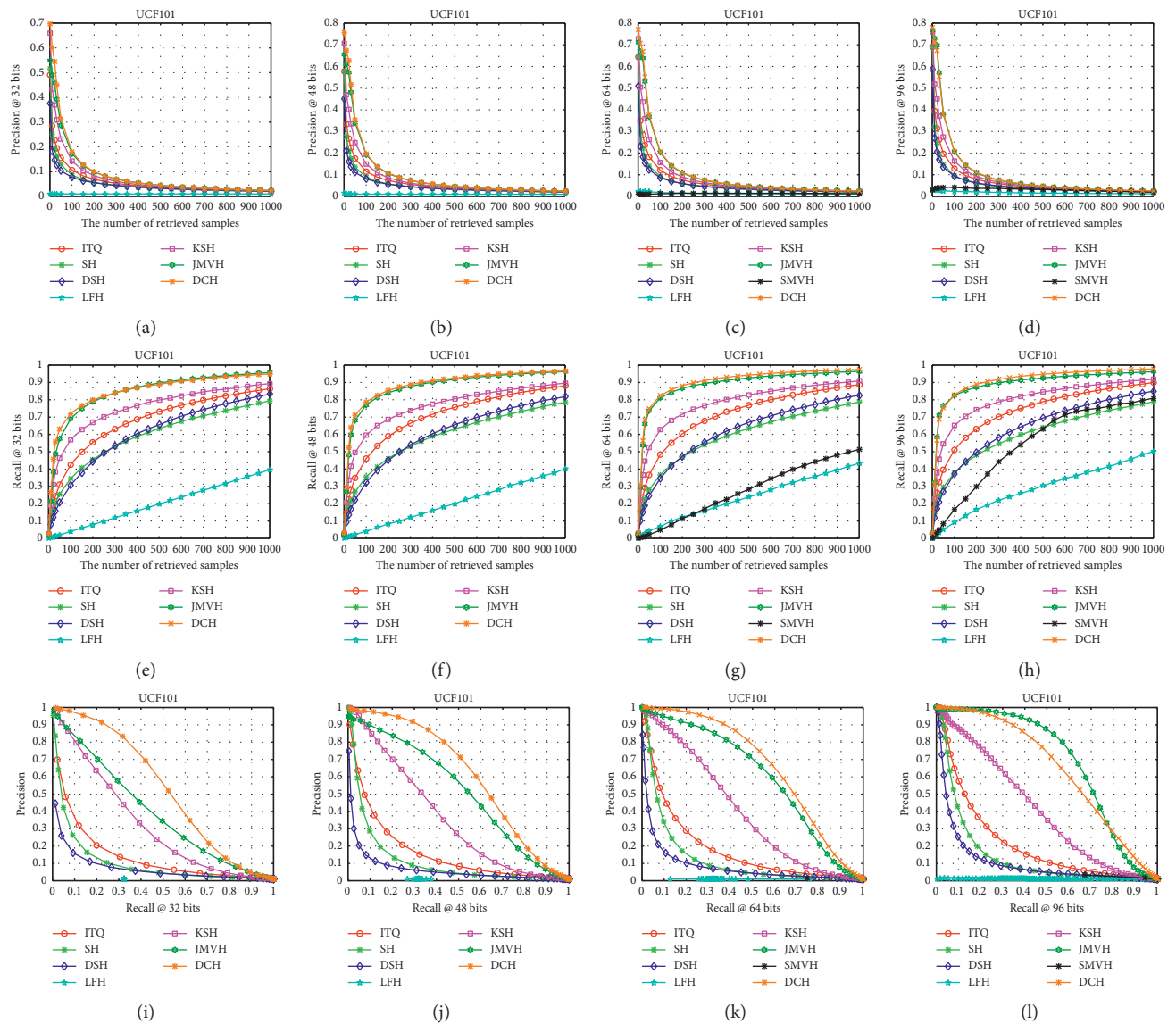


FIGURE 5: Precision@K (a–d), recall@K (e–h), and PR (i–l) curves on the UCF101 dataset.

## 5. Conclusion

In this paper, we propose a novel supervised video hashing framework, termed discriminative codebook hashing, which can generate discriminative binary codes for video retrieval. The proposed DCH encourages samples within the same category to converge to the same code word and maximizes the mutual distances between different categories. Specifically, we generate a discriminative codebook to distinguish between samples of different categories more accurately. Extensive experimental results prove that the performance of DCH is significantly improved compared to several state-of-the-art methods. In future work, we will use a smaller matrix storing the similarity information between samples to avoid consuming considerable training time and space when the amount of data is large. This will improve the performance of the model while reducing the time complexity.

## Data Availability

CC\_WEB\_VIDEO dataset can be downloaded from <http://vireo.cs.cityu.edu.hk/webvideo/>, the HMDB51 dataset can be downloaded from <https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#dataset>, and the UCF101 dataset can be downloaded from <https://www.crcv.ucf.edu/data/UCF101.php>.

## Conflicts of Interest

The authors declare that there are no conflicts of interest in the publication of this paper.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. 61902087, 61772149, 61936002, and 6202780103), Guangxi Science and Technology Project (nos. 2019GXNSFFA245014, AD18281079, AA18118039, and AD18216004), and Guangxi Key Laboratory of Image and Graphic Intelligent Processing (no. GIIP2001).

## References

- [1] X. Wu, C.-W. Ngo, A. G. Hauptmann, and H.-K. Tan, "Real-time near-duplicate elimination for web video search with content and context," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 196–207, 2009.
- [2] V. O. Maraghi and K. Faez, "Scaling human-object interaction recognition in the video through zero-shot learning," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9922697, 15 pages, 2021.
- [3] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.
- [4] X. Wu, A. G. Hauptmann, and C. Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th International Conference on Multimedia*, pp. 218–227, ACM, Bavaria, Germany, 2007.
- [5] Z. Lu, Y. Wang, Y. Li, X. Tong, C. Mu, and C. Yu, "Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 30, p. 1, 2021.
- [6] Y.-N. Ma, Y.-J. Gong, C.-F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: an ant colony algorithm incorporating alarm pheromone," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 141–154, 2019.
- [7] G.-H. Liu and Z. Wei, "Image retrieval using the fused perceptual color histogram," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8876480, 10 pages, 2020.
- [8] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, and X. Luo, "MaDNet: a fast and lightweight network for single-image super resolution," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1443–1453, 2021.
- [9] W. Li, Y. Zhang, Y. Sun et al., "Approximate nearest neighbor search on high dimensional data-experiments, analyses, and improvement," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1475–1488, 2020.
- [10] R. Lan, Y. Zhou, Z. Liu, and X. Luo, "Prior knowledge-based probabilistic collaborative representation for visual recognition," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1498–1508, 2020.
- [11] X. Wang, R. Lan, H. Wang, Z. Liu, and X. Luo, "Fine-grained correlation analysis for medical image retrieval," *Computers & Electrical Engineering*, vol. 90, Article ID 106992, 2021.
- [12] L. Shang, L. Yang, F. Wang, K. Chan, and X. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the 18th International Conference on Multimedia*, pp. 531–540, ACM, Firenze, Italy, 2010.
- [13] N. Q. Ly, T. K. Do, and B. X. Nguyen, "Large-scale coarse-to-fine object retrieval ontology and deep local multitask learning," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 1483294, 40 pages, 2019.
- [14] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th International Conference on Multimedia*, pp. 423–432, ACM, Scottsdale, AZ, USA, 2011.
- [15] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the 20th ACM Symposium on Computational Geometry*, pp. 253–262, ACM, Brooklyn, NY, USA, 2004.
- [16] W. Liu, J. Wang, S. Kumar, and S. Chang, "Hashing with graphs," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 1–8, Bellevue, WA, USA, 2011.
- [17] Y. Fang and Y. Ren, "Supervised discrete cross-modal hashing based on kernel discriminant analysis," *Pattern Recognition*, vol. 98, Article ID 107062, 2020.
- [18] X. Liu, X. Nie, X. Xi, L. Zhu, and Y. Yin, "MoBoost: a self-improvement framework for linear-based hashing," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 871–880, ACM, Beijing, China, 2019.
- [19] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.
- [20] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pp. 1753–1760, Vancouver, BC, USA, 2008.

- [21] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1362–1371, 2014.
- [22] P. Zhang, W. Zhang, W. Li, and M. Guo, "Supervised hashing with latent factor models," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 173–182, ACM, New York, NY, USA, 2014.
- [23] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashing with kernels," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2074–2081, IEEE Computer Society, Providence, RI, USA, 2012.
- [24] G. Wu, J. Han, Y. Guo et al., "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2019.
- [25] Y. Hao, T. Mu, J. Y. Goulermas, J. Jiang, R. Hong, and M. Wang, "Unsupervised t-distributed video hashing and its deep hashing extension," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5531–5544, 2017.
- [26] G. Wu, L. Liu, Y. Guo et al., "Unsupervised deep video hashing with balanced rotation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3076–3082, Sydney, Australia, 2017.
- [27] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep video hashing," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1209–1219, 2017.
- [28] S. Li, Z. Chen, J. Lu, X. Li, and J. Zhou, "Neighborhood preserving hashing for scalable video retrieval," in *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 8211–8220, IEEE, Seoul, South Korea, 2019.
- [29] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10635–10644, IEEE, Seattle, WA, USA, 2020.
- [30] R. Yang, F. Mentzer, L. V. Gool, and R. Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6627–6636, IEEE, Seattle, WA, USA, 2020.
- [31] Y. Wang, X. Nie, Y. Shi, X. Zhou, and Y. Yin, "Attention-based video hashing for large-scale video retrieval," *IEEE Transactions on Cognitive and Developmental Systems*, 2019.
- [32] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1997–2008, 2013.
- [33] G. Ye, D. Liu, J. Wang, and S. Chang, "Large-scale video hashing via structure learning," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 2272–2279, IEEE Computer Society, Sydney, Australia, 2013.
- [34] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas, "Stochastic multiview hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 1–14, 2017.
- [35] X. Nie, W. Jing, C. Cui, C. J. Zhang, L. Zhu, and Y. Yin, "Joint multi-view hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 1951–1965, 2020.
- [36] Y. Wu, X. Liu, H. Qin et al., "Boosting temporal binary coding for large-scale video search," *IEEE Transactions on Multimedia*, vol. 23, pp. 353–364, 2021.
- [37] L. Yuan, T. Wang, X. Zhang et al., "Central similarity quantization for efficient image and video retrieval," in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3080–3089, IEEE, Seattle, WA, USA, 2020.
- [38] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the 2011 IEEE International Conference on Computer Vision*, pp. 2556–2563, IEEE Computer Society, Barcelona, Spain, 2011.
- [39] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: a dataset of 101 human actions classes from videos in the wild," *Computing Research Repository*, 2012, <https://arxiv.org/abs/1212.0402>.

## Research Article

# An Improved Stacked Autoencoder for Metabolomic Data Classification

Xiaojing Fan,<sup>1</sup> Xiye Wang,<sup>2</sup> Mingyang Jiang ,<sup>3</sup> Zhili Pei ,<sup>3</sup> and Shicheng Qiao<sup>3</sup>

<sup>1</sup>College of Engineering, Inner Mongolia University for Nationalities, Tongliao 028000, China

<sup>2</sup>College of Chemistry and Chemical Engineering, Inner Mongolia University for Nationalities, Tongliao 028000, China

<sup>3</sup>College of Computer Science and Technology, Inner Mongolia University for Nationalities, Tongliao 028000, China

Correspondence should be addressed to Mingyang Jiang; [mingyangjiang@imun.edu.cn](mailto:mingyangjiang@imun.edu.cn) and Zhili Pei; [nmdpzl@sohu.com](mailto:nmdpzl@sohu.com)

Received 4 May 2021; Revised 28 June 2021; Accepted 28 July 2021; Published 16 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Xiaojing Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Naru3 (NR) is a traditional Mongolian medicine with high clinical efficacy and low incidence of side effects. Metabolomics is an approach that can facilitate the development of traditional drugs. However, metabolomic data have a high throughput, sparse, high-dimensional, and small sample nature, and their classification is challenging. Although deep learning methods have a wide range of applications, deep learning-based metabolomic studies have not been widely performed. We aimed to develop an improved stacked autoencoder (SAE) for metabolomic data classification. We established an NR-treated rheumatoid arthritis (RA) mouse model and classified the obtained metabolomic data using the Hessian-free SAE (HF-SAE) algorithm. During training, the unlabeled data were used for pretraining, and the labeled data were used for fine-tuning based on the HF algorithm for gradient descent optimization. The hybrid algorithm successfully classified the data. The results were compared with those of the support vector machine (SVM), *k*-nearest neighbor (KNN), and gradient descent SAE (GD-SAE) algorithms. A five-fold cross-validation was used to complete the classification experiment. In each fine-tuning process, the mean square error (MSE) and misclassification rates of the training and test data were recorded. We successfully established an NR animal model and an improved SAE for metabolomic data classification.

## 1. Introduction

Rheumatoid arthritis (RA) is a common systemic autoimmune disease characterized by symmetric polyarthritis and joint destruction [1]. It is traditionally treated with methotrexate combined with the botanical preparation of *Tripterygium wilfordii*. Good results are achieved with this treatment, which improves symptoms and delays disease progression. However, due to severe side effects, treatment compliance is poor. Naru3 (NR) is a traditional Mongolian medicine with a pure botanical preparation. Feng and Xiao [2] and Zhi [3] showed that the therapeutic effect of NR was similar to that of traditional RA treatment methods, and that it was a safe and effective drug of high medicinal value. However, the traditional Mongolian medicine (TMM) research methods are simplistic, and the technologies used are outdated. Therefore, it is necessary to combine these

methods with modern technologies and approaches to further promote the application of TMM in disease diagnosis and treatment.

In recent years, machine learning and its subfield deep learning have been successfully applied in various fields, such as image processing, speech recognition, and natural language processing. Furthermore, they have attracted widespread attention in the fields of medicine, chemistry, and biology, exerting a great impact on people's life.

With the development of high-throughput experimental technologies, high-dimensional, noisy, and redundant biological or medical data can be obtained. However, owing to the cost of the experiments, the sample data are scarce, rendering the standard method of multiple regression inefficient. Assuming that  $p$  is the dimensionality and  $n$  is the amount of data, then  $p \gg n$ . If we use limited data to build a distribution model with  $p$  parameters, it can easily lead to



overfitting in machine learning models. This is a well-known problem in the field of statistics, known as the “curse of dimension” [4]. In the abovementioned research fields, there have been many successful applications of machine learning methods in solving the  $p \gg n$  problem. Ueki and Tamiya have developed a new genetic prediction method using single nucleotide polymorphism (SNP) data in genome-wide association studies (GWASs), which has good predictive ability but is computationally expensive [5]. Ching et al. developed a new artificial neural network (ANN) framework, called Cox-nnet, to predict patient prognosis from high-throughput transcriptomic data, achieving the same or better predictive accuracy compared with that of other methods, including Cox-proportional hazards regression, random survival forests, and CoxBoost, while revealing richer biological information [6]. Xu et al. proposed a feature selection method for one-bit compressed sensing for the classification of high-throughput protein data based on mass spectrometry (MS), which has been employed on MS data to select important features with low dimensions, showing better classification performance for real MS data than traditional methods [7]. Yu et al. developed a support vector machine (SVM) algorithm that identifies optimal sorting gates based on machine learning using positive and negative control populations, taking advantage of more than two dimensions to enhance the ability to distinguish between populations [8]. Furthermore, Xie et al. proposed a Rank-Comp algorithm, which was mainly developed to identify individual-level differentially expressed genes (DEGs) that can be applied to identify population-level DEGs for one-phenotype data [9]. Fouaz and Hacene proposed a genetic algorithm to improve similarity searching pertaining to ligand-based virtual screening, which can identify the most important and relevant characteristics of chemical compounds [10].

In recent years, metabolomic data processing has attracted increasing attention [11]. Metabolomics mainly studies how an organism’s metabolites respond to changes in internal and external environmental conditions [12]. In metabolomics, a machine learning method is used to process data, screen biomarkers, and study the changes in metabolic pathways and the molecular mechanisms of diseases [13]. The analysis of metabolomic data is accompanied by multiple difficulties and challenges due to its high throughput, sparse, and high-dimensional nature and the  $p \gg n$  problem [14, 15]. At present, although traditional machine learning methods such as principal component analysis (PCA) [16], random forest (RF) [17], and SVM [18] have been successfully applied in the field of metabolomics, it is still necessary to find better methods to process metabolomic data. Deep learning methods have been successfully applied in many fields but less in metabolomics [19]. A stacked autoencoder (SAE) is a typical deep learning model with good feature selection and nonlinear expression. An improved SAE algorithm needs to be developed to solve the problem of metabolomic data classification.

Although deep learning is a machine learning subfield with a wide range of applications, a limited number of deep learning-based metabolomic studies have been so far

performed. Asakura et al. proposed an ensemble deep neural network (EDNN) algorithm, which they applied to metabolomic data of various fish species, that is helpful for regression analyses and concerns pertaining to classification in metabolomic studies. The dimensions of their experimental data were 106 and were derived from nuclear magnetic resonance (NMR) measurements [19]. Date and Kikuchi proposed an improved DNN-mean decrease accuracy (MDA) method that can be used for supervised classification and regression modeling and the determination of important variables for the evaluation of biological and environmental samples [20]. Alakwaa et al. proposed that metabolomics holds promise as a new technology for the diagnosis of highly heterogeneous diseases. However, it remains unknown whether DNN, a class of increasingly popular machine learning methods, is suitable for classifying metabolomic data. [21]. Bardley and Robert proposed that metabolomic data are complex because of their high dimensionality and high degree of multicollinearity between variables [22]. Risum and Bro successfully implemented a deep learning algorithm to perform automated spectral deconvolution [23]. Thus, it is reasonable to speculate that we are now within reach of a single deep learning algorithm for accurately classifying raw spectra directly from the instrument [24]. However, the limiting factor for success is to obtain sufficiently large datasets, which are required to train such computationally “greedy” algorithms [25].

Metabolomic data have a high throughput, sparse, high-dimensional, and small sample nature. Deep learning has good predictability, which shows that it can better distinguish different types of metabolomics data. If a good classification can be obtained, it will help us to further complete the selection of biomarkers based on deep learning. In this study, we aimed to introduce an improved framework, named Hessian-free [26] stacked autoencoder (HF-SAE), combining the Hessian-free algorithm and SAE model with Softmax regression for the classification of metabolomic data of NR-treated RA. We used this hybrid algorithm to perform the classification of metabolomic data of NR-treated RA and compared the results with those obtained using the SVM,  $k$ -nearest neighbor (KNN), and gradient descent SAE (GD-SAE) algorithms. A five-fold cross-validation was used to complete the classification experiment. In each fine-tuning process, the mean square error (MSE) and misclassification rates of the training data and test data were recorded. The hybrid algorithm successfully classified the data. A five-fold cross-validation was used to complete the classification experiment. In each fine-tuning process, the MSE and misclassification rates of the training and test data were recorded. We successfully established an NR animal model and an improved SAE for metabolomic data classification.

## 2. Methods

*2.1. Metabolomic Stacked Autoencoder.* The autoencoder was composed of an input layer, a hidden layer, and an output layer. The encoder encoded the input data, which were composed of an input layer and a hidden layer. The decoder completed the reconstruction of the input data, which

consisted of a hidden layer and an output layer. Its purpose was to make the output as close as possible to the input. The training steps of the autoencoder were as follows.

### 2.1.1. Calculation of the Activation Value of Each Layer.

The sample data were the input of the encoder, and the activation value of the hidden layer neurons was calculated by forward conduction. The activation value of the hidden layer neurons was the input of the decoder, and its output (reconstruction value) was calculated in the same manner. If  $f(z)$  is used to represent the activation function,  $a_i^l = f(z_i^l)$  is the activation value of the  $i$ -th neuron in layer  $l$ .  $z_j^{l+1}$  represents the weighted sum of all inputs of the  $j$ -th neuron in the  $l+1$  layer, and its formula is as follows:

$$z_j^{l+1} = \sum_{i=1}^n w_{ij}^l x + b_j^{l+1}, \quad (1)$$

where  $n$  is the number of neurons in the  $l$  layer,  $x$  is the input.  $w_{ij}^l$  is the weight between the  $j$ -th neuron of the  $l+1$  layer and the  $i$ -th neuron of the  $l$  layer, and  $b_j^{l+1}$  is the bias of the  $j$ th neuron in the  $l+1$  layer.

### 2.1.2. Updating Weights and Biases.

The back-propagation (BP) was used to calculate the residual between each layer of neurons and the output layer, and BP was based on gradient descent to reduce the training error of the network. The cost function was used to calculate the least mean square error between the expected output and the actual output.  $J(w, b)$  is the cost function, and the formula is as follows:

$$J(w, b) = \frac{1}{m} \sum_{k=1}^m \left( \frac{1}{2} \|a_{w,b}(x)^k - y^k\|^2 \right), \quad (2)$$

where  $m$  is the number of samples,  $x$  is the input,  $a_{w,b}(x)^k$  is the actual output, and  $y$  is the expected output. The error was used to adjust the weight and bias of the network based on BP so that the error was gradually reduced. Gradient descent was used to continuously update  $w$  and  $b$  so that the output of the autoencoder was close to the input. The adjusted value of  $w$  and  $b$  is proportional to  $\partial/\partial w J(w, b)$  and  $\partial/\partial b J(w, b)$ . The formulas for updating  $w$  and  $b$  are as follows:

$$\begin{aligned} w &= w - \alpha \frac{\partial}{\partial w} J(w, b), \\ b &= b - \alpha \frac{\partial}{\partial b} J(w, b). \end{aligned} \quad (3)$$

### 2.1.3. Activation Function.

SAE is a deep neural network composed of multiple AE units. The model is trained layer by layer using an unsupervised method, and the output of the previous layer is the input of the next layer. The output of the SAE is the input of the classifier that completes the classification. The multihidden layer in SAE can effectively reduce the noise, improve the generalization ability, increase robustness, and improve the classification accuracy.

In the training, the restricted Boltzmann machine (RBM) was used to obtain the initial weight, and the ReLU was used as the activation function. A sigmoid is a common activation function that maps the output between  $[0, 1]$ . However, when the input values are close to infinity or infinitesimal, their gradient is close to zero. Therefore, it is very important to initialize the parameters. If the initial parameters are very small, most neurons are in the saturated state; that is, the gradient is close to 0, which makes the learning of the neural network extremely difficult. As mentioned above, ReLU was selected as the activation function of the SAE, and its formula was as follows:

$$a_i^l = f(z_i^l) = \max(0, z_i^l), \quad (4)$$

where the gradient is always 1 when  $z_i^l > 0$ , which indicates that the gradient is unsaturated. When the error is back-propagated, the update of the SAE weight can be completed quickly. Moreover, the calculation of ReLU is simple, and thus the running speed of the SAE is significantly improved.

### 2.2. Sparse Autoencoder.

To better complete feature selection and reconstruction, the sparse method was used to limit the activity of neurons in the model. If  $x$  is the input of AE and  $a_j^{(2)}(x)$  is the activation value of the hidden node  $j$ , the average activation value of the hidden node  $j$  is as follows:

$$\hat{p}_j = \frac{1}{m} \sum_{i=1}^m [a_j^{(2)}(x^i)], \quad (5)$$

where  $m$  is the number of samples. In the sparse method, the penalty factor is added to the cost function of AE, and its formula is as follows:

$$\sum_{j=1}^{s_2} p \log \frac{p}{\hat{p}_j} + (1-p) \log \frac{1-p}{1-\hat{p}_j}, \quad (6)$$

where  $p$  is a sparse parameter and its value is close to zero,  $\hat{p}_j$  is determined by the connection weights and biases between the nodes of each layer, and  $s_2$  is the number of nodes in the hidden layer. We would like the average activation of each hidden neuron  $j$  to be close to zero. To achieve this, we add an extra penalty term to our optimization objective that penalizes  $\hat{p}_j$  deviating significantly from  $p$ . The optimized cost function of the sparse method is as follows:

$$J(w, b) = J(w, b) + \beta \sum_{j=1}^{s_2} p \log \frac{p}{\hat{p}_j} + (1-p) \log \frac{1-p}{1-\hat{p}_j}, \quad (7)$$

where  $\beta$  is the weight of the sparse penalty factor.

### 2.3. Fine-Tuning.

The proposed HF-SAE consists of SAE and Softmax regression. SAE completes feature selection, and Softmax regression completes the classification of metabolomic data. The structure of our neural network is 4573-1000-500-100-5, which includes three AE units and one Softmax unit. In the pretraining of the SAE, two adjacent layers formed an AE, and the connection weights between layers were obtained by AE training. The input of each AE

hidden layer was the input of the next AE. In the process of fine-tuning, the entire SAE was considered as an encoder, and the mapping of the SAE was considered as a decoder. SAE and its mapping were combined into more hierarchical networks, and HF was used to fine-tune the weights. The fine-tuning structure is illustrated in Figure 1.

### 3. Results

#### 3.1. Dataset

**3.1.1. Chemicals and Reagents.** NR was provided by the Mongolian Medicine Manufacturing Room of the Affiliated Hospital of Mongolia University for the Nationalities (Tongliao, China). NR powder was dissolved in a 0.5% carboxymethyl cellulose (CMC) sodium aqueous solution up to a concentration of 1.00 g/mL and stored at 4°C for animal experimentation.

*Radix Aconiti kusnezoffii* (AK) and *Piper longum* (PL) were purchased from Liqun Drugstore (Tongliao, China). The AK and PL powders were refluxed eight times with ethanol for three times (2 h each time). The extraction solution was slightly boiled. After filtration, the concentrations of AK and PL supernatants were diluted to 0.28 and 0.17 g/mL, respectively.

Complete Freund's adjuvant (CFA) was purchased from Sigma Chemical Co. (St. Louis, MO, USA). Methanol and formic acid (Fisher Scientific, UK) were of HPLC grade. The assays were purchased from Nanjing Jiancheng Bioengineering Institute (Nanjing, China).

**3.1.2. Adjuvant-Induced Arthritis Model Establishment and Treatment.** The study was approved by the ethics committee of the Medicine College of Inner Mongolia University for the Nationalities (IMUNMCEC20210412 [1]). Male Wistar rats (200 ± 10 g) were provided by YiSi Laboratory Animal Technology Co., Ltd. (Changchun, China). All animals were reared under standard conditions (21 ± 2°C, daily sunshine for 14 h) with free access to rodent chow and water in the Affiliated Hospital of Inner Mongolia University for Nationalities and allowed to acclimatize in metabolism cages for 1 week prior to the experiment. The rats were divided into five treatment groups: control (CG), model (MG), NR, AK, and PL, with eight rats in each group. On day 1, the rats in the MG NR, AK, and PL groups were intradermally injected with 0.1 mL CFA in the right posterior toe, while the rats in the CG group were injected with 0.1 mL saline. After 7 days, the rats in the MG, NR, AK, and PL groups were injected with 0.1 mL CFA. On day 14, the rats in the NR, AK, and PL groups were administered NR, AK, and PL, with the doses of 1.00, 0.28, and 0.17 g/kg/day, respectively, for 21 consecutive days, and on day 35 all the rats were euthanized. Blood was collected from the hepatic portal vein and centrifuged at 3500 rpm for 10 min at 4°C. The supernatants were immediately frozen, stored at -20°C, and thawed before analysis. Arthroal cartilage was fixed in 10% formaldehyde for paraffin embedding.

**3.1.3. Serum Sample Preparation.** The serum samples were thawed before analysis, and 100- $\mu$ L aliquots were added to 400  $\mu$ L acetonitrile, followed by vortexing for 30 s and centrifugation at 12000 rpm for 10 min at 4°C. The supernatant was subsequently filtered through a 0.22- $\mu$ m filter membrane.

**3.1.4. Ultrahigh-Performance Liquid Chromatography (UHPLC) Conditions.** A Thermo Dionex Ultimate 3000 UHPLC system coupled with a Q Exactive Focus Orbitrap mass spectrometer (Thermo, USA) was used for metabolomic analysis.

The Waters Acquity UHPLC BEH C18 Column (1.7- $\mu$ m, 2.1 mm × 50 mm, Waters, UK) was maintained at 40°C with a flow rate of 0.3 mL/min<sup>-1</sup> for the separation. The mobile phases were 0.1% formic acid in deionized water (A) and methanol (B). The gradient elution with B was performed according to the following schedule: 8% B for 0–0.5 min, 8–60% B for 0.5–1.5 min, 60–100% B for 1.5–6 min, 100% B for 6–8 min, 100–8% B for 8–9 min, and 8% B for 9–10 min. The sample injection volume was 10  $\mu$ L.

The optimal conditions used for UHPLC-high-definition MS (HDMS) analysis were as follows: nitrogen was used as the sheath and aux gas (at flow rates of 30 and 5 bar, respectively), the spray voltage was 3.0 kV, and capillary and aux gas heater temperatures were 320°C and 300°C, respectively.

The MS data were collected in switching mode (switching between positive and negative spectra) in the mass range of 100–1000 Da. The resolution of the full MS was 70000. In the dd-MS2 discovery mode, the resolution was 17500, and the isolation window was set to 3.0 m/z. The MS2 collision energy was set to 30 eV.

**3.1.5. Data Analysis.** A pooled quality control (QC) sample was prepared by mixing aliquots (20  $\mu$ L) of each sample to monitor the instrument stability. Every day, six QC samples were analyzed to test the stability of the instrument. The Compound Discoverer software (version 2.0) was used for peak detection, alignment, and normalization of the peak area.

**3.2. Five-Fold Cross-Validation Classification Experiment.** The metabolomic dataset contained a small number of samples. To verify the reliability and stability of the HF-SAE model for classification, a five-fold cross-validation method was adopted. The data were divided into five groups on average. Each time, four groups were selected as the training set, and one group was selected as the validation set. The process was repeated until each group of data became a validation set.

We obtained 40 samples of metabolomic data from the NR-treated animal model. To better complete the training of the model, we used the synthetic minority oversampling technique (SMOTE) [27] algorithm to expand the experimental data to 320. There were 255 samples in the training set, 65 samples in the test set, and 4573 variables. The

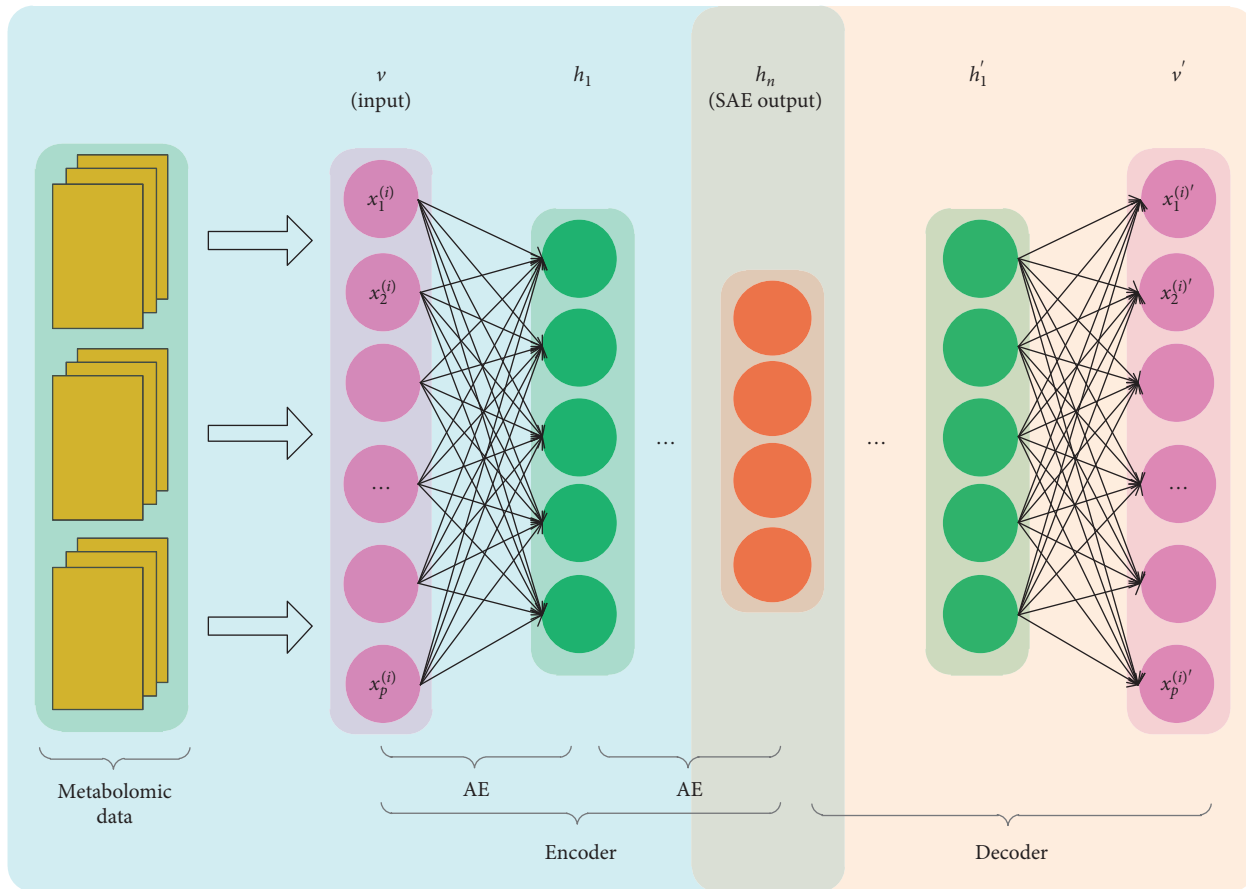


FIGURE 1: Fine-tuning structure (SAE, stacked autoencoder).

experimental data were preprocessed and normalized and divided into five groups (CG, MG, NR, AK, and PL). The structure of our neural network was 4573-1000-500-100-5. The learning rate was set to 0.01. First, the unsupervised method was used to complete the SAE training (the pre-training of the model was completed, and the initial weight was obtained). Second, a supervised method was used to complete the training of the Softmax classifier. Finally, fine-tuning of the model was completed, in which the GD and HF algorithms were used to minimize the cost function. Owing to the small number of training and test data, a min-batch was not used in the training process. The number of iterations in each RBM during training was 500, and the number of iterations during fine-tuning was 4000. The classification accuracies are presented in Table 1.

Table 1 shows the results of the five-fold cross-validation classification experiment for the different datasets. The KNN classification accuracy was between 81.54% and 86.15%, with the lowest accuracy being observed in the third group. The SVM classification accuracy fluctuated dramatically between 73.85% and 81.54%, with the lowest accuracy being observed in the first group. When we used the method combining SAE with Softmax regression, in which fine-tuning was based on GD or HF, the GD-SAE classification accuracy was between 70.77% and 76.92%, and that of HF-SAE was over 90% for each group and did not fluctuate dramatically. The SVM

TABLE 1: Classification accuracy in the five-fold cross-validation experiment (%).

Group	KNN	SVM	GD-SAE	HF-SAE
1	84.62	73.85	70.77	93.85
2	86.15	76.92	76.92	92.31
3	81.54	81.54	73.85	90.77
4	84.62	75.38	75.38	93.85
5	87.69	78.46	73.85	93.85
Mean	84.92	77.23	74.15	92.93

classification accuracy varied greatly and lacked robustness. Although the classification results of KNN and GD-SAE were stable, the classification accuracies were not satisfactory. Therefore, the proposed method is more stable, reliable, and suitable for the classification of metabolomic data. To further compare the effects of different fine-tuning algorithms on the SAE, we recorded the MSE of the training set and the misclassification rate of the training and test sets. A comparison of MSE, training, and test classification error rates is shown in Figure 2.

For terms of the running time, KNN, SVM, GD-SAE, and HF-SAE were about 80 seconds, 80 seconds, 650 seconds, and 900 seconds, respectively. Although HF-SAE had a good classification effect, the computational complexity was very high. In addition, we also evaluated the classification

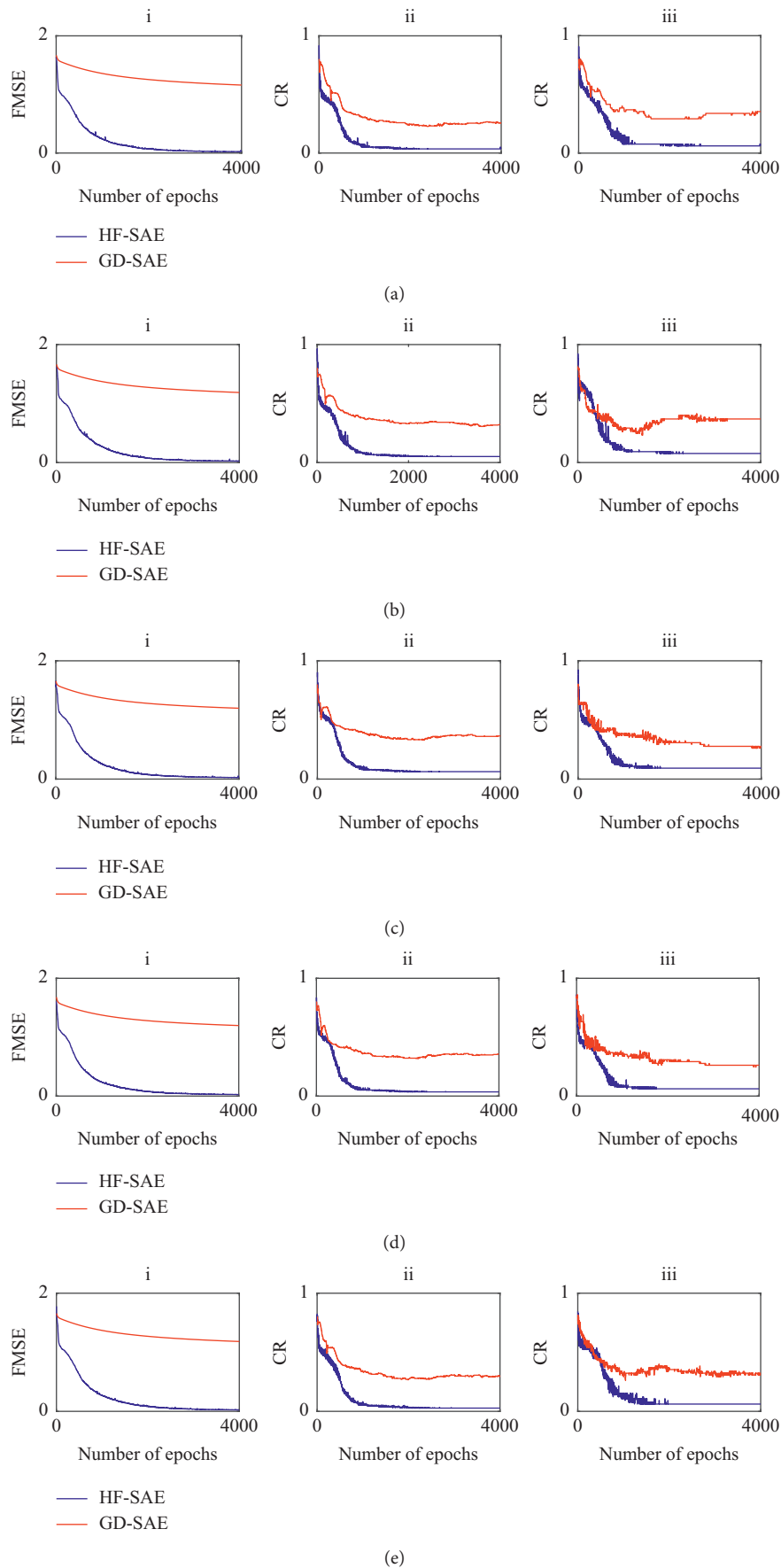


FIGURE 2: Fine-tuning of experimental results on the five-fold data sets. The red and blue lines represent the GD-SAE and HF-SAE results, respectively. In each subgraph of (a) to (e), (i) shows the FMSE, (ii) shows the CR of the training set, and (iii) shows the CR of the test set (GD-SAE, gradient descent stacked autoencoder; HF-SAE, Hessian-free SAE; FMSE, fine-tuning mean square error; CR, classification rate).

accuracy by calculating the kappa value [28], and the range of this value is [0, 1]. If the value was closer to 1, it indicated that the classification accuracy of the model was better. The kappa value of KNN, SVM, GD-SAE, and HF-SAE was 0.81, 0.72, 0.68, and 0.91, respectively. The proposed HF-SAE method had the best kappa value, which further showed that the method had better classification ability.

**3.3. Classification Experiments of Different Training and Test Datasets.** Metabolomic data have a high throughput, sparse, high-dimensional, and small sample nature, which increases the classification difficulty. To further verify the effect of different methods on metabolomic data classification, six datasets with different sizes were established, and the data content difference between each group was 10%. The experiment algorithm was the same as that used in the five-fold cross-validation classification experiment. The number of training sets and test sets for each group, as well as the classification results, is listed in Table 2.

Table 2 shows that when the training data decrease with the decrease in total samples, the classification accuracy of KNN, GD-SAE, and HF-SAE also significantly declines. The reason is that the above three machine learning methods are affected by the reduction of the features that can be obtained, while the accuracy of SVM is relatively stable and less affected by this. Compared with the other three methods, HF-SAE can provide better results. In metabolomic data classification experiments of different scales, it is shown that although the training data are reduced, HF-SAE can still obtain better metabolomic data characteristics.

## 4. Discussion

In the five-fold cross-validation classification experiment, the GD-SAE average classification accuracy rate was the lowest, while that of HF-SAE was the highest. The experimental results show that if the fine-tuning methods of the SAE classification model are different, the effect on the results is very obvious. As the number of iterations increased, the fine-tuning process differed significantly. To further compare the effects of different fine-tuning algorithms on the SAE, we recorded the MSE of the training set and the misclassification rate of the training and test sets. In the five-fold cross-validation experiment based on the GD fine-tuning method, the MSE decreased slowly with the increase in iteration, and the misclassification rate of the training and test sets also gradually decreased during the oscillation process. However, this downward trend was not obvious. When the iteration reached a certain number of times, only a certain range of oscillation occurred, but there was no trend of continuous decline. In the five-fold cross-validation experiment based on the HF fine-tuning method, each indicator had a fast decline speed and small amplitude, and fewer iterations were needed to reach a stable interval compared with GD.

In the process of fine-tuning, the classification accuracy of GD and HF tended to be stable after 2000 iterations, but their classification effect was obviously different. This shows

TABLE 2: Classification accuracies of the different training and test sets (%).

Training set	Test set	KNN	SVM	GD-SAE	HF-SAE
255	65	86.15	76.92	76.92	92.31
230	58	82.76	84.48	77.59	91.38
204	52	80.77	80.77	76.92	88.46
179	45	77.78	75.56	77.78	84.44
153	39	76.92	79.49	74.36	84.62
128	32	71.88	78.13	68.75	81.25

that GD only reaches the local optimal state during fine-tuning and cannot jump out of the local minimum. The change in MSE also explains the difference in the classification accuracy. The MSE of the HF showed a clear downward trend and stabilized after approximately 2000 iterations. Although the GD showed a downward trend, the change was small. The HF-SAE proposed in this paper is superior to the GD-SAE in both the classification results and the fine-tuning process. Moreover, the HF-SAE is stable, reliable, and suitable for metabolomic data classification. A comparison of MSE, training, and test classification error rates is shown in Figure 2. For terms of the running time, KNN was the shortest, HF-SAE time was the longest, and the computational complexity was the highest. HF-SAE achieved better classification results at the cost of consuming more computing resources.

In the classification experiments of different training and test datasets, the number of fine-tuning iterations was 4000. The training data for the experiment were reduced from 255 to 128, and the test data were reduced from 65 to 32. In each group of experiments, the classification result of HF-SAE was better than that of GD-SAE. In the fine-tuning process, the HF-SAE error rate amplitude was relatively large in the initial stage. The classification error rate decreased faster and entered a stable and small-amplitude oscillation range in a short time. The GD-SAE classification accuracy only showed a significant decline in the initial stage of fine-tuning. However, there was no significant change in the classification accuracy, which was significantly different from the classification results of HF-SAE.

In the method comparison, accuracies of HF-SAE were superior for the classification of metabolomic data of NR-treated RA compared with KNN, SVM, and GD-SAE. Accompanying development of metabolomics, robust and accurate classification methods to predict sample labels are in critical need. These results indicated that the HF-SAE developed here was a helpful tool for analyzing biomarkers from the metabolomic data. We concluded that the HF-SAE was capable of identifying important variables that contributed to the constructed HF-SAE model.

Although HF-SAE has excellent classification performance, there are still some considerations in metabonomics research. Compared with some other machine learning methods, HF-SAE is time-consuming computation. In addition, metabonomics datasets are typically small compared with other data, such as text and images. For the classification of metabolomic data of NR-treated RA, we obtained 40 samples. To better complete the training of the model, we

used the SMOTE algorithm to expand the experimental data to 320 because very small data sets may not be suitable for HF-SAE. We also experimented with the effects of reducing training set size and test set size and found that HF-SAE is indeed sensitive to the sample size of the study.

## 5. Conclusions

NR is a traditional Mongolian medicine and a pure botanical preparation, and it has achieved good results in improving symptoms and delaying RA progression. However, the TMM research methods are simplistic, and the technologies used are outdated. Therefore, HF-SAE was used to classify the metabolomic data of NR-treated RA. Metabolomic data are highly dimensional and sparse. With the proposed method, we not only diagnosed RA but also completed an evaluation of NR. In the five-fold cross-validation classification experiment, the proposed method is more stable, reliable, and suitable for the classification of metabolomic data compared with KNN, SVM, and GD-SAE. To further verify the effect of different methods on metabolomic data classification, we performed classification experiments using different training and test datasets. The results show that although the training data are reduced, HF-SAE can still obtain better metabolomic data characteristics. Although the HF-SAE algorithm is a useful tool for the classification, the performance of the method depends on sample size, and how to select biomarkers and explain the model scientifically through the model proposed is also an urgent problem to be solved in this field at this stage.

## Data Availability

Our data still need to be studied in the next stage, so it is not convenient to provide it directly. The data can be made available upon request via email.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported by Science and Technology Projects of Inner Mongolia Autonomous Region (2020GG0190), Research Program of Science and Technology at Universities of Inner Mongolia Autonomous Region (NJZY20112), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (NJYT-19-B18), Natural Science Foundation of Inner Mongolia Autonomous Region of China (2019MS08036 and 2021LHMS06007), Industry Innovation Talent Team of Inner Mongolia Grassland Talent Engineering (2017), Industry-University-Research Innovation Fund of Ministry of Education Science and Technology Development Center—"Zhi Rong Xing Jiao" Fund (2018A01027), Science Research Project of Inner Mongolia University for Nationalities (NMDYB19060), and Inner

Mongolia University for Nationalities Doctoral Research Start Fund Project (BS543 and BS603).

## References

- [1] V. Dziedziejko, M. Kurzawski, K. Safranow et al., "Lack of association between CAG repeat polymorphism in the androgen receptor gene and the outcome of rheumatoid arthritis treatment with leflunomide," *European Journal of Clinical Pharmacology*, vol. 68, no. 4, pp. 371–377, 2012.
- [2] B. Feng and J. Xiao, "Observation on curative effect of Mongolian medicine nar-u-3 pills in treating rheumatoid arthritis," *Journal of North Pharmacy*, vol. 11, no. 2, pp. 36–37, 2014.
- [3] W. Zhi, "Analysis of the clinical efficacy and safety of Mongolian medicine Naru-3 pills in the treatment of rheumatoid arthritis," *Electronic Journal of Clinical Medical Literature*, vol. 67, no. 5, pp. 166–168, 2018.
- [4] A. Narita, M. Ueki, and G. Tamiya, "Artificial intelligence powered statistical genetics in biobanks," *Journal of Human Genetics*, vol. 66, no. 1, pp. 61–65, 2021.
- [5] M. Ueki and G. Tamiya, "Smooth-threshold multivariate genetic prediction with unbiased model selection," *Genetic Epidemiology*, vol. 40, no. 3, pp. 233–243, 2016.
- [6] T. Ching, X. Zhu, and L. X. Garmire, "Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data," *PLoS Computational Biology*, Article ID e1006076, 2018.
- [7] W. Xu, Y. Tian, S. Wang, and Y. Cui, "Feature selection and classification of noisy proteomics mass spectrometry data based on one-bit perturbed compressed sensing," *Bioinformatics*, vol. 36, no. 16, pp. 4423–4431, 2020.
- [8] J. S. Yu, D. A. Pertusi, A. V. Adeniran et al., "CellSort: a support vector machine tool for optimizing fluorescence-activated cell sorting and reducing experimental effort," *Bioinformatics*, vol. 33, no. 6, pp. 909–916, 2016.
- [9] J. Xie, Y. Xu, H. Chen et al., "Identification of population-level differentially expressed genes in one-phenotype data," *Bioinformatics*, vol. 36, no. 15, pp. 4283–4290, 2020.
- [10] B. Fouaz and B. Hacene, "Genetic algorithm-based feature selection approach for enhancing the effectiveness of similarity searching in ligand-based virtual screening," *Current Bioinformatics*, vol. 15, no. 5, pp. 431–444, 2020.
- [11] K. Raja, M. Patrick, Y. Gao et al., "A review of recent advancement in integrating omics data with literature mining towards biomedical discoveries," *International Journal of Genomics*, vol. 2017, Article ID 6213474, 2017.
- [12] W. Andrew, N. Jeremy, H. John et al., "A metabonomic approach to the investigation of drug-induced phospholipidosis: an NMR spectroscopy and pattern recognition study," *Biomarkers*, vol. 5, no. 6, pp. 410–423, 2000.
- [13] A. Scalbert, L. Brennan, C. Manach et al., "The food metabolome: a window over dietary exposure," *The American Journal of Clinical Nutrition*, vol. 99, no. 6, pp. 1286–1308, 2014.
- [14] M. Kircher and J. Kelso, "High-throughput DNA sequencing - concepts and limitations," *BioEssays*, vol. 32, no. 6, pp. 524–536, 2010.
- [15] H. Mohamadi, H. Khan, and I. Birol, "Ntcard: a streaming algorithm for cardinality estimation in genomics data," *Bioinformatics*, vol. 33, no. 9, pp. 1324–1330, 2017.
- [16] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

- [17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [18] G. Harman and S. Kulkarni, "Statistical learning theory and induction," in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed., Springer, New York, NY, USA, pp. 3186–3188, 2012.
- [19] T. Asakura, Y. Date, and J. Kikuchi, "Application of ensemble deep neural network to metabolomics studies," *Analytica Chimica Acta*, vol. 1037, pp. 230–236, 2018.
- [20] Y. Date and J. Kikuchi, "Application of a deep neural network to metabolomics studies and its performance in determining important variables," *Analytical Chemistry*, vol. 90, no. 3, pp. 1805–1810, 2018.
- [21] F. M. Alakwaa, K. Chaudhary, and L. X. Garmire, "Deep learning accurately predicts estrogen receptor status in breast cancer metabolomics data," *Journal of Proteome Research*, vol. 17, no. 1, pp. 337–347, 2018.
- [22] W. Bradley and P. Robert, "Multivariate analysis in metabolomics," *Current Metabolomics*, vol. 1, pp. 92–107, 2013.
- [23] A. B. Risum and R. Bro, "Using deep learning to evaluate peaks in chromatographic data," *Talanta*, vol. 204, pp. 255–260, 2019.
- [24] K. M. Mendez, D. I. Broadhurst, and S. N. Reinke, "The Application of Artificial Neural Networks in Metabolomics: A Historical Perspective," *Metabolomics*, vol. 15, no. 11, Article ID 142, 2019.
- [25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [26] J. Martens and I. Sutskever, "Training deep and recurrent networks with hessian-free optimization," Edited by G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Springer, Berlin Heidelberg, Germany, Second edition, pp. 479–535, Berlin Heidelberg, Germany, 2012, Lecture Notes in Computer Science.
- [27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [28] Z. Qiu, S. W. Lyon, and E. Creveling, "Defining a topographic index threshold to delineate hydrologically sensitive areas for water resources planning and management," *Water Resources Management*, vol. 34, no. 11, pp. 3675–3688, 2020.



## Research Article

# Automatic Diagnosis of Alzheimer's Disease and Mild Cognitive Impairment Based on CNN + SVM Networks with End-to-End Training

Zhe Huang , Minglang Sun , and Chengan Guo 

School of Information and Communication Engineering, Dalian University of Technology, Dalian 116023, China

Correspondence should be addressed to Chengan Guo; [cguo@dlut.edu.cn](mailto:cguo@dlut.edu.cn)

Received 20 May 2021; Revised 29 July 2021; Accepted 6 August 2021; Published 14 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Zhe Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Alzheimer's disease (AD) is an irreversible neurodegenerative disease, and, at present, once it has been diagnosed, there is no effective curative treatment. Accurate and early diagnosis of Alzheimer's disease is crucial for improving the condition of patients since effective preventive measures can be taken in advance to delay the onset time of the disease.  $^{18}\text{F}$ -Fluorodeoxyglucose positron emission tomography ( $^{18}\text{F}$ -FDG PET : PET) is an effective biomarker of the symptom of AD and has been used as medical imaging data for diagnosing AD. Mild cognitive impairment (MCI) is regarded as an early symptom of AD, and it has been shown that MCI also has a certain biomedical correlation with PET. In this paper, we explore how to use 3D PET images to realize the effective recognition of MCI and thus achieve the early prediction of AD. This problem is then taken as the classification of three categories of PET images, including MCI, AD, and NC (normal controls). In order to get better classification performance, a novel network model is proposed in the paper based on 3D convolution neural networks (CNN) and support vector machines (SVM) by utilizing both the excellent abilities of CNN in feature extraction and SVM in classification. In order to make full use of the optimal property of SVM in solving binary classification problems, the three-category classification problem is divided into three binary classifications, and each binary classification is being realized with a CNN + SVM network. Then, the outputs of the three CNN + SVM networks are fused into a final three-category classification result. An end-to-end learning algorithm is developed to train the CNN + SVM networks, and a decision fusion algorithm is exploited to realize the fusion of the outputs of three CNN + SVM networks. Experimental results obtained in the work with comparative analyses confirm the effectiveness of the proposed method.

## 1. Introduction

Alzheimer's disease (AD), as a chronic neurodegenerative disease characterized by irreversible loss of neurons and genetically complex disorder, is often found in the elderly people [1]. Unfortunately, there is no effective curative treatment to reverse AD at present due to the irreversible brain atrophy. Thus, the early diagnosis of AD and its prodromal stage, i.e., mild cognitive impairment (MCI), is vital for patient care and slowing down progressive deterioration [2]. However, patients with MCI only have subtle typical changes, so the accurate diagnosis of MCI is still a difficult problem in early AD diagnosis.

Since the metabolic rate and structure of the brain change accordingly with the progression of AD, the positron emission tomography (PET) is usually utilized to quantify the changes and further applied for computer-aided diagnosis (CAD) of AD [3–5]. In computer-aided AD diagnosis, various pattern recognition-based methods have been employed to predict AD and MCI, and these methods can be roughly divided into two steps, feature extraction and classification. The feature extraction step is to extract discriminative features from the PET images, and the classification step is to get prediction results according to the extracted features. Gray et al. [6] used two support vector machine (SVM) classifiers to identify NC vs. MCI and NC vs. AD, in which the SVMs are trained with the features of

mean signal intensity in the region of native MRI-space of each subject. Garali et al. [7] proposed a novel brain region validity ranking method to separate AD from healthy controls, where SVM and random forest are employed for classification with the features obtained from selected 21 regions. Silveira and Marques [8] developed a boosting classification method that mixed a group of simple classifiers to perform feature selection and segmentation. Cabral and Silveira [9] used different ensemble classifiers based on SVM and random forest to extract diverse features on different sets of brain voxels for classification. Lu et al. [10] extracted three groups of spatial features from PET images and proposed a semisupervised classification method based on random manifold learning with affinity regularization for AD detection.

In recent years, deep learning technology has made great strides on compute vision tasks, e.g. segmentation, classification, and detection. Different from the conventional methods mentioned above, deep learning-based methods can automatically find discriminative features from inputs, avoiding complex processing procedures and manually designed feature extraction operators. Inspired by the impressive performance, amounts of promising studies based on deep learning have been developed for AD prediction. As the 3D PET images can be divided into 2D slices, some scholars employed 2D CNNs to classify AD. Wang et al. [11] proposed an eight-layer convolutional neural network (CNN) with the leaky rectified linear unit and max-pooling layer for AD classification, in which 2D slice of 3D MRI is employed as the input of CNN. Ding et al. [12] introduced the inception v3 that stacks 11 inception modules [13] into the method for AD classification with the  $4 \times 4$  grid images generated from the 3D PET as inputs. Liu et al. [14] proposed a classification framework based on 2D CNN and recurrent neural network (RNN) for AD classification, in which the 2D CNN is used to capture the intraslice features, and RNN is employed to learn and integrate the interslice features. Afterwards, the final results were obtained by fusing the prediction scores from three directions of 3D PET.

Although the mentioned methods with 2D CNNs show effectiveness in AD classification, one of the shortcomings of the methods is that the spatial information of the 3D image is not fully utilized. In order to solve this problem, CNNs with 3D kernels are developed to better utilize the spatial information. Huang et al. [15] constructed a 3D VGG variant model based on single modality for AD diagnosis and achieved multimodality detection by concatenating the multimodality features obtained from MRI and PET images. In addition, the experimental results in [15] showed that hippocampus segmentation is not necessary for improving the performance of the CNN-based classification method. Liu et al. [16] developed a CNN-based model for AD automatic diagnosis with various techniques for designing the CNN model. Zhou et al. [17] utilized a sparse-response deep belief network (SR-DBN) with extreme learning machine (ELM) to classify NC, MCI, and AD. Liu et al. [18] designed a diagnostic framework to extract complementary information from multiple inputs by using zero-masking strategy for prediction. Yee et al. [19] designed a 3D CNN-based network with residual connections for AD diagnosis, and class

activation maps implicate many known regions affected by AD. Pan et al. [20] developed a multiview separable pyramid network-based classification model for AD prediction, in which the features are extracted from axial, coronal, and sagittal views of PET scans with the 3D CNN framework.

As inferred from literature, most of the existing studies for AD diagnosis aim at recognizing AD vs. NC or MCI vs. NC, which regard AD diagnosis as a binary classification problem. Due to the importance of MCI in early diagnosis of AD, the MCI should be accurately recognized from AD and NC. Thus, the three-category classification including NC, MCI, and AD is more reasonable for AD prediction. However, MCI is a transition state from NC to AD, and it is more difficult to be correctly identified compared with the identification of AD and NC. To tackle the 3-category classification, one direct way is to build a 3-category classifier for classification, but this is usually not able to achieve excellent enough performance as usual, especially for the prediction of MCI. Therefore, more attention needs to be paid on the identification of MCI than the other two categories.

Besides, there is still a big space for improving the performance in AD diagnosis of deep learning-based methods due to the limitation of scarce training samples. Since the success of deep learning is partially attributed to the training data, it is believed that a discriminative and robust deep learning-based model can be learned with a large-scale and variable dataset. However, because of the difficulties of PET image acquisition and the high cost of manual annotation, it is infeasible to obtain sufficient training data, which decreases the generalization ability in working data.

In view of the optimal property of SVM in solving binary classifications and the powerful feature extraction ability of deep CNNs, in this paper, we proposed a hybrid model integrated with CNN and SVM networks for AD prediction. The CNN model composed of 3D convolution kernels is developed to extract deep features, while the SVM [21] is utilized for classification. Moreover, an end-to-end training algorithm is developed for further fine-tuning the hybrid system. Since the SVM-based classifier is designed for binary classification, to tackle the 3-category classification problem with the proposed hybrid model, a decision fusion algorithm is proposed to fuse the results of three hybrid models for performing NC, MCI, and AD prediction, in which one network is employed for two of three-category prediction. Extensive experiments have been conducted in the work, and the experimental results show that the proposed approach achieves outstanding performance, compared with the state-of-the-art methods.

The sequel of this paper is organized as follows: Section 2 presents the detailed description of the proposed method, and Section 3 gives the experimental results and performance analysis on the database used in the work. Finally, Section 4 draws conclusions of the contributions made in the paper.

## 2. Proposed Method

*2.1. Overall Scheme of the Proposed Method.* In this paper, we proposed a hybrid model integrated with CNN and SVM networks to predict NC, MCI, and AD. The structure of the

proposed model is shown in Figure 1 that consists of two modules, a feature extraction module based on CNN with 3D kernels (3DCNN), and a SVM-based classification module. Briefly, the feature extraction module is to extract deep features of the input 3D PET images and the classification module is to classify the features to get final decisions. Inspired by [16], the 3DCNN model is redesigned here in according to the purpose of this paper so as to utilize the spatial information provided by the PET images. In addition, to further improve the performance of the model with small batch sizes caused by large 3D data, instance normalization (IN) [22] is employed for normalization. Besides, channel attention [23] is also introduced into the 3DCNN to select more important features. Under the assumption of scarce annotated training data, the SVM-based classification module with the kernel function is employed to find the global structural optimal hyperplane of the training features from all the training samples.

In the training stage, the training data are first sent to the feature extraction module for classification. Then, the outputs of the global average pooling layer (GAP) of the feature extraction module shown in Figure 2 are taken as the inputs of the SVM-based classification module. Next, the parameters of the SVM are solved by the extracted features of training data. Finally, the hybrid model is trained end-to-end by the designed strategy to further optimize the parameters of the model. In the testing stage, the inputs are first sent to the 3DCNN module to extract deep features. Then, the classification results are obtained by the SVM according to the extracted features.

For early AD diagnosis, the proposed model should tackle the problem of 3-category classification. Due to the optimal classification performance for binary classifications of SVM, we divide the three-category classification problem into three binary classification problems so as to boost the performance of 3-category classification, each binary problem being solved by one hybrid model. The overall structure of this three-category classification system is shown in Figure 3, in which it consists of three branches, each binary classification being realized with one 3DCNN + SVM hybrid network. In order to obtain the final classification decision according to the three branch classifiers, a decision fusion algorithm is proposed to fuse the outputs of three 3DCNN + SVM classifiers. The details of the proposed classification system will be given in the sequel sections.

**2.2. 3DCNN-Based Feature Extraction Module.** CNN is widely used in the field of computer vision currently [24] owing to its powerful feature extraction ability. Different from conventional methods that extract features manually, CNN can automatically learn features through an end-to-end training process. In order to utilize the advantage of CNN and the spatial information of input 3D PET images, we design a 3DCNN-based feature extraction module to extract deep features. The structure of the designed 3DCNN network is described in Table 1 and Figure 2, and it is composed of 6 convolutional layers with 3D kernels to extract features, 4 max-pooling layers for downsampling,

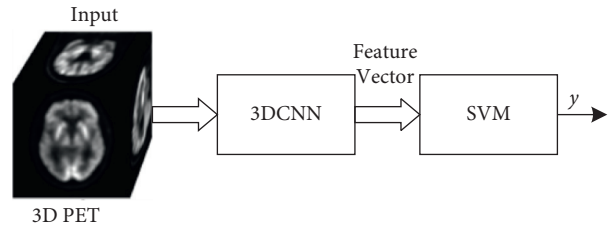


FIGURE 1: Block diagram of the scheme for the 3DCNN + SVM method.

and 4 attention layers to select the informative channels. The typical 3D CNNs, such as 3D DenseNet [25] and 3D ResNet [26], usually employ large-scale kernels to compress the input in the first convolutional layer, which may lose the detailed information. To better learn the lesion feature from the 3D PET images, the first two convolutional layers involved in the model do not perform dimension reduction. The kernel size of the two layers is  $1 \times 1 \times 1$  and  $3 \times 3 \times 3$  with a stride of 1, and the number of kernels is set to 32 and 64 to extend the features, separately. Afterwards, to reduce the computational complexity, a  $2 \times 2 \times 2$  MaxPooling3D layer is employed to reduce the size of the features by half. Then, four convolutional layers, each followed by a channel attention module and a  $2 \times 2 \times 2$  MaxPooling3D layer, are adopted to learn more generalization representations. The channel attention mechanism utilized here is based on the CBAM [23] to enable the model to pay more attention to significant features. The mechanism employs multilayer perceptron (MLP) with one hidden layer to generate attention vector  $W$  as attention weights for feature selection, and  $W$  can be computed as

$$W(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))), \quad (1)$$

where  $F$  denotes the input feature map and  $\sigma(\cdot)$  is the sigmoid function. The MaxPooling3D layer followed the mechanism module is to compress the deep features. Moreover, to speed up the network training and maintain excellent performance on small batch size, the IN [22] layer after each convolutional layer is introduced into the system as in [16] to conduct feature normalization. Besides, after each convolutional layer, a Rectified Linear Unit (ReLU) is utilized as the activation function to conduct nonlinear transformation, thereby preventing the network from degrading into a linear system.

To optimize the model using the annotated data, a fully connected layer after a global average pooling (GAP) layer is utilized to perform binary classification at the end of the last convolutional. Notably, the fully connected layer here is only to optimize the network to gain initial weights, and the outputs of the feature extraction module obtained after the GAP layer are used for subsequent classification.

In addition, to improve the robustness of the model against small batch size training, we update the network with the average gradient from multiple batches. Moreover, the technologies of dropout and label smoothing are employed [27, 28] as well.

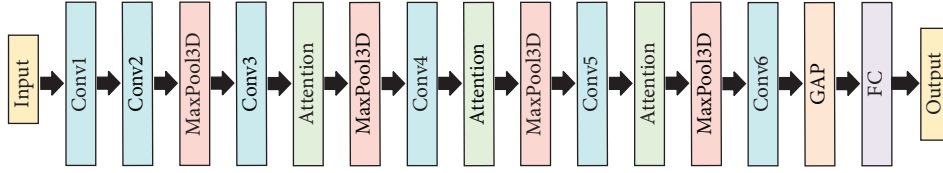


FIGURE 2: The structure of the proposed feature extraction module.

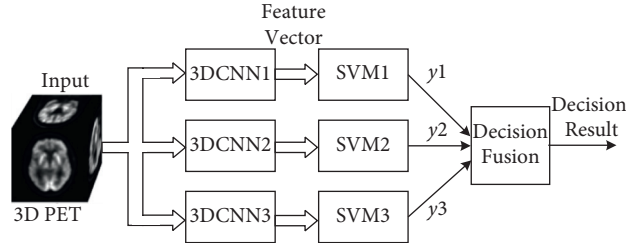


FIGURE 3: Block diagram of the overall scheme for three-category classification.

TABLE 1: The architecture of 3DCNN designed in the paper.

Layer ID	Layer	Kernel number	Kernel size/stride	Output size
0	Input			$1 \times 80 \times 100 \times 76$
1	Conv1	32	(1, 1, 1)/1	$32 \times 80 \times 100 \times 76$
2	Conv2	64	(3, 3, 3)/1	$64 \times 80 \times 100 \times 76$
3	MaxPool3D		(2, 2, 2)/2	$64 \times 40 \times 50 \times 38$
4	Conv3	128	(3, 3, 3)/1	$128 \times 40 \times 50 \times 38$
5	Attention			$128 \times 40 \times 50 \times 38$
6	Maxpool3D		(2, 2, 2)/2	$128 \times 20 \times 25 \times 19$
7	Conv4	256	(3, 3, 3)/1	$256 \times 20 \times 25 \times 19$
8	Attention			$256 \times 20 \times 25 \times 19$
9	Maxpool3D		(2, 2, 2)/2	$256 \times 10 \times 12 \times 9$
10	Conv5	512	(3, 3, 3)/1	$512 \times 10 \times 12 \times 9$
11	Attention			$512 \times 10 \times 12 \times 9$
12	Maxpool3D		(2, 2, 2)/2	$512 \times 5 \times 6 \times 4$
13	Conv6	512	(3 × 3 × 3)/1	$512 \times 3 \times 4 \times 2$
14	GAP			$512 \times 1 \times 1 \times 1$
15	Flatten			512
16	FC			2

**2.3. SVM-Based Classification Module and an End-to-End Training Algorithm for CNN + SVM Model.** SVM with the nonlinear kernel function is able to transform a nonlinear separable problem into a linear separable problem and then finds the structural optimal separate hyperplane that has the maximum margin between the two classes [21]. Because of the small size of annotated training data, the global optimal solution of the training data is available in the conditional that the features extracted by the feature extraction module are fixed. To this end, we employ the SVM with polynomial kernel as the classification module to find the structural optimal solution from all the training data. Nevertheless, it is known that the performance of SVM depends on the support vectors. Once the CNN is trained, the support vectors are fixed. In order to further optimize the parameters of the CNN by using the optimal hyperplane obtained by SVM in the embedding feature space, an end-to-end training algorithm is developed for the proposed hybrid model. The

details of the SVM-based classification module and the end-to-end training algorithm are introduced as follows.

As introduced in [21], the purpose of SVM is to find a separation hyperplane, which maximizes the distances between the margins of two kinds of categories. For  $n$  sample features  $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ ,  $\mathbf{x}_i \in \mathbf{R}^{1 \times d}$ ,  $\mathbf{x}_i = \{x_i^1, x_i^2, \dots, x_i^d\}$ , and  $y_i \in \{-1, 1\}$ , the objective function of SVM is defined by

$$L(w, b, \alpha, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n [a_i y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1], \quad (2)$$

where  $\mathbf{w} \in R^{d \times 1}$  is the coefficient vector,  $b$  is the bias term,  $\alpha \geq 0$  is Lagrange multiplier,  $\xi$  is the slack variables, and  $C \geq 0$  is a penalty parameter used to control the degree of penalty for misclassification. To optimize the SVM by minimizing the objective function, (2) is usually solved by the following dual problem:

$$Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n \{\alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\}, \quad (3)$$

$$\text{Subjected to: } 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^n \alpha_i y_i = 0,$$

where  $i$  and  $j \in 1, \dots, n$  and  $K(x_i, x_j)$  is the kernel function. In the paper, the polynomial kernel function is utilized as the kernel function that is defined as

$$K(x, x_i) = [(x \cdot x_i) + 1]^q, \quad (4)$$

where  $x$  is the input vector,  $x_i$  denotes the support vector of SVM, and  $q$  is the order of polynomial.

For the input  $x$ , the decision function is defined as

$$y = \text{sign} \left( \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right) = \text{sign}(s). \quad (5)$$

Obviously, after solving the parameters of  $\alpha_i$  and  $b$ , the classification result of  $x$  can be obtained. In the paper, the sequential minimal optimization (SMO) algorithm [29] is utilized to calculate  $\alpha_i$  and  $b$ .

As shown in (5), a nonderivable sign function is employed to binarize the value of the linear output of SVM to obtain finally prediction. Due to that the output of sign function is 1 or  $-1$ , the influence of the linear output value  $s$  of SVM is ignored. In general, higher value of the output in the classification indicates higher confidence that the input belongs to the corresponding category. In addition, the BP algorithm cannot be performed by using a nondifferentiable sign function. In order to tackle the problems, a modified SVM is proposed for classification and an end-to-end training algorithm integrated with CNN and modified SVM is proposed to further optimize the hybrid model.

For the modified SVM, the sign function is replaced with a differentiable softmax-based function. Since SVM only has one output, the linear value  $s$  together with its opposite value,  $-s$ , are utilized as the inputs of softmax function. The structure of the modified SVM is shown in Figure 4, and its output can be computed as

$$\mathbf{y} = f \left( \sum_{i=1}^n \mathbf{w}_i K(\mathbf{x}, \mathbf{x}_i) + b \right) = f(s), \quad (6)$$

where  $w_i = \alpha_i y_i$  can be regarded as the weights of the output of  $K(x_i, x)$ ,  $f(\cdot)$  is the softmax function-based differentiable function, and  $\mathbf{y} = \{y_0, y_1\}$  is the output of the modified SVM, in which  $\mathbf{y}$  can be obtained by

$$y_0 = q(x \in d_+) = \frac{e^s}{e^s + e^{-s}}, \quad (7)$$

$$y_1 = q(x \in d_-) = \frac{e^{-s}}{e^s + e^{-s}}, \quad (8)$$

where  $x$  is the input feature,  $s$  is the linear output value of SVM,  $q(x \in d_+)$  denotes the probability of  $x$  belonging to the

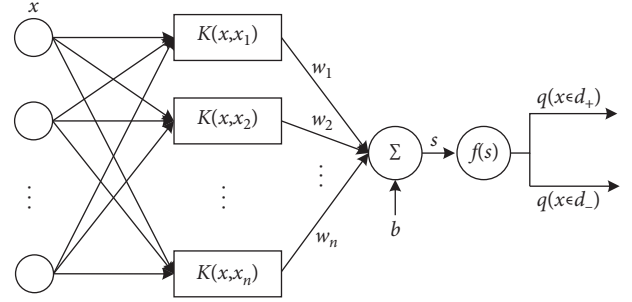


FIGURE 4: The equivalent neural network of SVM with the non-linear kernel function.

positive class, and  $q(x \in d_-)$  denotes the probability of  $x$  belonging to the negative class.

The modified SVM shown in Figure 4 can be equivalent to a neural network with one hidden layer, thus the hybrid model can be trained end-to-end. In the article, the cross-entropy loss is employed to optimize the hybrid model, in which the loss function is defined as

$$H(p, q) = - \sum_{i=1}^n (p(x_i \in d_+) \log q(x_i \in d_+) + (1 - p(x_i \in d_+)) \log(1 - q(x_i \in d_+))), \quad (9)$$

where  $p$  is the label function that is defined as  $p = 1$  if  $x \in$  positive sample; else,  $p = 0$ ; and  $n$  indicates the total number of the training samples.

Equations (7) and (8) can also be represented by

$$q(x \in d_+) = \frac{1}{1 + e^{-2s}}, \quad (10)$$

$$q(x \in d_-) = \frac{1}{1 + e^{2s}}.$$

Obviously, for a positive class feature, only  $s$  tends to positive infinity,  $q(x \in d_+)$  equals to 1, and loss function  $H(p, q)$  tends to 0. Since  $s$  is positively related to the distance from  $x$  to the hyperplane of SVM, the larger  $s$  means the greater distance between  $x$  and the hyperplane. For a negative class feature, the loss function  $H(p, q)$  tends to 0 when  $s$  tends to negative infinity. Thus, the loss function can be utilized to optimize the features of CNN and further increase the margin between the two classes.

The optimization of SVM is to find the optimal hyperplane from all training samples, which is different from the backwardpropagation (BP) algorithm-based optimization of 3DCNN. In order to jointly optimize the hybrid system with the BP algorithm and maintain the optimal structure of SVM, the parameters of the SVM are not adjusted in the process of optimizing CNN with the BP algorithm. After CNN converged, the parameters of SVM are re-calculated by the SMO algorithm to find the new separate hyperplane for further optimization.

Details for these operation steps are as follows:

- (i) Initialize a 3DCNN and a SVM to be trained, and divide the PET dataset into 3 subsets (training set, verification set, and test set)
- (ii) Train the 3DCNN by using the samples in the training set until converged, and then, use the converged 3DCNN to extract the feature vector output from its last pooling layer using all the samples in the training set and in the verification set as input
- (iii) Train the SVM by using the extracted feature vectors as training samples obtained by using the input samples in the training set in Step (ii), until the SVM converged
- (iv) Construct a 3DCNN+SVM network using the trained 3DCNN and SVM, and replace sign function with softmax function as described in (7) and (8)
- (v) Fine-tune the 3DCNN+SVM network by using the samples both in the training set and in the verification set and the loss function computed according to (9), with the weights of the SVM fixed (without updated), until the 3DCNN converged basically
- (vi) Re-train the SVM by using the extracted feature vectors output from the 3DCNN obtained in Step (v) without updating the 3DCNN, until the SVM converged basically
- (vii) Repeat the Steps (iv)–(vi), until the whole 3DCNN+SVM network converged
- (viii) Test the trained 3DCNN+SVM network by using the samples in the test set

**2.4. Decision Fusion Algorithm of Three Binary Classifiers.** At present, most of the existing studies related to AD aim to solve binary classification problems, such as AD vs. NC and MCI vs. NC. However, in practical applications, a robust 3-category classification model is crucial for the early diagnosis of AD as mentioned above. Generally, this problem can be well solved directly by a 3-category classifier, but it may not be suitable for AD prediction with a simple 3-category classifier as the MCI is hard to be accurately identified from AD and NC. Since the proposed SVM-based classification module can achieve global optimal structure solutions for binary classification on the training data, 3-category classification task can be solved by using three hybrid models with the proposed decision fusion algorithm.

As shown in Figure 3, three  $3DCNN_i + SVM_i$  networks ( $i = 1, 2, \text{ and } 3$ ) are built up to cope with the three-category classification with one network for solving two of three-category classification. Before making a final decision, three 3DCNN+SVM hybrid networks need to be trained in advance for performing the binary classifications of AD vs. NC, MCI vs. NC, and AD vs. MCI. Afterwards, for a 3D PET image to be classified, it is first fed into the three  $3DCNN_i + SVM_i$  networks ( $i = 1, 2, \text{ and } 3$ ) respectively, and then, outputs of the three classification models can be

obtained. In order to use the results of the three classifiers effectively, in the paper, we design a decision fusion algorithm as follows to get the final decision:

- (1) If the results of two classification models belong to the same category, the category is regarded as the final classification result
- (2) If all the three classification results are different, the final decision is made according to the absolute value,  $|s_i|$ , of the linear output of the  $SVM_i$  ( $i = 1, 2, \text{ and } 3$ ) as follows:

$$k = \arg \max_i (|s_i|). \quad (11)$$

Then, final classification result is selected as the binary classification result of the  $k$ th 3DCNN+SVM network (i.e., the output of the  $SVM_k$ ).

### 3. Experiments

**3.1. Database and Data Preprocessing.** In order to evaluate the proposed method in AD prediction, the  $^{18}\text{F}$ -Fluorodeoxyglucose positron emission tomography ( $^{18}\text{F}$ -FDG PET:PET) data obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [30] launched in 2003 are utilized in the paper, in which ADNI has been committed to tracking the progress of AD through biomarkers and clinical assessments. By identifying sensitive and specific markers of early AD progression in the database provided by the participants at different time, it can help researchers and clinicians develop new treatments, monitor the effectiveness, and reduce the cost of clinical trials.

In this work, we adopt 2706 3D PET images from 959 ADNI participants, including 267 AD subjects, 340 MCI subjects, and 352 NC subjects. Table 2 presents the demographic details of the studied subjects in the work, where MMSE is the abbreviation of the Mini-Mental State Examination. The PET images are first preprocessed by performing image registration, spatial normalization, intensity normalization, and image smoothing. Then, the voxels outside the brain are removed from the PET images, and the images are cropped to a size of  $80 \times 100 \times 76$ .

**3.2. Implementation Settings and Evaluation Indexes.** All the models and algorithms adopted in the work have been implemented, and all the experiments are conducted by using Python on a CPU+GPU platform with the CPU of Intel®Core™ i77700@3.60 GHz and the GPU of NVIDIA GeForce GTX 1080Ti.

In the experiment, five-fold cross-validation is performed, where the dataset is divided into 5 equal parts in which 1 part is used as the testing data and 4 parts are used as training data with 1 part of them as verification data. And, the experiments are conducted 5 times in turn, and the mean values of the results of 5 trials are used as final indexes of the method. The data are strictly divided according to patient's IDs to ensure that the image samples of the same person will not be put into different datasets, i.e., the PET images of one participant are put into only one part in the data partition to

TABLE 2: Demographic characteristics of the studied subjects.

Diagnosis	Number	Age	Gender (F/M)	MMSE
AD	514	75.98 ± 7.62	305/209	19.26 ± 5.64
MCI	1247	76.47 ± 7.54	809/438	22.83 ± 6.56
NC	945	76.99 ± 5.95	544/405	27.83 ± 3.63

TABLE 3: Evaluation of the proposed 3DCNN + SVM with E2E applied to binary classification of AD vs. NC samples (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
Gray [6]*	—	—	—	—	81.60	82.7	80.4	90.0
Lu [10]*	—	—	—	—	89.44	88.89	90.0	—
Silveira [8]*	—	—	—	—	<b>90.97</b>	—	—	—
Ding et al. [12]	98.92	99.49	98.61	98.95	86.27	86.97	85.78	90.50
Liu et al. [14]	98.61	<b>99.59</b>	98.07	99.84	89.31	87.50	90.32	92.96
Huang et al. [15]	<b>99.21</b>	99.43	98.48	99.35	88.68	87.74	89.17	91.98
Proposed	99.19	99.39	<b>99.54</b>	<b>99.88</b>	<b>90.82</b>	<b>91.29</b>	<b>90.59</b>	<b>93.75</b>

avoid data leakage. The stochastic gradient descent (SGD) algorithm is utilized to minimize the loss function in training the proposed model. The batch size is set to 4, and the weights of the network are updated every four batches for better convergence in the training process.

To better evaluate the performance of the proposed method and state-of-the-art methods, 4 technical indexes [20] are employed for evaluation, including accuracy (ACC), sensitivity (SEN), specificity (SPE), and AUC (area under ROC curve). The ACC, SEN, and SPE are the proportion of correct predictions among all samples, positive samples, and negative samples, respectively. Each of the indexes is identified as

$$\begin{aligned}
 \text{ACC} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\
 \text{SEN} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
 \text{SPE} &= \frac{\text{TN}}{\text{TN} + \text{FP}},
 \end{aligned} \tag{12}$$

where TP, FP, TN, and FN separately indicate the true positive, false positive, true negative, and false negative. The AUC is obtained by computing the area under the receiver operating characteristic curve (ROC) which is the curve to describe the relationship between the true positive rate (TPR) and the false positive rate (FPR) under varied threshold settings. Obviously, the higher result stands for better performance.

**3.3. Evaluation of the Proposed Method Applied to Binary Classification.** In this section, experiments are conducted for the proposed 3DCNN + SVM classification method and also for the other state-of-the-art methods, respectively. The methods proposed in the cited literature were originally designed for solving binary classification problems, such as the prediction of AD vs. NC or MCI vs. NC. For our proposed method, since a single 3DCNN + SVM model with end-to-end training is also proposed for solving a binary

classification problem, we just need to use a single 3DCNN + SVM network to perform the classification without needing three such networks.

Aiming to better evaluate the generalization performance of the proposed method and the state-of-the-art ones, we test the approaches on both training and testing sets. Tables 3–5 present the experimental results implemented on the data of AD vs. NC, MCI vs. NC, and AD vs. MCI, respectively. Since the experimental results given in the cited literature were obtained by using different data partitions under different experiment settings, in order to make a fair comparison, the methods without “\*” are implemented by using the same PET data under the same experiment settings as in ours in the paper; meanwhile, the results of the methods with “\*” are cited by the corresponding reference. From the results shown in the tables, one can see that the proposed method generally performs better than the other ones, and its effectiveness can be confirmed by the experiments.

In addition, Figure 5 displays the comparisons of the ROC curves on AD vs. NC, MCI vs. NC, and AD vs. MCI. From the figure, we can observe that the proposed method achieves the best AUC compared with the mentioned state-of-the-art methods and proves the robustness and effectiveness of the hybrid model.

**3.4. Evaluation of the Proposed Method Applied to 3-Category Classification.** As mentioned before, in order to solve the early prediction of AD symptoms, a hybrid 3-category classification system is developed by integrating three binary 3DCNN + SVM classifiers with an optimal decision fusion scheme. In this section, we present the experimental results to evaluate this 3-category classification system by using the 3D PET images from MCI, AD, and NC subjects. In order to demonstrate the effectiveness of the proposed method, the “CNN + BGRU” method introduced in [14] and the “ADCNN” model proposed by Liu et al. [16] are implemented in the paper for comparison. In this work, we re-implement the CNN-based state-of-the-art methods and train and test by using the same 3D PET images as used in

TABLE 4: Evaluation of the proposed 3DCNN + SVM with E2E applied to binary classification of MCI vs. NC samples (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
Gray [6]*	—	—	—	—	70.20	73.80	62.30	73.0
Lu [10]*	—	—	—	—	<b>79.63</b>	—	—	—
Silveira [8]*	—	—	—	—	70.00	46.96	<b>80.44</b>	—
Ding et al. [12]	98.70	98.05	99.55	99.43	72.37	74.70	69.31	79.19
Liu et al. [14]	99.04	98.52	99.74	99.73	73.80	73.16	74.69	80.45
Huang et al. [15]	98.30	97.72	99.09	<b>99.97</b>	73.52	75.50	70.90	79.65
Proposed	<b>99.54</b>	<b>99.26</b>	<b>99.90</b>	99.88	<b>76.68</b>	<b>77.80</b>	<b>75.57</b>	<b>82.39</b>

TABLE 5: Evaluation of the proposed 3DCNN + SVM with E2E applied to binary classification of AD vs. MCI samples (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
Gray [6]*	—	—	—	—	68.2	58.3	73.0	70.0
Lu [10]*	—	—	—	—	—	—	—	—
Silveira [8]*	—	—	—	—	70.0	—	—	—
Ding et al. [12]	92.39	97.50	90.29	98.59	71.19	68.52	72.36	77.28
Liu et al. [14]	96.10	<b>99.93</b>	94.52	99.18	73.79	<b>75.00</b>	73.28	79.16
Huang et al. [15]	96.09	99.66	94.53	99.39	73.83	74.93	73.42	78.53
Proposed	<b>98.45</b>	99.24	<b>97.31</b>	<b>99.91</b>	<b>74.29</b>	70.78	<b>75.48</b>	<b>80.11</b>

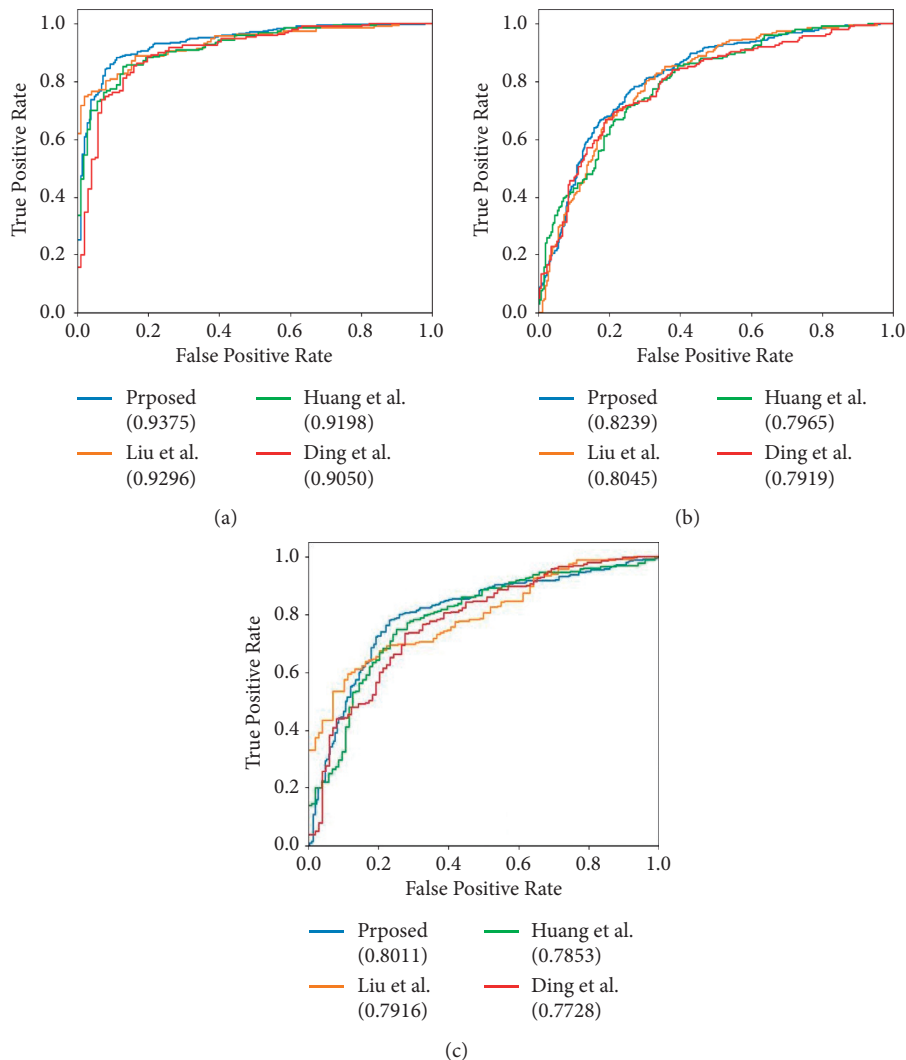


FIGURE 5: ROC curves of the proposed method and state-of-the-art methods on AD vs. NC, MCI vs. NC, and AD vs. MCI. (a) AD vs. NC. (b) MCI vs. NC. (c) AD vs. MCI.



TABLE 6: Evaluation of the proposed method applied to 3-category classification in terms of ACC (%).

Method	Training set				Testing set			
	AD	MCI	NC	Average	AD	MCI	NC	Average
Cabral et al. [9]*	—	—	—	—	—	—	—	66.78
3DCNN	<b>99.85</b>	98.62	99.89	<b>99.45</b>	65.63	62.12	70.43	65.66
CNN + BGRU [14]	97.75	<b>99.89</b>	99.86	99.17	58.65	66.22	68.28	65.53
ADCNN [16]	99.81	98.35	<b>99.99</b>	99.38	65.16	63.25	68.63	65.44
Proposed	99.17	97.83	99.37	98.79	<b>73.42</b>	<b>67.86</b>	<b>72.28</b>	<b>71.19</b>

TABLE 7: Ablations studies of the proposed 3DCNN + SVM model applied to binary classification of AD vs. NC (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
3DCNN	<b>99.50</b>	<b>99.72</b>	99.39	<b>99.97</b>	89.83	90.94	89.26	92.68
3DCNN + SVM	98.62	99.01	98.40	99.95	90.20	90.34	90.19	93.36
3DCNN + SVM + E2E	99.19	99.39	<b>99.54</b>	99.88	<b>90.82</b>	<b>91.29</b>	<b>90.59</b>	<b>93.75</b>

TABLE 8: Ablations studies of the proposed 3DCNN + SVM model applied to binary classification of MCI vs. NC (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
3DCNN	<b>99.80</b>	<b>99.66</b>	99.55	<b>99.99</b>	75.04	75.54	72.97	79.74
3DCNN + SVM	98.35	98.30	98.41	<b>99.99</b>	75.58	76.42	74.41	80.80
3DCNN + SVM + E2E	99.54	99.26	<b>99.90</b>	99.88	<b>76.68</b>	<b>77.80</b>	<b>75.57</b>	<b>82.39</b>

TABLE 9: Ablations studies of the proposed 3DCNN + SVM model applied to binary classification of AD vs. MCI (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
3DCNN	98.37	<b>99.73</b>	<b>97.72</b>	99.81	73.56	<b>73.84</b>	73.51	77.82
3DCNN + SVM	97.72	99.43	96.80	99.84	73.95	71.88	74.89	78.75
3DCNN + SVM + E2E	<b>98.45</b>	99.24	97.31	<b>99.91</b>	<b>74.29</b>	70.78	<b>75.48</b>	<b>80.11</b>

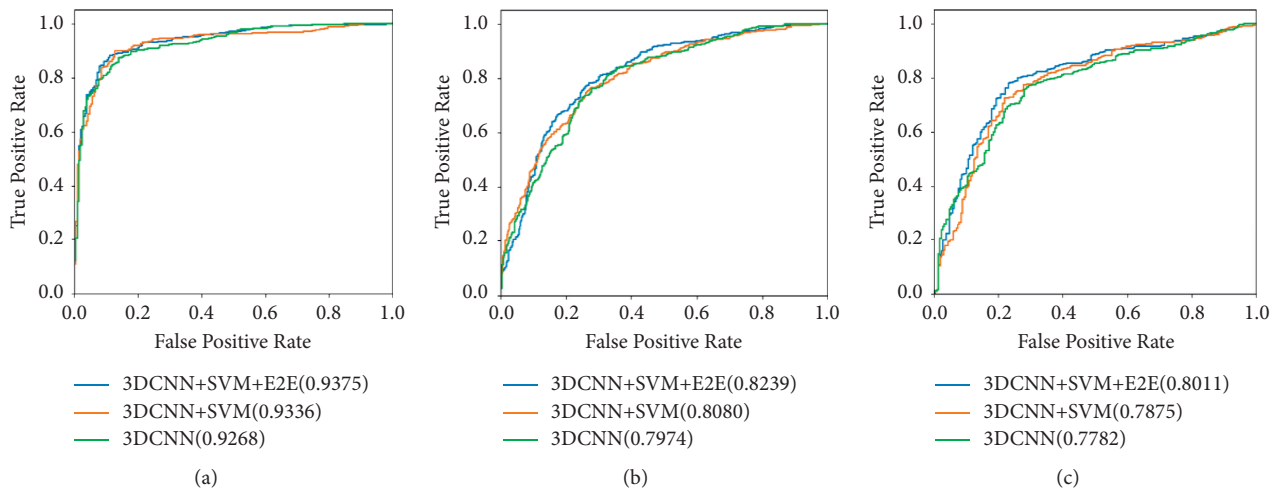


FIGURE 6: ROC curves of the ablation experiments on 3DCNN + SVM. (a) AD vs. NC. (b) MCI vs. NC. (c) AD vs. MCI.

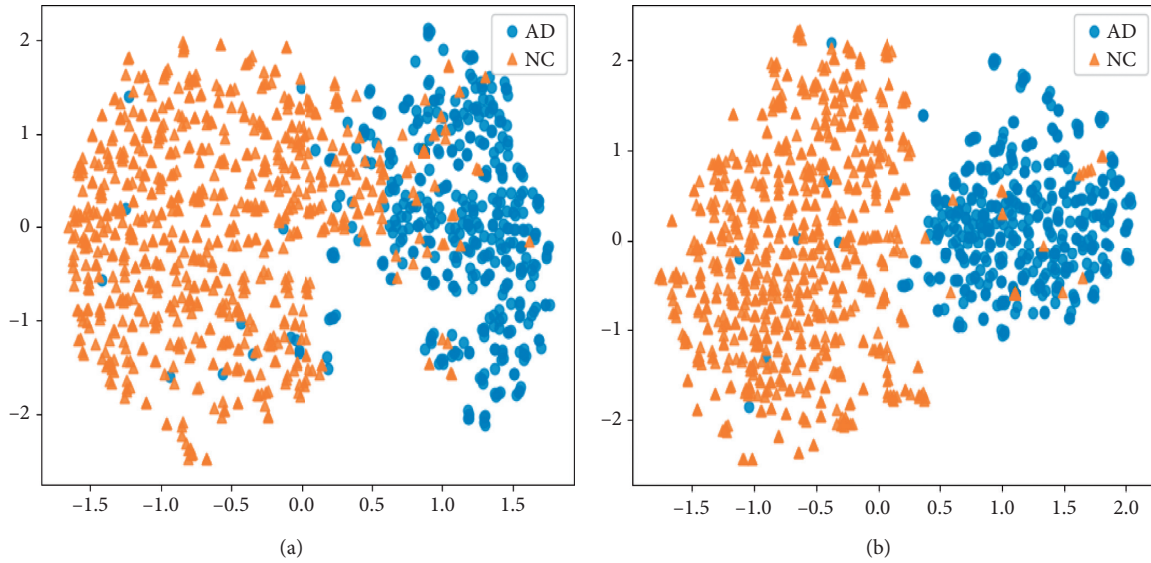


FIGURE 7: The visualization results of the features extracted from the 3DCNN before and after the end-to-end training algorithm on AD vs. NC. (a) The results before end-to-end training. (b) The results after end-to-end training.

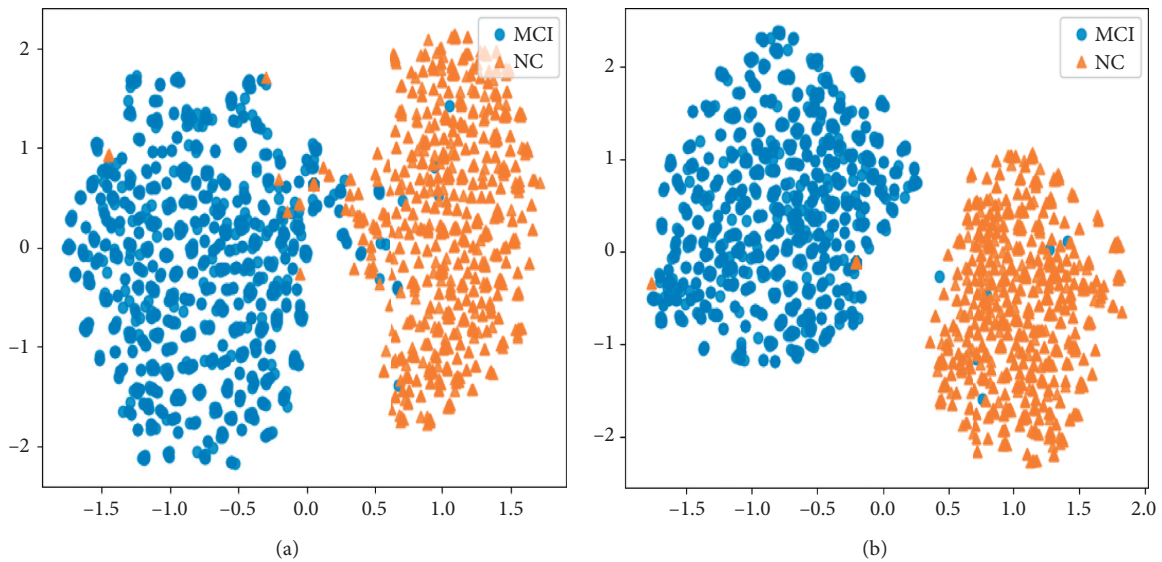


FIGURE 8: The visualization results of the features extracted from the 3DCNN before and after the end-to-end training algorithm on MCI vs. NC. (a) The results before end-to-end training. (b) The results after end-to-end training.

the paper. Table 6 shows the experiment results on training and testing sets, in which the experimental results of “3DCNN” are also included that are obtained by using a three-dimensional CNN network with the same structure as the 3DCNN shown in Figure 2 but adjusting the number of the output fully connected layer nodes from 2 to 3. This “3DCNN” model is also trained and tested by using the same data as the other models and also used for performance comparison in the experiment.

From the results shown in Table 6, it can be seen that the proposed hybrid 3-category classification system obtains a significant improvement on all the four evaluation indexes, compared with the others. According to the results of Tables 3–6,

it implies that the proposed method not only achieves excellent performance in binary classification tasks but also outperforms the other methods in three category classification by applying the proposed decision strategy with three proposed binary classifiers.

*3.5. Ablation Experiments of the CNN + SVM Hybrid Model with End-to-End Training Algorithm.* For the proposed method, the SVM is employed to replace the fully connected layer of the proposed 3DCNN as the classifier, and an end-to-end algorithm is developed to optimize the hybrid model.

In order to compare the performance of the improvement and, meanwhile, validate the effectiveness of the

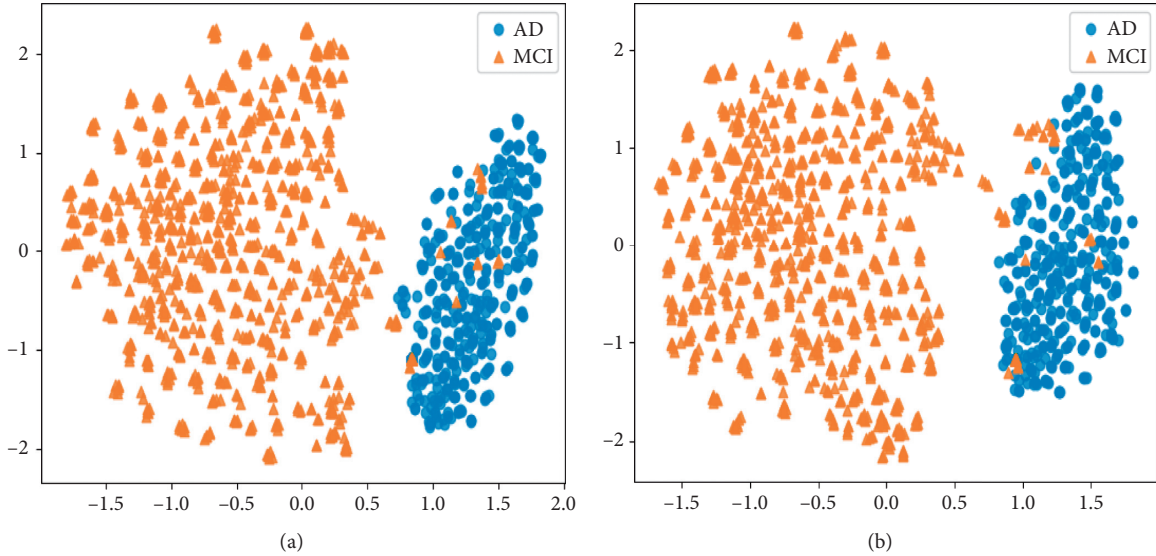


FIGURE 9: The visualization results of the features extracted from the 3DCNN before and after the end-to-end training algorithm on AD vs. MCI. (a) The results before end-to-end training. (b) The results after end-to-end training.

TABLE 10: Ablations studies of the channel attention mechanism on AD vs. NC (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
3DCNN w/o Atten	98.74	<b>99.88</b>	99.12	99.78	89.41	90.55	88.83	91.92
3DCNN with Atten	<b>99.50</b>	99.72	<b>99.39</b>	<b>99.97</b>	<b>89.83</b>	<b>90.94</b>	<b>89.26</b>	<b>92.68</b>

TABLE 11: Comparison of different normalization functions of SVM on AD vs. NC (%).

Method	Training set				Testing set			
	ACC	SEN	SPE	AUC	ACC	SEN	SPE	AUC
3DCNN with BN	99.04	99.21	<b>99.48</b>	99.92	89.36	89.68	89.16	91.96
3DCNN with IN	<b>99.50</b>	<b>99.72</b>	99.39	<b>99.97</b>	<b>89.83</b>	<b>90.94</b>	<b>89.26</b>	<b>92.68</b>

integration, we conduct ablation experiments to evaluate the proposed improvement including the SVM-based classifier and the end-to-end algorithm. Tables 7–9 show the ablation results of the proposed module evaluated on the data of AD vs. NC, MCI vs. NC, and AD vs. MCI on both training and testing sets, respectively. In order to make a fair comparison, the 3DCNN network illustrated in Figure 2 is employed as the baseline for further comparison. To assess the effects of the SVM-based classifier, in this section, the results of “3DCNN + SVM” are obtained by directly combining the baseline with an SVM without the proposed end-to-end algorithm, i.e., the two modules are trained separately. From the results of the three binary-category classification tasks, the “3DCNN + SVM” can give relatively better overall performance than the baseline, which proves the effectiveness of the SVM-based classifier on AD prediction with scarce training data. To further optimize the hybrid model, the end-to-end algorithm is developed to fine-tune the 3DCNN model. The results of “3DCNN + SVM + E2E” are obtained by using the proposed end-to-end training methods. With the assistance of the end-to-end algorithm, the performance of the proposed module is

improved again on the indexes of ACC, SEN, SPE, and AUC. Figure 6 displays the comparisons of the ROC curves on AD vs. NC, MCI vs. NC, and AD vs. MCI for the above ablation experiments, which further proves the effectiveness of the proposed implementations for AD prediction. Therefore, according to the ablation studies, the proposed SVM-based classifier and the end-to-end algorithm play an important role in boosting the performance of the baseline on AD diagnosis.

In addition, we also visualize the features extracted by the outputs after the global average pooling layer of 3DCNN before and after end-to-end training, and the visualization results are shown in Figures 7–9. From the results, it can be seen that the features in visual are easier to be recognized after end-to-end training, which confirms the feasibility of the proposed end-to-end algorithm.

*3.6. Ablation Studies of the Implemented 3DCNN.* In this section, we validate the effectiveness of the key technologies employed in the 3DCNN model, mainly including the channel attention mechanism and the instance normalization method.

Table 10 shows the ablation results on AD vs. NC prediction. The “3DCNN w/o Atten” is the model that removes the channel attention mechanism from the designed 3DCNN, and the “3DCNN with Atten” is the proposed 3DCNN model shown in Figure 2. As can be seen, the model with channel attention is superior to the model without the attention mechanism in the four indexes, which shows that the measure is effective for improving the recognition accuracy.

In addition, due to the small batch size caused by a large scale of image data, the instance normalization (IN) is employed to replace the typical batch normalization (BN) for the designed 3DCNN model. The comparison experiments are conducted in Table 11, in which the 3DCNN with BN is the model that uses BN as the normalization function, and the 3DCNN with IN is the proposed 3DCNN model. It can be seen from the results that the performance of the 3DCNN is improved after replacing BN with IN, and the sensitivity is the most obvious. As a result, from the results in Tables 10 and 11, the measures introduced into the proposed 3DCNN model are helpful in improving the performance of the model.

#### 4. Summary and Further Working Direction

In this paper, we proposed a new classification system for early automatic diagnosis of AD symptoms based on 3DCNN and SVM, in which the original 3-category classification problem is divided into three binary classification problems; each binary classification is realized with a 3DCNN + SVM model. Furthermore, an end-to-end learning algorithm is developed for training the 3DCNN + SVM networks, and an optimal decision fusion scheme is proposed to fuse the outputs of three 3DCNN + SVM classifiers based on the criteria of majority voting. By using these methods, the advantages of both CNN and SVM models can be fully utilized; thus, the overall performance of the system can be significantly improved. Experimental results obtained in the paper confirm the effectiveness of the proposed approach that outperforms the existing start-of-the-art methods in terms of the class accuracy, sensitivity, specificity, and area under ROC.

It is noticed that, from the experimental results obtained in the paper, the classification performance of MCI samples still leaves some room for further improvement, and the correct identification of this category samples is crucial for the early diagnosis of AD. Therefore, a more effective method is needed to be developed to overcome this shortage, which will be the future research direction of the paper.

#### Data Availability

The publicly available ADNI dataset [30] can be downloaded through the website at <http://adni.loni.usc.edu/>.

#### Disclosure

This paper was partly published in the 13th International Conference on Advanced Computational Intelligence (ICACI), Wanzhou, China, May 14–16, 2021 (<https://ieeexplore.ieee.org/document/9435894>).

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### References

- [1] A. Association, “2019 Alzheimer’s disease facts and figures,” *Alzheimer’s and Dementia*, vol. 15, no. 3, pp. 321–387, 2019.
- [2] X. Zhu, H.-I. Suk, Y. Zhu, K.-H. Thung, G. Wu, and D. Shen, “Multi-view classification for identification of Alzheimer’s disease,” *Machine Learning in Medical Imaging*, pp. 255–262, 2015.
- [3] W. Jagust, A. Gitcho, F. Sun, B. Kuczynski, D. Mungas, and M. Haan, “Brain imaging evidence of preclinical Alzheimer’s disease in normal aging,” *Annals of Neurology*, vol. 59, no. 4, pp. 673–681, 2006.
- [4] L. Mosconi, V. Berti, L. Glodzik, A. Pupi, S. De Santi, and M. J. de Leon, “Pre-clinical detection of Alzheimer’s disease using FDG-PET, with or without amyloid imaging,” *Journal of Alzheimer’s Disease*, vol. 20, no. 3, pp. 843–854, 2010.
- [5] C. R. Jack, D. S. Knopman, W. J. Jagust et al., “Hypothetical model of dynamic biomarkers of the Alzheimer’s pathological cascade,” *The Lancet Neurology*, vol. 9, no. 1, pp. 119–128, 2010.
- [6] K. R. Gray, R. Wolz, S. Keihaninejad et al., “Regional analysis of FDG-PET for use in the classification of Alzheimer’s disease,” in *Proceedings of the 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1082–1085, Chicago, IL, USA, March 2011.
- [7] I. Garali, M. Adel, S. Bourennane, and E. Guedj, “Region-based brain selection and classification on PET images for Alzheimer’s disease computer aided diagnosis,” in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1473–1477, Quebec City, Canada, September 2015.
- [8] M. Silveira and J. S. Marques, “Boosting Alzheimer’s disease diagnosis using PET images,” in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pp. 23–26, Istanbul, Turkey, August, 2010.
- [9] C. Cabral and M. Silveira, “Classification of Alzheimer’s disease from FDG-PET images using favorite class ensembles,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2477–2480, Osaka, Japan, July 2013.
- [10] S. Lu, Y. Xia, T. W. Cai et al., “Semi-supervised manifold learning with affinity regularization for Alzheimer’s disease identification using positron emission tomography imaging,” in *Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2251–2254, Milan, Italy, August 2015.
- [11] S.-H. Wang, P. Phillips, Y. Sui et al., “Classification of Alzheimer’s disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling,” *Journal of Medical Systems*, vol. 42, no. 5, p. 85, 2018.
- [12] Y. M. Ding, J. H. Sohn, M. G. Kawczynski et al., “A deep learning model to predict a diagnosis of Alzheimer disease by using 18F-FDG PET of the brain,” *Radiology*, vol. 290, no. 2, pp. 456–464, 2018.
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe et al., “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Seattle, WA, USA, June 2016.
- [14] M. Liu, D. Cheng, and W. Yan, “Classification of Alzheimer’s disease by combination of convolutional and recurrent neural

- networks using FDG-PET images,” *Frontiers in Neuroinformatics*, vol. 12, 2018.
- [15] Y. Huang, J. Xu, Y. Zhou, T. Tong, and X. Zhuang, “Diagnosis of Alzheimer’s disease via multi-modality 3d convolutional neural network,” *Frontiers in Neuroscience*, vol. 13, 2019.
- [16] S. Liu, C. Yadav, C. Fernandez-Granda et al., “On the design of convolutional neural networks for automatic detection of Alzheimer’s disease,” 2020, <https://arxiv.org/abs/1911.03740v3>.
- [17] P. Zhou, S. Jiang, L. Liu et al., “Use of a sparse-response deep belief network and extreme learning machine to discriminate Alzheimer’s disease mild cognitive impairment, and normal controls based on amyloid PET/MRI images,” *Frontiers of Medicine*, vol. 7, 2021.
- [18] S. Liu, S. Liu, W. Cai et al., “Multimodal neuroimaging feature learning for multiclass diagnosis of alzheimer’s disease,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 4, pp. 1132–1140, 2014.
- [19] E. Yee, K. Popuri, and M. F. Beg, “Quantifying brain metabolism from FDG-PET images into a probability of Alzheimer’s dementia score,” *Human Brain Mapping*, vol. 41, no. 1, pp. 5–16, 2020.
- [20] X. Pan, T.-L. Phan, M. Adel et al., “Multi-view separable pyramid network for AD prediction at MCI stage by 18F-FDG brain PET imaging,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 81–92, 2021.
- [21] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [22] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” 2016, <https://arxiv.org/abs/1607.08022>.
- [23] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: convolutional block attention module,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Glasgow, UK, August 2018.
- [24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [25] G. Huang, Z. Liu, V. D. M. Laurens et al., “Densely connected convolutional networks,” in *Proceedings the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, January 2016.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [27] N. Srivastava, G. Hinton, A. Krizhevsky et al., “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [28] R. Muler, S. Kornblith, and G. Hinton, “When does label smoothing help?,” 2019, <https://arxiv.org/abs/1906.02629>.
- [29] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” in *Advances in Kernel Methods-Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., MIT Press, Cambridge, MA, USA, 1998.
- [30] *ADNI Database*, <http://adni.loni.usc.edu/>, 2020.

## Research Article

# Diversity Evolutionary Policy Deep Reinforcement Learning

Jian Liu <sup>1,2</sup> and Liming Feng<sup>1,2</sup>

<sup>1</sup>*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China*

<sup>2</sup>*Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China*

Correspondence should be addressed to Jian Liu; liujiansqjxt@126.com

Received 19 June 2021; Revised 10 July 2021; Accepted 19 July 2021; Published 4 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Jian Liu and Liming Feng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The reinforcement learning algorithms based on policy gradient may fall into local optimal due to gradient disappearance during the update process, which in turn affects the exploration ability of the reinforcement learning agent. In order to solve the above problem, in this paper, the cross-entropy method (CEM) in evolution policy, maximum mean difference (MMD), and twin delayed deep deterministic policy gradient algorithm (TD3) are combined to propose a diversity evolutionary policy deep reinforcement learning (DEPRL) algorithm. By using the maximum mean discrepancy as a measure of the distance between different policies, some of the policies in the population maximize the distance between them and the previous generation of policies while maximizing the cumulative return during the gradient update. Furthermore, combining the cumulative returns and the distance between policies as the fitness of the population encourages more diversity in the offspring policies, which in turn can reduce the risk of falling into local optimal due to the disappearance of the gradient. The results in the MuJoCo test environment show that DEPRL has achieved excellent performance on continuous control tasks; especially in the Ant-v2 environment, the return of DEPRL ultimately achieved a nearly 20% improvement compared to TD3.

## 1. Introduction

Reinforcement learning [1, 2], as an important branch of machine learning [3, 4], has always been a research hotspot. Reinforcement learning constantly improves its policy by interacting with the actual environment, so that the policy can get the maximum cumulative return in the current environment. In recent years, deep learning has exerted more and more influence on various research fields. The combination of deep learning and reinforcement learning produces a variety of deep reinforcement learning algorithms. Deep reinforcement learning can be divided into three types: value-based deep reinforcement learning [5–7], policy-based deep reinforcement learning [8], and deep reinforcement learning based on actor-critic structure [9–11].

Value-based deep reinforcement learning methods estimate the value function through a neural network and use the value function output by the neural network to guide the

agent to choose policies, such as deep Q network (DQN) algorithm [12]. Policy-based deep reinforcement learning methods can parameterize policies and achieve policy optimization through learning parameters, so that the agent can obtain the largest cumulative return, such as deterministic policy gradient (DPG) algorithm [5]. This type of algorithm has good performance when dealing with high-dimensional continuous space problems, but it is easy to cause gradient disappearance in the process of policy update and then fall into the local optimal solution problem [8]. Deep reinforcement learning methods based on actor-critic structure combine value-based and policy-based methods to learn policies while fitting value functions, such as deep deterministic policy gradient (DDPG) algorithm. Actor network parameters are trained according to the value function output by the critic network, and the critic network parameters are updated in a single step using the time difference (TD) method. Although the actor-critic-based methods have the advantages of both value-based and

policy-based methods, they also inherit the shortcomings of the policy gradient algorithm; that is, the policy update falls into a local optimal solution due to the disappearance of the gradient.

The DDPG algorithm combines the ideas of DQN [12] and DPG [5] to solve tasks under continuous action. As an off-policy actor-critic algorithm, DDPG can be trained with historical data through experience playback pool, which greatly improves the utilization of samples and achieves better results in continuous action tasks. Subsequently, inspired by double DQN [13], twin delayed deep deterministic policy gradient algorithm (TD3) [10] on the basis of DDPG simultaneously uses two critic networks to fit the state action value function. And it takes the minimum value of the two target network outputs as the final estimate. TD3 solves the problem of overestimation of the DDPG median function and improves the stability of the agent. However, since DDPG and TD3 both use a similar way to the policy-based algorithms when updating the policy, they also rely on the gradient information for updating policy, which undoubtedly suffers from the vanishing gradient problem during the update process. By adding a small amount of random noise to the policy output by the neural network, the influence of the disappearance of the gradient on the policy update can be alleviated to a certain extent. For example, NoisyNets [14] enhance the exploration ability of the algorithm by directly adding random noise to the parameters of the neural network. However, since the influence of random noise on the policy is random and nondirectional, the effect of this method is limited. The combination of policy gradient and deep learning can be applied to complex and challenging tasks such as game simulation [15], robot control [16], and dialogue system [17]. However, when the policy gradient methods are applied to the continuous control field, there still exists a basic problem, that is, the local optimal problem caused by the disappearance of gradient in the updating process. Tessler et al. [8] put forward that the generation model can be used to learn policies. In this way, although local optimal problem can be avoided, the difficulty of algorithm training is increased.

Evolutionary policy has been used as a nongradient optimization algorithm for decades and performs well in some reinforcement learning benchmark environments. Compared with gradient optimization, the evolution policy is simpler to implement, uses fewer hyperparameters, does not require gradient information, is easier to expand in a distributed environment, and is less affected by sparse rewards. Wierstra et al. [18] proposed Natural Evolution Policies (NES), which optimizes the policy by searching for the distribution of parameters and uses natural gradients to update the distribution in the direction of higher fitness. Inspired by the NES, Tim et al. [19] used the NES as a nongradient black box optimizer to find the optimal policy parameters. Khadka and Tumer [20] proposed evolutionary reinforcement learning (ERL) by effectively combining the evolutionary algorithm based on population with DDPG. Based on ERL, Pourchot and Sigaud [21] combined the cross-entropy method (CEM) with reinforcement learning and proposed CEM-RL method, which further improved the performance of the algorithm.

At present, most of the algorithms that combine reinforcement learning and evolutionary policy only make use of the cumulative return information of policies in each generation population but do not make full use of the distance information of policies between different generations. Effectively increasing the distance between policies of different generations is conducive to the generation of diverse policies for future generations and can improve the exploration of the environment by the reinforcement learning agent. Simultaneously, compared with the single policy, the diverse policies can effectively reduce the risk of falling into the local optimal solution in the updating process. Therefore, in this paper, a diversity evolutionary policy deep reinforcement learning (DEPRL) algorithm is proposed. DEPRL uses maximum mean discrepancy (MMD) to measure the distance between different policies. In the contemporary population, some policies maximize the cumulative return while maximizing the distance from the previous generation policy during the gradient update process. In the process of population evolution, the distance information and cumulative return of the policy are taken as the fitness of the population. The difference between the new generation policy and the previous generation policy is enlarged on the basis of ensuring the higher cumulative return of the new generation policy. By diversifying the policies in the population, DEPRL reduces the risk that the algorithm will fall into local optimum due to the disappearance of gradient in the process of updating and improves the exploration efficiency of agents. Finally, the effectiveness of DEPRL in continuous action task is verified by MuJoCo simulation environment.

The remainder of this paper is organized as follows. The next section describes the related works of DEPRL method. Section 3 represents the framework and details of DEPRL method. Then, in Section 4, a series of comparison experiments on MuJoCo test environment are conducted. The final section provides our concluding remarks and points out our future work orientation.

## 2. Related Works

*2.1. Markov Decision Process (MDP).* In reinforcement learning, the interaction process between reinforcement learning agents and the environment can be represented by Markov decision process (MDP). MDP can be represented by a tuple  $M = (S, A, R, P^f, \gamma)$ , where  $S$  is the state space,  $A$  is the action space,  $R$  is the reward function,  $P^f$  is the state transition probability, and  $\gamma \in [0 \sim 1]$  is the discount factor. When the agent interacts with the environment, the way of choosing an action is called an action policy. Generally, the action policy can be a random policy or a deterministic policy. The random policy  $\pi$  is a probability value, which represents the probability that the agent chooses different actions from the action space in the state  $S$ , and the deterministic policy  $\pi_{\eta}$  represents the choice of a certain action in the state  $S$ . In each time step, the agent observes the current state  $s_t \in S$  according to the environment and chooses action  $a_t \sim \pi(s_t)$  according to the policy to get the reward  $r_t = r(s_t, a_t)$  of the environment feedback.

Subsequently, the agent enters the next state according to the state transition probability  $P^s$ . The goal of reinforcement learning is to train the agent so that the agent finds an optimal policy  $\pi^*$  that can obtain the largest cumulative return.

**2.2. Cross-Entropy Method (CEM).** Evolutionary algorithms update the population by managing a finite number of individuals and generating new individuals near the previous elite sample. Some evolutionary algorithms are temporary optimization methods based on heuristics, such as genetic algorithm (GA) [22]. And the other part is based on the distribution algorithm that estimates the elite sample, such as estimation of distribution algorithms (EDA) [23, 24]. Cross-entropy method (CEM) is a simple EDA algorithm. Suppose that the total number of individuals in the population is  $K$ , where the total number of elite individuals is fixed at a certain value  $K_e$ , which is usually set to half of the total number of individuals in the population. After evaluating all the individuals in the population, the first  $K_e$  outstanding individuals are used to calculate the new mean and variance of the population. Then, additional variance is added to prevent premature convergence, and the next generation is sampled from the new population. A new distribution is obtained by adding Gaussian noise  $\varepsilon$  around the average value  $\mu$  of the distribution, so that each individual  $(x_i)_{i=1,\dots,K}$  is sampled from this new distribution, that is,  $x_i \sim \mathcal{N}(\mu, \Sigma)$ , where  $\Sigma$  represents the current covariance matrix. By calculating the fitness of these newly generated individuals related to specific problems, CEM uses the best performing  $K_e$  individuals  $(z_i)_{i=1,\dots,K_e}$  to update the distribution parameters.

**2.3. Neural Networks.** In recent years, many neural networks, such as extreme learning machine (ELM) [25], probabilistic neural network (PNN) [26], and deep neural networks (DNN) [27], have been proposed and applied in many research fields. For example, Yi et al. [26] proposed a self-adaptive probabilistic neural network (SaPNN) method for transformer fault diagnosis problem. SaPNN can select the best spread self-adaptively all the time and always get the best prediction accuracy. To improve the accuracy and usefulness of target threat assessment in the aerial combat, Wang et al. proposed Elman-AdaBoost strong predictor [28] and multiple wavelet function wavelet neural networks (MFWFNN) [29] to solve threat assessment. Elman-AdaBoost strong predictor uses the Elman neural network as a weak predictor and obtains a strong predictor composed of multiple Elman neural network weak predictors through the Elman-AdaBoost algorithm. In [29], a wavelet mother function selection algorithm was proposed with minimum mean squared error and used to construct MFWFNN network. Cui et al. [30] proposed a novel method that used convolutional neural network (CNN) to improve the detection of malware variants. They converted the malicious code into grayscale images and used CNN to identify and classify the images.

Neural networks can also be applied to reinforcement learning. Traditional reinforcement learning is limited to small action space and sample space, which are generally discrete. However, more complex and more realistic tasks often have a large state space and continuous action space. When the input data is image or sound, it usually has a very high dimension, which is difficult for traditional reinforcement learning to deal with. Deep reinforcement learning is to combine the high-dimensional input of deep neural networks with reinforcement learning. Deep Q network (DQN) [12] can be regarded as the beginning of the successful combination of the two. It uses a deep network to represent the value function. Based on Q-learning in reinforcement learning, it provides target values for the deep network and constantly updates the network until convergence. After that, many deep reinforcement learning algorithms have been proposed, such as double DQN [13], DPG [5], and TD3 [10].

**2.4. Twin Delayed Deep Deterministic Policy Gradient Algorithm (TD3).** Both DDPG and TD3 are off-policy reinforcement learning algorithms based on the actor-critic structure. DDPG is easy to cause the problem of overestimation of value function, which affects the stability of algorithm. To mitigate the negative effects of overestimation, TD3 uses both critic networks to estimate the state action values and takes the minimum value of the two target network outputs as the final estimate.

In order to make the parameters of actor and critic networks updated stably, TD3 makes the updating frequency of network parameters of actor network lower than that of critic network during the training process. TD3 also adds random noise to the action output by the target policy, which not only improves the agent's exploration ability, but also fits the state action value of a small area around the target action. TD3 makes the value function learned by critic network smoother in the action dimension. Since the update direction of actor network parameters is affected by the value function learned from the critic network, the policy learned from actor network also tends to be smoother in the action dimension. By adding random noise, TD3 improves the stability of the agent during training process. The calculation formula of the action value of the target state in TD3 is as follows:

$$y(r, s') = r + \gamma \min_{i=1,2} Q_{\phi_i}(s', \pi_{\theta}(s') + \varepsilon), \quad (1)$$

$$\varepsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c).$$

### 3. Methods

**3.1. Diversity Evolutionary Policy Deep Reinforcement Learning (DEPRL).** The objective function of DEPRL mainly includes the objective function of critic network and actor network. To mitigate the impact of overestimation of the value function, critic network takes the minimum value of the two target network outputs to calculate the final target value. Assuming that  $\theta_1$  and  $\theta_2$  represent the estimated



network parameters of the two critic networks,  $\theta_{\text{target},1}$  and  $\theta_{\text{target},2}$  represent target network parameters of the two critic networks. Then, the update process of the critic networks in DEPRL is shown in Figure 1. The target value of state action under time steps  $t$  is

$$Y(s_t, a_t) = r + \gamma \min_{i=1,2} Q_{\theta_{\text{target},i}}(s_{t+1}, \pi_\phi(s_{t+1})), \quad (2)$$

where  $r$  is the reward to the environment,  $Q_{\theta_{\text{target},i}}(s_{t+1}, \pi_\phi(s_{t+1}))$  represents the target network output value of the  $i$ -th critic network,  $\phi$  represents the network parameters of the actor network, and  $\gamma$  is the discount factor. Assume that  $Q_{\theta_i}(s_t, a_t)$  represents the estimated value output by the  $i$ -th estimation network under the number of time steps  $t$ , and then the objective function of critic network can be written as

$$J_Q(\theta_i) = \mathbb{E}_{(s_t, a_t) \sim D} \left[ \frac{1}{2} (Q_{\theta_i}(s_t, a_t) - Y(s_t, a_t))^2 \right]. \quad (3)$$

Therefore, the estimated network parameters  $\theta_1$  and  $\theta_2$  can minimize the objective function  $J_Q(\theta_i)$  through gradient descent. That is, gradient descent is used to minimize the mean square error between the estimate and the target value:

$$\begin{aligned} \theta_1 &\leftarrow \theta_1 - \alpha \nabla_{\theta_1} \frac{1}{2} (Q_{\theta_1}(s_t, a_t) - Y(s_t, a_t))^2, \\ \theta_2 &\leftarrow \theta_2 - \alpha \nabla_{\theta_2} \frac{1}{2} (Q_{\theta_2}(s_t, a_t) - Y(s_t, a_t))^2, \end{aligned} \quad (4)$$

where  $\alpha$  represents the update step size. In the process of gradient updating, the target network parameters  $\theta_{\text{target},1}$  and  $\theta_{\text{target},2}$  are kept constant to ensure the stability of updating.

After the estimated network parameters are updated, the parameters of the target network are updated by soft update method. The formula is as follows:

$$\theta_{\text{target},1} \leftarrow \tau \theta_1 + (1 - \tau) \theta_{\text{target},1}, \quad (5)$$

$$\theta_{\text{target},2} \leftarrow \tau \theta_2 + (1 - \tau) \theta_{\text{target},2}, \quad (6)$$

where  $\tau$  is the coefficient of soft update method. For the parameter  $\phi$  of actor network, the gradient update direction is to maximize the distance between the current policy and  $\pi_\eta$  while maximizing the cumulative return. The distance between  $\pi_\eta$  and the current policy can be calculated by using the square of the maximum mean discrepancy (MMD).

Given samples  $x_1, \dots, x_n \sim P$  and  $y_1, \dots, y_m \sim G$ , the square of the MMD can be estimated only from the sample of the distribution. Then, the square of MMD between distribution  $P$  and  $G$  can be written as

$$\text{MMD}^2(\{x_1, \dots, x_n\}, \{y_1, \dots, y_m\}) = \frac{1}{n^2} \sum_{i,i'} k(x_i, x_{i'}) - \frac{2}{nm} \sum_{i,j} k(x_i, y_j) + \frac{1}{m^2} \sum_{j,j'} k(y_j, y_{j'}), \quad (7)$$

where  $k(\cdot, \cdot)$  is the kernel function. Here, Gaussian kernel is used in DEPRL, that is,

$$k(x_i, x_{i'}) = \exp\left(-\frac{\|x_i - x_{i'}\|^2}{2\sigma^2}\right), \quad \sigma > 0, \quad (8)$$

where  $\sigma$  is standard deviation. Record the square of MMD between policy  $\pi_\eta$  and policy  $\pi_\phi$  as  $D_{\text{MMD}}(\pi_\eta, \pi_\phi)$ , and the formula is as follows:

$$D_{\text{MMD}}(\pi_\mu, \pi_\phi) = \text{MMD}^2(\pi_\mu(\cdot|s), \pi_\phi(\cdot|s)) \quad s \sim D, \quad (9)$$

where  $D$  is the experience pool.

To sum up, the objective function of actor network only considering the maximum cumulative return is

$$J_\pi(\phi) = E_{s \sim D, a \sim \pi_\phi(\cdot|s)} [Q_{\theta_1}(s, a)]. \quad (10)$$

When  $D_{\text{MMD}}(\pi_\eta, \pi_\phi)$  that satisfies the gradient update requirement is obtained, the objective function of the actor network can be written as

$$\begin{aligned} J_{\text{MMD}}(\phi) &= \mathbb{E}_{s \sim D, a \sim \pi_\phi(\cdot|s)} [Q_{\theta_1}(s, a)] \\ &+ \beta \mathbb{E}_{s \sim D} [\text{MMD}^2(\pi_\mu(\cdot|s), \pi_\phi(\cdot|s))], \end{aligned} \quad (11)$$

where  $\beta > 0$  is the weighting factor. The number of actors that only consider cumulative returns is recorded as  $K_1$ , and then the number of actors that maximize  $D_{\text{MMD}}(\pi_\eta, \pi_\phi)$  at the same time is  $K/2 - K_1$ .

**3.2. The Framework of DEPRL.** In CEM-RL method, the total number of individuals in the population is set to  $K$ . The mean  $\mu$  and covariance matrix  $\Sigma$  of the policy parameter distribution are obtained by random initialization. According to the covariance matrix and the mean value,  $K$  parameters are extracted from the distribution as the parameters of actor network in the population. The actor network with half of the total number of individuals in the population is randomly selected for gradient update according to the value function output from critic network. The goal is to maximize the cumulative return of the actor network's corresponding policy. The critic network that guides actor network gradient updates throughout the process is the same; that is, half of the actors in the

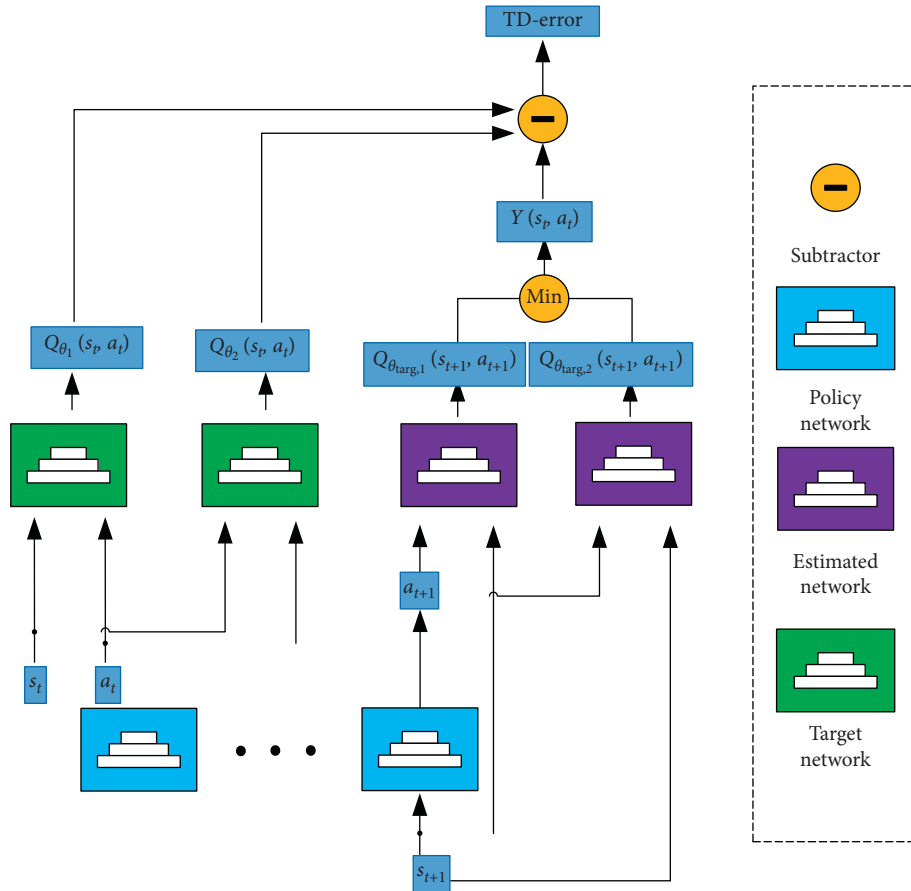


FIGURE 1: The update process of critic networks in DEPRL.

population use the same critic network to guide updates. In a population, the data generated by the interaction between the actor and the environment is stored in the experience pool and is used to train the critic network. By evaluating the cumulative returns of the policies corresponding to all actors in the population after gradient updating, the policies ranked in the top half of the cumulative returns are selected as the elite sample. The number of the elite sample  $K_e$  is usually set to  $K_e = K/2$ . Finally, according to the parameters of contemporary elite samples,  $\mu_{\text{new}}$  and  $\Sigma_{\text{new}}$  of the new generation actor network parameter distribution are generated.

The framework of DEPRL algorithm is shown in Figure 2. Assume that the corresponding policy of Actor  $\mu$  composed of elite sample parameters is  $\pi_{\eta}$ . When the critic network guides the next generation policy update, it needs to maximize the MMD between a part of policies and  $\pi_{\eta}$ . By increasing the diversity of descendant policies, more space is explored, and the probability of the algorithm falling into the local optimal solution is reduced. When selecting the elite sample, not only the cumulative return of each policy should be considered, but also the MMD between each policy and  $\pi_{\eta}$  should be considered. In the population, the updated new policy is first sorted according to the cumulative return from high to low, and the policies with cumulative return ranked between 2 and  $K/2$  greater than  $\pi_{\eta}$  cumulative return are taken out, and the MMD values between these policies and

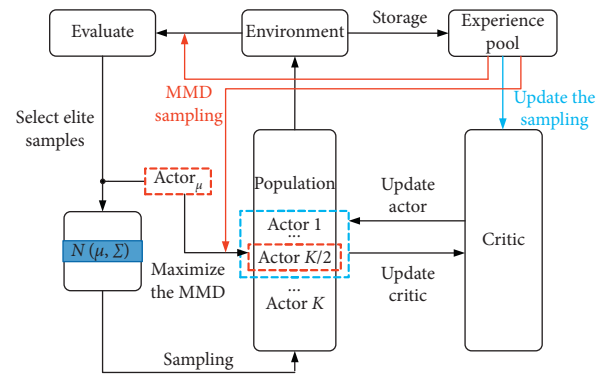


FIGURE 2: The framework of DEPRL.

$\pi_{\eta}$  are calculated, and reorder the MMD value from largest to smallest. In the population, the updated new policy is first sorted according to the cumulative return from high to low. Then, the policies in which the cumulative return is between 2 and  $K/2$  greater than the cumulative return of  $\pi_{\eta}$  are taken out. Finally, the MMD values between these policies and  $\pi_{\eta}$  are calculated. These policies are reordered in descending order of MMD value.

Use MMD as the standard to select policies that is quite different from  $\pi_{\eta}$  among contemporary policies, which helps transfer the diversity policy to the next generation

distribution. The new generation policy generated by sampling in the new distribution is quite different from the old policy, which makes the trajectory of the new generation policy more diversified and can increase the exploration space. In order to reduce the amount of calculation when calculating the new distribution parameters,  $\Sigma$  is constrained to be a diagonal matrix. The update formulas of the new distribution parameters  $\mu_{\text{new}}$  and  $\Sigma_{\text{new}}$  are as follows:

$$\mu_{\text{new}} = \sum_{i=1}^{K_e} \lambda_i z_i, \quad (12)$$

$$\Sigma_{\text{new}} = \sum_{i=1}^{K_e} \lambda_i (z_i - \mu_{\text{old}})(z_i - \mu_{\text{old}})^T + \varepsilon, \quad (13)$$

where  $\lambda_i$  represents the weight of the parameter corresponding to the  $i$ -th elite policy in the population, and  $\varepsilon$  is the Gaussian noise.  $\lambda_i$  can be defined as

$$\lambda_i = \frac{(\log(1 + K_e)/i)}{\sum_{i=1}^{K_e} (\log(1 + K_e)/i)}. \quad (14)$$

The above formula indicates that the higher the ranking of the parameters corresponding to the elite policy, the greater the value of a  $\lambda_i$ .

To sum up, the update process of DEPRL can be simply summarized as follows: (1) the parameter distribution of the initialization policy is  $N(\mu_0, \Sigma_0)$ ; (2)  $K$  group policies are randomly selected corresponding to  $K$  group parameters from the distribution; (3) gradient updating is performed by randomly selecting  $K/2$  policy; (4) the fitness of the corresponding policy under the  $K$  set of parameters is calculated; (5) the parameters corresponding to the current elite policy are used to calculate the parameter distribution  $(\mu, \Sigma)$  of the next generation policy, as shown in equations (12) and (13); (6) whether the parameter distribution of the contemporary policy meets the requirements is determined; if so, stop updating; if not, repeat step (2).

The pseudocode of DEPRL algorithm is shown in Algorithm 1.

## 4. Results and Analysis

**4.1. Experiment Settings.** In this section, we use the MuJoCo test environment implemented in OpenAI Gym [31] to evaluate the performance of the proposed algorithm and comparison Algorithms. Gym is a basic platform for testing deep reinforcement learning algorithms provided by OpenAI. It provides a large number of simple interfaces for the training of the agent, greatly simplifies the interaction process between the agent and the environment, and facilitates related researchers to implement deep reinforcement learning algorithms and test the performance of deep reinforcement learning algorithms. Figure 3 shows the corresponding status screens of the four tasks in the MuJoCo test environment. Table 1 describes the state dimension and action dimension of the four tasks in the MuJoCo test environment, as well as specific task goals. According to the state dimension and action dimension information provided

by MuJoCo, it is convenient to design the corresponding neural network for learning. The version of OpenAI Gym used in the experiment is 0.17.3, and the version of MuJoCo is 2.0.

Experiment settings are set up as follows:

- (1) We chose to compare TD3, multiactor TD3, CEM, and CEM-TD3 to verify the superiority of the proposed DEPRL. The common superparameter settings of the five algorithms are the same as shown in Table 2, and the total numbers of population individuals and elite individuals of CEM-TD3 and DEPRL are the same, 10 and 5, respectively. When DEPRL calculates  $D_{\text{MMD}}$ , the data size  $M$  extracted from the experience pool is 600, the number of Gaussian kernel function  $m = n = 5$ , and the value of  $K_1$  is 4. The weighting factor  $\beta$  in the objective function  $J_{\text{MMD}}$  is 0.2 in the Ant-v2 environment, and 0.1 in all other test environments.
- (2) In order to make a fair comparison between different algorithms, we combined CEM and TD3 to form CEM-TD3 algorithm for experiment. And the network structure used by CEM to represent policies is consistent with that of DEPRL, CEM-TD3, multiactor TD3 and TD3. Multiactor TD3 is a variant of TD3. Compared with TD3, multiactor TD3 has multiple actors. The experience data generated by the interaction between multiple actors and the environment are sent to the experience pool together, and the critic remains unchanged. In the experiment, the number of actors in multiactor TD3 is set to 5, and the total number of gradient updates of critic and actor in multiactor TD3, CEM-TD3, and DEPRL is the same.
- (3) We selected four environments HalfCheetah-v2, Hopper-v2, Walker2d-v2, and Ant-v2 for comparison, and the details of the test environment are shown in Table 1. The experimental results are shown in Figure 4, where the horizontal axis represents the number of time steps, and the vertical axis represents the cumulative return value of a round in the evaluation stage. During the training process, the performance of the current algorithm is evaluated every 1000 steps. Each algorithm was repeated with five different random seeds in different test environments. When drawing the reward curve, the sliding window size is set to 100. The curve part and shaded part in the figure represent the mean value and the standard deviation of the accumulated return value under multiple random seeds, respectively. We also present the mean and standard deviation of the cumulative return per turn in different MuJoCo tasks. The results can be found in Table 3.

### 4.2. Analysis of Experimental Results

- (1) As can be seen from Figure 4, DEPRL performs best overall in the test environment and also performs best in the environment with higher state dimension

**Input:** the coefficient of soft update method  $\tau$ , sampling size of the experience pool  $N$  and  $M$ , maximum number of time steps  $T_{\max}$ , discount factor  $\gamma$ , experience pool capacity  $\Delta_{\text{size}}$ , population parameter  $K$  and  $K_1$

**Output:** actor network parameters  $\phi^*$  corresponding to the optimal policy  $\pi^*$

- (1) Initialize critic network parameters  $\theta_1, \theta_2, \theta_{\text{targ},1}, \theta_{\text{targ},2}$  and actor network parameter distribution  $(\mu_0, \Sigma_0)$
- (2)  $T_{\text{total}} = 0, T_{\text{actor}} = 0$
- (3) **WHILE**  $T_{\text{total}} < T_{\max}$ :
- (4) Extract  $K$  sets of parameters  $para$  from the current distribution  $(\mu, \Sigma)$
- (5) **FOR**  $k = 1$  **TO**  $K/2$ :
- (6) Initialize the actor according to the parameter  $para[k]$
- (7) **FOR**  $t = 1$  **TO**  $2 * T_{\text{actor}}/K$ :
- (8) Sampling  $N$  samples from  $\Delta$  to minimize the objective function (3)
- (9) Update  $\theta_{\text{targ},1}$  and  $\theta_{\text{targ},2}$  through equations (5) and (6)
- (10) **FOR**  $k = 1$  **TO**  $K_1$ :
- (11) Initialize the actor according to the parameter  $para [k]$
- (12) **FOR**  $t = 1$  **TO**  $T_{\text{actor}}$ :
- (13) Sample  $N$  samples from  $\Delta$  to maximize the objective function (11)
- (14) Replace the original parameter  $para [k]$  with the new actor parameter
- (15) **FOR**  $k = K_1 + 1$  **TO**  $K/2$ :
- (16) Initialize the actor according to the parameter  $para [k]$
- (17) **FOR**  $t = 1$  **TO**  $T_{\text{actor}}$ :
- (18) Sample  $N$  samples from  $\Delta$  to maximize the objective function (12)
- (19) Replace the original parameter  $para [k]$  with the new actor parameter
- (20)  $T_{\text{actor}} = 0$
- (21) **FOR**  $k = 1$  **TO**  $K$ :
- (22) Initialize the actor according to the parameter  $para[k]$
- (23) Interact with the environment to calculate the cumulative payoff  $G$  and the total number of time steps used  $T_{\text{episode}}$
- (24) Store data  $(s, a, s', r)$  in the experience pool  $\Delta$
- (25) Sample  $M$  samples from  $\Delta$  to calculate the  $D_{\text{MMD}}$  between them and  $\text{Actor}_\mu$
- (26)  $T_{\text{actor}} = T_{\text{actor}} + T_{\text{episode}}$
- (27)  $T_{\text{total}} = T_{\text{total}} + T_{\text{actor}}$
- (28) Select elite samples according to  $G$  and  $D_{\text{MMD}}$ , and update the distribution according to equations (12) and (13)
- (29) **END WHILE**

ALGORITHM 1: DEPRL.

and action dimension, such as Ant-v2 and Walker2d-v2. CEM performs worst overall and learns few effective policies in environments with higher state and action dimensions. Therefore, it can be shown that both the sample utilization and learning rate of CEM are significantly lower than those of other algorithms based on single-step update.

- (2) In order to explore whether the improvement of DEPRL effect is due to the adoption of multiactor structure, we tested the influence of multiactor structure on the algorithm. Compared with the traditional actor-critic structure, the training data used by the critic in the multiactor structure is generated by the interaction between multiple actors and the environment. By comparing the reward curves of TD3, multiactor TD3, and DEPRL in Figure 4, it can be found that the reward curve of multiactor TD3 is only slightly higher than that of TD3 based on the traditional actor-critic structure. Therefore, it can be explained that the multiactor structure does not improve the algorithm much. In the Hopper-v2 training environment, multiactor TD3 began to oscillate when the cumulative return of the policy reached about 3200 and could not learn a

better policy, while DEPRL with the same multiactor structure could get about 3600 cumulative returns. By comparing the reward curves among TD3, multiactor TD3, and DEPRL, it can be shown that the performance improvement of DEPRL does not simply depend on the multiactor structure.

- (3) To explore the benefits of DEPRL in encouraging offspring diversity, we compared it with CEM-TD3, which only uses cumulative returns as a policy learning goal. CEM-TD3 also uses multiactor structure, and the total number of population individuals and the number of elite individuals is set the same as DEPRL. It can be seen from Figure 4 that the reward curve of DEPRL is significantly higher, and the reward curve of CEM-TD3 gradually levelled off in the second half due to the decline of exploration ability. Except for the Hopper-v2 test environment, DEPRL still maintained a relatively high growth trend in the second half of the reward curve.
- (4) As can be seen from Table 3, the DPRL algorithm has the highest mean cumulative return of all the algorithms. The CEM algorithm performs the worst, which once again demonstrates that CEM, as a turn update algorithm with no experience replay,

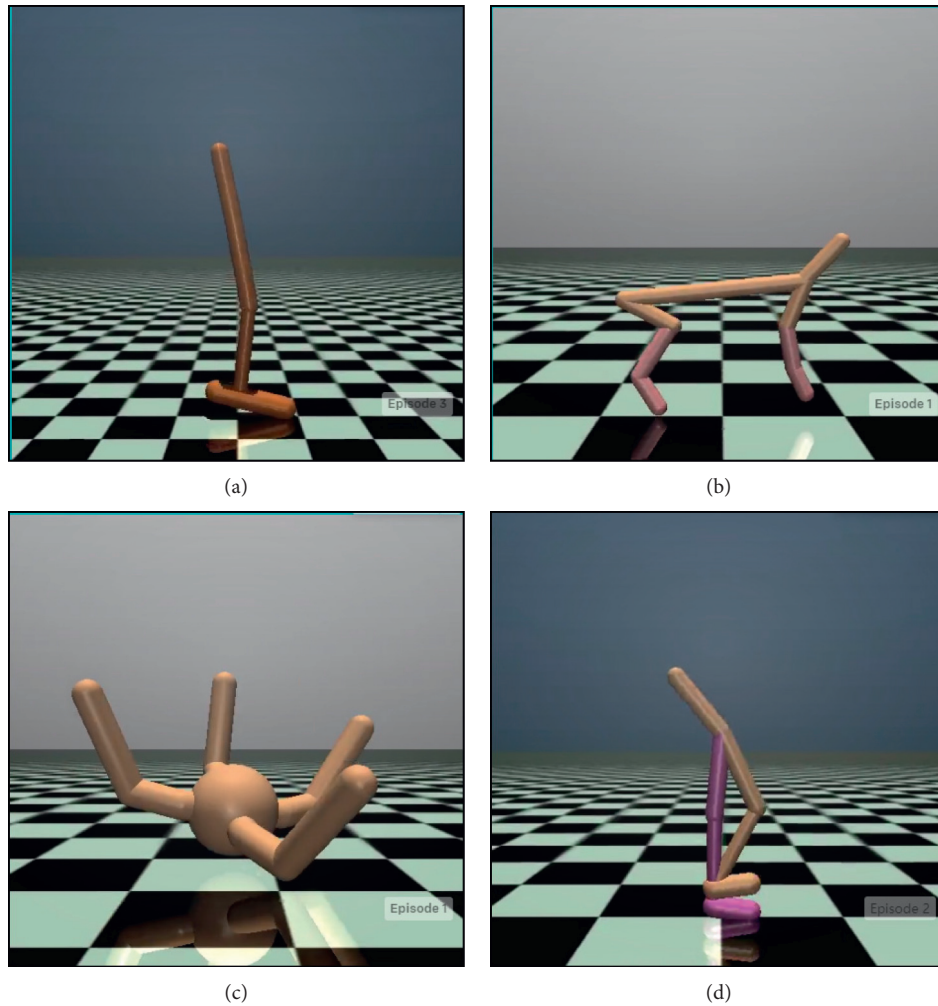


FIGURE 3: MuJoCo test environments. (a) Hopper-v2, (b) HalfCheetah-v2, (c) Ant-v2, and (d) Walker2d-v2.

TABLE 1: The test environment in the MuJoCo benchmark.

Environment	Action dimension/state dimension	Task goals
Hopper-v2	3/11	Make a two-dimensional one-legged robot hop forward as fast as possible
HalfCheetah-v2	6/17	Make the 2D cheetah robot run fast
Ant-v2	8/111	Make a four-legged creature walk forward as fast as possible
Walker2d-v2	6/17	Make a two-dimensional bipedal robot walk forward as fast as possible

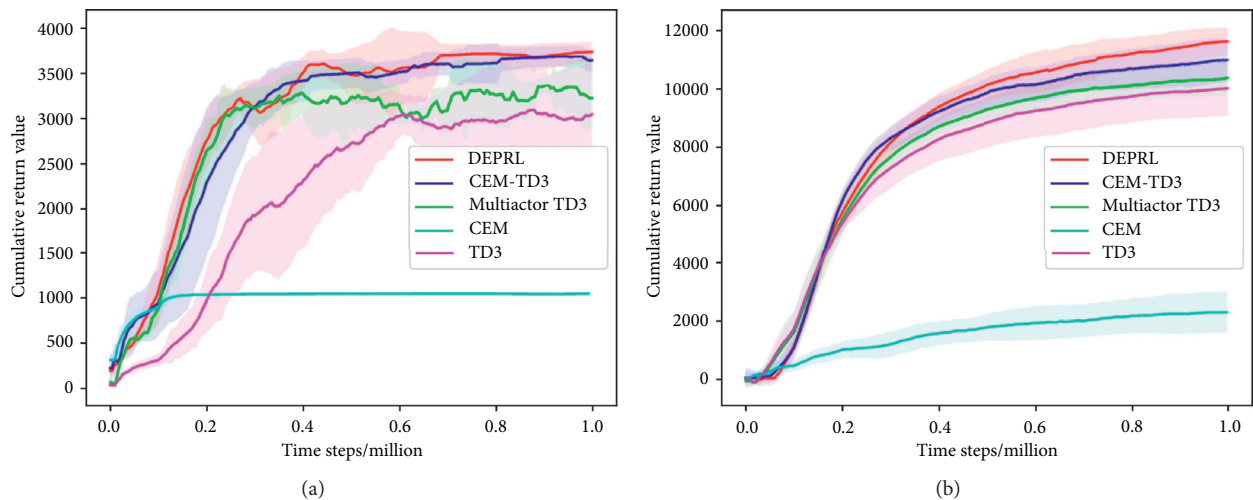


FIGURE 4: Continued.

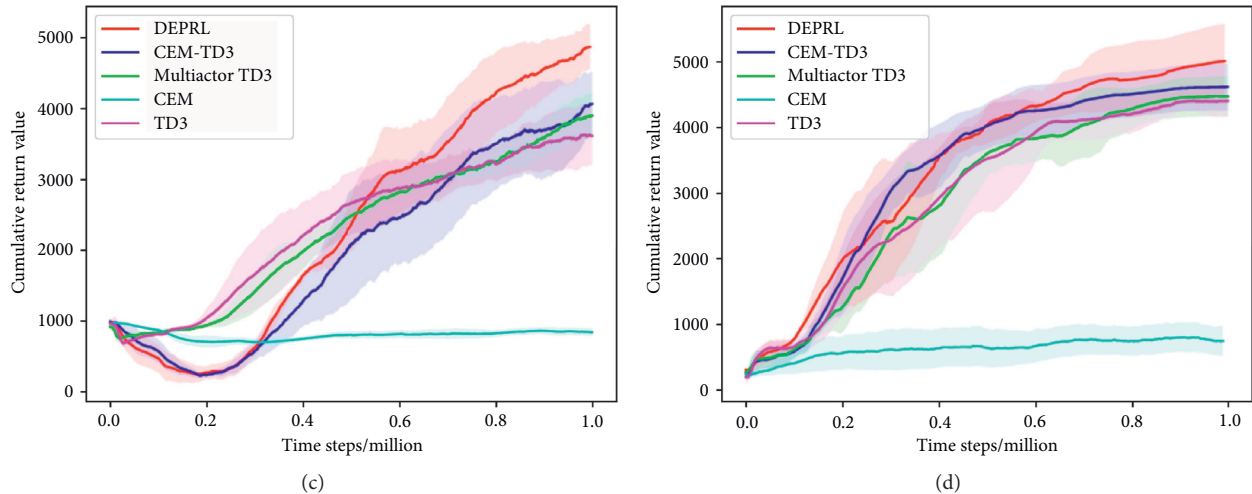


FIGURE 4: Results of each algorithm in MuJoCo test environment. (a) Hopper-v2. (b) HalfCheetah-v2. (c) Ant-v2. (d) Walker2d-v2.

TABLE 2: Values of hyperparameter.

Hyperparameter	Values
Critic/actor learning rate	0.0003
Critic/actor hidden layer	2
Number of neurons	400/300
Critic activation	Relu
Actor activation	Tanh
Discount factor	0.99
Optimizer	Adam
Soft update coefficient	0.005
Experience pool capacity	$10^6$
Experience pool sample size	100
Gauss noise	Clip ((0, 0.2), -0.5, 0.5)

TABLE 3: The mean and standard deviation of the cumulative return per turn in different MuJoCo tasks.

Task	TD3	Multiactor TD3	CEM	CEM-TD3	DEPERL
Hopper-v2	$3025 \pm 577$	$3241 \pm 363$	$1054 \pm 17$	$3652 \pm 116$	$3732 \pm 106$
HalfCheetah-v2	$10002 \pm 930$	$10341 \pm 578$	$2298 \pm 690$	$10978 \pm 758$	$11615 \pm 464$
Ant-v2	$3618 \pm 425$	$3881 \pm 319$	$845 \pm 52$	$4037 \pm 466$	$4852 \pm 317$
Walker2d-v2	$4399 \pm 238$	$4470 \pm 301$	$743 \pm 225$	$4612 \pm 357$	$5001 \pm 562$

could not learn effective strategies. Compared with TD3 and multiactor TD3 algorithms, DEPERL and CEM-TD3 algorithms have higher average cumulative returns, which is due to the addition of evolutionary strategy into DEPERL and CEM-TD3 algorithms. Compared with the CEM-TD3 algorithm, the DEPERL algorithm achieves better results, because it increases exploration by encouraging the generation of diversity strategies in the offspring. In addition, in the Hopper-v2, HalfCheetah-v2, and Ant-v2 test environments, DEPERL has smaller standard deviations than TD3, multiactor TD3, and CEM-TD3 algorithms, which indicates that DEPERL algorithm has more stable results than the other three algorithms. To some extent, this also shows that DEPERL algorithm can explore more effective strategies.

The above results clearly show that DEPERL improves the exploration ability of reinforcement learning agents and, to some extent, reduces the risk of policy updating falling into local optimum due to the disappearance of gradient.

## 5. Conclusions and Discussions

In this paper, we propose the DEPERL algorithm, which combines CEM and TD3 to measure the distance between different policies through MMD method. Some contemporary policies maximize the cumulative return while maximizing the distance between them and the previous generation policies and obtain policies with large differences to increase the scope of exploration. In the course of evolution, combining the cumulative return of a contemporary policy with the distance between the previous generation's policy as fitness helps the next generation's policy have more

diversity based on a higher cumulative return. By combining TD3 with gradient updating and CEM without gradient updating, DEPRL can reduce the risk of policy updating falling into local optimal solution due to gradient disappearance by encouraging the generation of diversified policies in the offspring. By comparing DEPRL with CEM-RL, TD3, CEM, and multiactor TD3 in MuJoCo test environment, the experimental results show that DEPRL achieves more effect without increasing the number of update steps.

In DEPRL, we use an estimation of distribution algorithm to estimate the distribution of the elite samples and then select the elite samples that meet certain conditions to improve the diversification of the elite strategy. Except for estimation of distribution algorithms, some of the most representative computational intelligence algorithms can be used to reinforcement learning. Monarch butterfly optimization (MBO) [32] algorithm generates offspring by migration operator, which can be adjusted by the migration ratio of monarch butterflies. It is followed by tuning the positions for other butterflies by means of butterfly adjusting operator. In reinforcement learning, MBO can adjust the selection of elite samples in the global scope to avoid the loss of potential elite samples. In earthworm optimization algorithm (EWA) [33], the offspring are generated through Reproduction 1 and Reproduction 2 independently, and then, the weighted sum of all the generated offspring is used to get the final earthworm for next generation. Reproduction 1 generates only one offspring by itself that is also special kind of reproduction in nature. Reproduction 2 is to generate one or more than one offspring at one time. EWA can be used to replicate elite samples to ensure the high efficiency of elite strategies in reinforcement learning and speed-up learning. In elephant herding optimization (EHO) [34], the elephants in each clan are updated by its current position and matriarch through clan updating operator. It is followed by the implementation of the separating operator, which can enhance the population diversity at the later search phase. EHO is an appropriate way to increase the diversity of a population. Not only can it be used to eliminate bad reinforcement learning strategies, but it can also be used to add new strategies that did not exist before. Exploration is a vital part of reinforcement learning. Exploratory algorithms in computational intelligence algorithms can provide meaningful guidance for reinforcement learning. For example, slime mould algorithm (SMA) [35] uses adaptive weights to simulate the process of producing positive and negative feedback of the propagation wave of slime mould based on bio-oscillator to form the optimal path for connecting food with excellent exploratory ability and exploitation propensity. According to the moth's phototaxis and Levy flight characteristics, moth search (MS) [36] algorithm can do exploitation and exploration at the same time and ensures local search and global search. Harris Hawks Optimizer (HHO) [37] is a popular population-based nongradient optimization algorithm, which has many active time varying exploration and development stages. It has strong global searching ability.

We only analyzed the possibilities of the above computational intelligence algorithms in reinforcement learning

applications, but these algorithms are not really used in reinforcement learning. Therefore, in the future work, we will devote ourselves to applying computational intelligence algorithms to strategy optimization, exploration enhancement, and acceleration of learning speed in reinforcement learning.

## Data Availability

The data used to support the findings of this study are available from the first author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest in this work.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61906198) and the Natural Science Foundation of Jiangsu Province (Grant no. BK20190622).

## References







- [1] R. Sutton and A. Barto, "Reinforcement learning: an introduction," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 285–286, 2005.
- [2] H. Wang, Y. Gao, and X. G. Chen, "Transfer of reinforcement learning: the state of the art," *Acta Electronica Sinica*, vol. 36, no. S1, pp. 39–43, 2008.
- [3] T. G. Dietterich, "Machine-learning research," *AI Magazine*, vol. 18, no. 4, p. 97, 1997.
- [4] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [5] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the International Conference on Machine Learning*, pp. 387–395, Beijing, China, June 2014.
- [6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, <https://arxiv.org/abs/1707.06347>.
- [7] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," *Computer Science*, vol. 3, pp. 1889–1897, 2015.
- [8] C. Tessler, G. Tennenholtz, and S. Mannor, "Distributional policy optimization: an alternative approach for continuous control," 2019, <https://arxiv.org/abs/1905.09855>.
- [9] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2015, <https://arxiv.org/abs/1509.02971>.
- [10] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018, <https://arxiv.org/abs/1802.09477>.
- [11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, <https://arxiv.org/abs/1801.01290>.
- [12] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

- [13] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," 2016, <https://arxiv.org/abs/1509.06461>.
- [14] M. Fortunato, M. G. Azar, B. Piot et al., "Noisy networks for exploration," 2017, <https://arxiv.org/abs/1706.10295>.
- [15] O. Vinyals, I. Babuschkin, W. M. Czarnecki et al., "Grandmaster level in starcraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
- [16] C. Y. Liu, Y. Q. Tan, C. A. Liu et al., "Application of multi-agent reinforcement learning in robot soccer," *Acta Electronica Sinica*, vol. 38, no. 8, pp. 1958–1962, 2010.
- [17] J. D. Williams, K. A. Atui, and G. Zweig, "Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning," 2017, <https://arxiv.org/abs/1702.03274>.
- [18] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, and J. Schmidhuber, "Natural evolution policies," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, 2014.
- [19] S. Tim, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution policies as a scalable alternative to reinforcement learning," 2017, <https://arxiv.org/abs/1703.03864>.
- [20] S. Khadka and K. Tumer, "Evolutionary reinforcement learning," 2018, <https://arxiv.org/abs/1805.07917>.
- [21] A. Pourchot and O. Sigaud, "CEM-RL: combining evolutionary and gradient-based methods for policy search," 2018, <https://arxiv.org/abs/1810.01222>.
- [22] W. X. Yun, "Research progress of genetic algorithm," *Application Research of Computers*, vol. 4, pp. 1201–1206, 2012.
- [23] L. Pedro and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer, Berlin, Germany, 2001.
- [24] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.
- [25] G.-G. Wang, M. Lu, Y.-Q. Dong, and X.-J. Zhao, "Self-adaptive extreme learning machine," *Neural Computing and Applications*, vol. 27, no. 2, pp. 291–303, 2016.
- [26] J. Yi, J. Wang, G. W. Hauschild, and M. Pelikan, "Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem," *Advances in Mechanical Engineering*, vol. 8, no. 1, pp. 1–13, 2016.
- [27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [28] G. G. Wang, L. H. Guo, H. Duan et al., "The mode and algorithm for the target threat assessment base on elman-adaboost storing predictor," *Acta Electronica Sinica*, vol. 40, no. 5, pp. 901–906, 2012.
- [29] G. G. Wang, L. H. Guo, and H. Duan, "Wavelet neural network using multiple wavelet functions in target threat assessment," *The Scientific World Journal*, vol. 2013, Article ID 632437, 7 pages, 2013.
- [30] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.
- [31] G. Brockman, V. Cheung, L. Pettersson et al., "Openai gym," 2016, <https://arxiv.org/abs/1606.01540>.
- [32] G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.
- [33] G. G. Wang, S. Deb, and L. D. S. Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 12, no. 1, pp. 1–22, 2018.
- [34] G. Wang, S. Deb, and L. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational And Business Intelligence (ISCBI)*, Bali, Indonesia, December 2015.
- [35] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, no. 3, pp. 300–323, 2020.
- [36] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.
- [37] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.



## Research Article

# A Defect Detection Method for Rail Surface and Fasteners Based on Deep Convolutional Neural Network

Danyang Zheng <sup>1</sup>, Liming Li <sup>1,2</sup>, Shubin Zheng <sup>1</sup>, Xiaodong Chai <sup>1</sup>,  
Shuguang Zhao <sup>2</sup>, Qianqian Tong <sup>1</sup>, Ji Wang <sup>1</sup>, and Lizheng Guo<sup>3</sup>

<sup>1</sup>School of Urban Railway Transportation, Shanghai University of Engineering Science, Shanghai 201620, China

<sup>2</sup>School of Information Science and Technology, Donghua University, Shanghai 201620, China

<sup>3</sup>School of Computer and Data Science, Henan University of Urban Construction, Pingdingshan 467036, Henan, China

Correspondence should be addressed to Liming Li; [liliming@sues.edu.cn](mailto:liliming@sues.edu.cn)

Received 9 May 2021; Revised 30 June 2021; Accepted 20 July 2021; Published 2 August 2021

Academic Editor: Nian Zhang

Copyright © 2021 Danyang Zheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a result of long-term pressure from train operations and direct exposure to the natural environment, rails, fasteners, and other components of railway track lines inevitably produce defects, which have a direct impact on the safety of train operations. In this study, a multiobject detection method based on deep convolutional neural network that can achieve nondestructive detection of rail surface and fastener defects is proposed. First, rails and fasteners on the railway track image are localized by the improved YOLOv5 framework. Then, the defect detection model based on Mask R-CNN is utilized to detect the surface defects of the rail and segment the defect area. Finally, the model based on ResNet framework is used to classify the state of the fasteners. To verify the robustness and effectiveness of our proposed method, we conduct experimental tests using the ballast and ballastless railway track images collected from Shijiazhuang-Taiyuan high-speed railway line. Through a variety of evaluation indexes to compare with other methods using deep learning algorithms, experimental results show that our method outperforms others in all stages and enables effective detection of rail surface and fasteners.

## 1. Introduction

In recent years, rail transportation has become one of the most important modes of travel. As the total mileage of rail transit continues to increase, how to ensure safe railway operation has become a dominant issue that has attracted public attention. As shown in Figure 1, the rail is the main component of the railway track and is utilized to guide the wheels of the train forward and bear the pressure of the wheel set. The rail and its fasteners in the service are affected by contact forces such as extrusion and impact of the train wheel-rail, poor environment, and material aging. These problems have led to the continuous deterioration of railways, inducing the formation of rail surface defects such as peeling, collapse, abrasion, and corrosion, as well as fastener defects such as fracture and loosening [1]. Research results show that many rail fractures or train derailments are caused

by rail surface or fastener defects. Therefore, it is crucial to ensure that the rail and its fasteners are in a healthy state that maintains the safety and stability of train operation. At present, state detection of the rail and its fasteners on the railway track line is mainly conducted through inspections by railway staff. Although this inspection method has the advantages of simplicity and low cost, it also has disadvantages such as low detection efficiency, high missed detection rate, and poor real-time performance. In recent years, defect detection technology based on computer vision has been widely used in industry [2–6]. Some scholars have begun to employ computer vision technology to detect the defects of rails and their fasteners, so that the problems of manual inspection can be solved.

Using a localization algorithm is necessary to improve the accuracy of defect detection and localize the track components to be inspected, so that the influence of

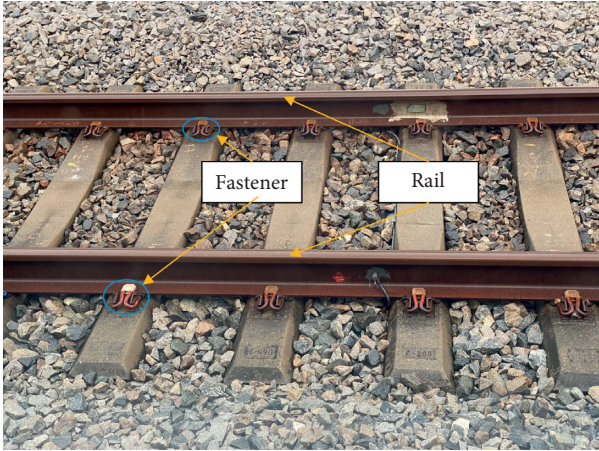


FIGURE 1: Railway track line.

redundant information such as background can be reduced. Commonly used localization methods are template matching [7], pixel statistics [8], and edge detection [9]. However, localization methods of pixel statistics and edge detection are susceptible to uneven lighting and complex backgrounds. The traditional template matching method is difficult to use in localizing deformed or damaged track parts. To solve this problem and localize the track fasteners, Qiu et al. [10] proposed a double-template matching method. First, the rail template is used to localize the rail in the horizontal direction and then use the fastener template to localize the fastener in the vertical direction. In addition, Li et al. [11, 12] used the geometric characteristics of track components to localize fasteners, and Wei et al. [13] used the variance projection and wavelet transform to localize the edges of the rail, fasteners, and backing plates based on the fixed positional relationship between the track components.

The detection method of rail fasteners based on traditional vision mainly uses artificially designed features to extract the features within the fastener area and then inputs the extracted features into a classification model based on shallow learning to classify the state of the fasteners. The shallow features used in the research articles on fastener detection mainly include Haar-like feature [7, 14], Dense-SIFT feature [13], direction field feature [15], edge feature [16], HOG feature [17], Gabor filter feature [18], and Hough transform feature [11, 12]. Classification models mainly include AdaBoost classifier [7, 19], support vector machine (SVM) [17, 18, 20], probabilistic graphical models (PGM) [13], and multilayered perception neural classifier [21, 22]. However, this type of detection method extracts features for the fastener area rather than the detection object. The extracted features are susceptible to the influence of background information, with low robustness and low accuracy for the identification of fasteners in abnormal states. In recent years, as the application of deep learning technology in image processing has achieved great success, many scholars have also begun to try to apply deep learning technology to rail fastener detection. Li et al. [23] used a method based on semantic segmentation algorithm to detect the state of fasteners. First, the saliency model is used to

localize the track fastener area, and then PSPNet is used to semantically segment the fastener subimages. Finally, the state of the fastener is judged by the vector geometry measurements of the fastener. Gibert et al. [24] used a customized fully convolutional network to extract the highly abstract features of fasteners and identify fastener types and then utilized customized support vector machines to classify the state of fasteners for various types of fasteners. Ma et al. [25] cropped out the bolt area subimages that were not related to the identification of the fastener state on the fastener area image and then used the CNN network for classification. Through this approach, the accuracy rate is improved compared to that with the classification directly in the fastener area. To address the impact of the imbalance problem of the dataset samples on the performance of the detection model, Liu et al. [26] proposed a similarity-based deep network, which obtains a large number of training samples by combining an abnormal sample with multiple normal samples. Liu et al. [27] proposed to use U-Net to generate a large number of defective fastener samples, after which the fasteners were detected using convolutional neural network.

In the last decade, many scholars have conducted research on the detection methods of rail surface defects. These methods mainly solve three problems, namely, the classification of rail surface defects [28, 29], location of rail surface defects [30–33], and pixel-level segmentation of rail surface defects [34–37]. Among them, the pixel-level segmentation of rail surface defects is a key research problem. Nieniewski [34] proposed a detection method based on morphological processing for pixel-level extraction of rail surface defects. The main advantage of this method is the fast detection speed that can reach 50 ms/frame. Yu et al. [35] proposed a three-stage coarse-to-fine model. At the first stage, the background subtraction model is used to filter the images of the defect-free rail surface area; at the second stage, the region extraction model is used to localize the defective area; and at the last stage, a pixel subtraction model is used to detect the defective contours and perform pixel-level extraction. However, this method involves many steps and is sensitive to noise. Niu et al. [36] applied a binocular line-scanning system to the detection of rail surface defects and used global low-rank, nonnegative reconstruction saliency algorithm, and depth outlier detection to combine the two-dimensional saliency map and the three-dimensional defect contour to obtain the final output result. In recent years, there has been a great development of the detection of rail surface defect using deep learning techniques. Faghih-Roohi et al. [38] proposed to use DCNN to classify images of rail surface areas with defects. Shang et al. [39] used traditional object positioning algorithms to localize the rail surface area on the original track image and then used a fine-tuned CNN network to divide the rail surface subimages into two categories: defective and intact. However, the aforementioned two methods did not detect the specific location of the defect. Song et al. [40] used the YOLOv3 network to localize the defect on the rail surface, but this method did not obtain the specific size and shape

information of the defect. Liang et al. [41] used the SegNet network to identify and segment the defects, but the segmentation accuracy of this method needs to be improved. James et al. [42] proposed TrackNet, which integrates U-Net and ResNet for defect semantic segmentation and classification, respectively. This method improves the accuracy of defect recognition, but the accuracy of semantic segmentation needs to be improved.

The aforementioned methods are mainly aimed at detecting a single railway track component. However, the track images collected in the railway line usually contain both rails and fasteners. If both are detected at the same time, the detection efficiency can be greatly improved. To the best of our knowledge, only one article considers the defect detection problem of rail surface and fasteners simultaneously. Wei et al. [43] used the improved YOLOv3 model to realize the simultaneous detection of rail surface defects and fasteners in the railway track line image and obtained high detection accuracy. However, the types of fasteners considered in this article are different from those considered in our study. This method cannot detect the specific location and size of the rail surface defects, and the detection speed is difficult to meet the actual needs of the project. Realizing the pixel size detection of the surface defect area of the rail helps the inspector judge the degree of the rail disease. For this reason, we propose a detection method based on convolutional neural network (CNN) to automatically detect the rail surface defects and the state of the fasteners on the railway line, in Figure 2. First, we utilize the improved YOLOv5 framework to localize the rail and fasteners in the original railway track line image. Then, a defect detection model based on the Mask R-CNN is designed to semantically segment the defects in the rail subimages. In addition, the ResNet network is used to classify the fastener state in the fastener subimages into normal, loosening, and broken.

The contributions of this study are summarized as follows:

- (1) A railway line key component multiobject detection method is proposed based on a series of deep convolutional neural networks, which can achieve the detection of rail surface defects and fastener state.
- (2) An improved YOLOv5s framework is proposed to localize the rail and fastener in the railway track line image at the same time, and the Ghost bottleneck is used to optimize the backbone network of the original YOLOv5s to effectively reduce the number of parameters and the computational cost. This method can be used for both ballast and ballastless track line image detection. Compared with the original YOLOv5s and other advanced object detection models, the detection speed is significantly improved while maintaining high accuracy.
- (3) The two-stage object detection algorithm, Mask R-CNN, is used in the detection of rail surface defects, which effectively improves the recognition and segmentation accuracy.
- (4) A set of state classification criteria for slab fast clip (SFC) type fastener are proposed.

The rest of this article is organized as follows: Section 2 introduces the rail and fastener positioning method based on the improved YOLOv5. Section 3 describes the rail surface defect detection model based on Mask R-CNN algorithm. Section 4 introduces the state classification criteria of SFC-type fasteners and the classification model used in this paper. Section 5 designs comparative experiments with other competitive methods to verify the effectiveness of our method. Finally, conclusions and future work are presented in Section 6.

## 2. Localization of the Rail and Fastener

*2.1. YOLOv5 Framework.* In this study, we use the improved YOLOv5s object detection neural network to localize the rail and fasteners in the original track images collected from the railway site. The network framework is shown in Figure 3.

The You Only Look Once (YOLO) series network is a one-stage object detection algorithm for object localization and recognition in the image. This algorithm extracts image features by CNN and directly calculates the classification score and object localization [44]. Compared with YOLOv3 and YOLOv4, YOLOv5 is optimized for data enhancement, network structure, and loss function. YOLOv5 uses the following data enhancement methods to improve the robustness of the model: mosaic data enhancement, adaptive anchor box calculation, and adaptive image scaling. Both YOLOv5 and YOLOv4 use mosaic data enhancement to improve the detection ability of the model for small objects. Adaptive anchor box calculation can calculate the best anchor box value depending on different training data sets. Adaptive image scaling can improve the speed of object detection by adding a minimum of black borders when scaling the image. In terms of network structure, YOLOv5 adds a Focus component to the Backbone to perform slicing operations on images, retaining more complete image downsampling information for subsequent feature extraction by adding some floating point operations (FLOPs). The Neck Network chose path aggregation network (PANet) [45] to improve the problem of difficult propagation of low-level features of the original feature pyramid networks (FPN) [46] and strengthened the fusion of extracted features. The Head network chose the same as YOLOv3 and YOLOv4 to realize object detection. The loss function of YOLOv5 is mainly composed of three parts, including bounding box loss, classification loss, and confidence loss. The binary cross entropy is used as the loss function of the classification loss and the confidence loss to calculate the category probability and the target confidence score. We use CIoU loss as the loss function of bounding box, which better describes the regression of rectangular boxes [47].

*2.2. Backbone Optimization.* The original YOLOv5 network used cross stage partial (CSP) bottleneck [48] to increase the depth of the network and thus improve the network's ability

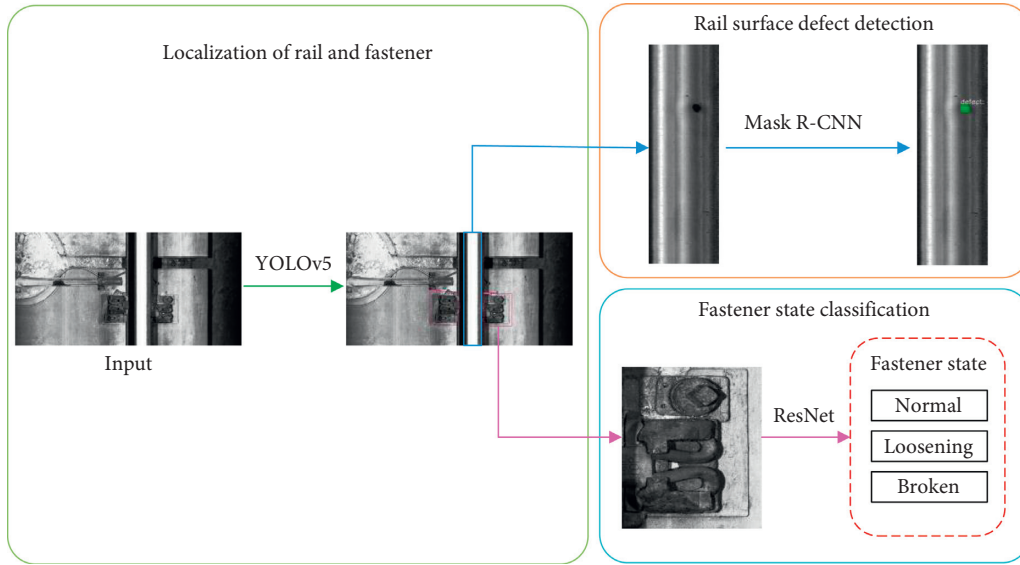


FIGURE 2: Overall framework of rail surface and fastener defect detection method.

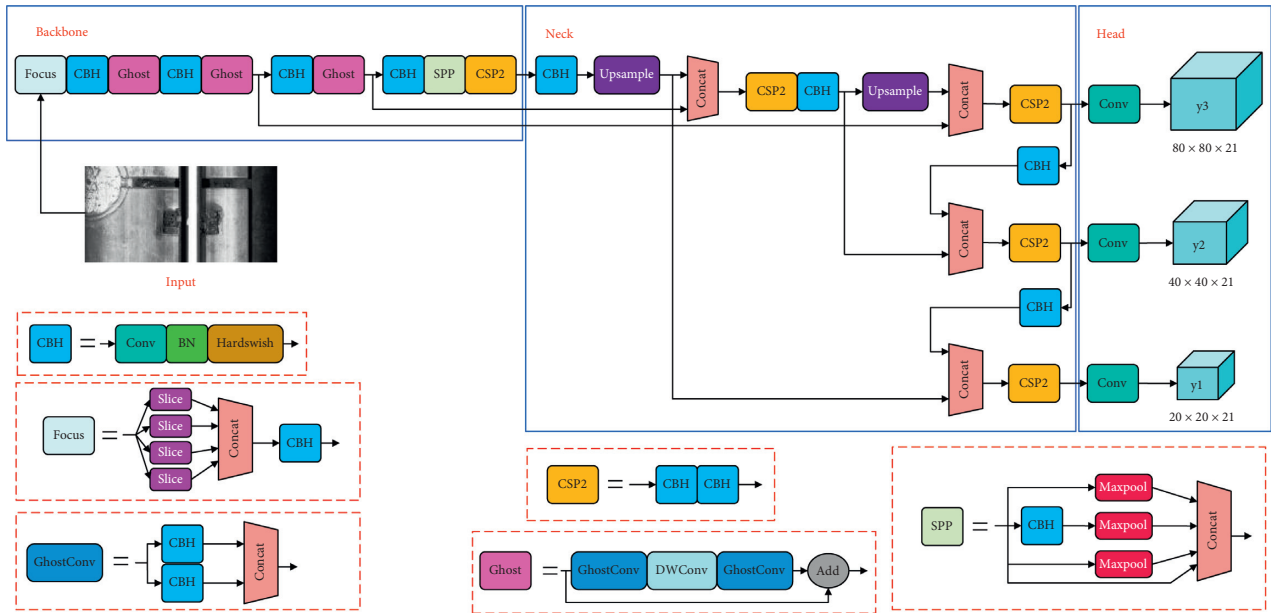


FIGURE 3: Improved YOLOv5 network.

to extract features. However, in the task of rail and fastener localization, we have found that utilizing some modules with lower computational costs to simplify the structure of the model can also achieve satisfactory experimental results. To facilitate our model to be deployed on some low-performance devices with small memory, such as track inspection vehicles or embedded devices, we used a lightweight Ghost bottleneck [49] instead of the CSP bottleneck in the original network to reduce the size of the model and increase the inference speed of the network, as shown in Figure 4. The core idea of the Ghost bottleneck is to use some cheap cost linear operation to generate many feature maps with rich

information. Specifically, first, use a small amount of conventional convolution operations on the feature map to generate intrinsic features, then use some cheap cost linear transformation on the feature map to generate another part of the feature, and finally integrate the two parts together as the final output feature.

The structure of the Ghost bottleneck is shown in Figure 5. Ghost bottleneck consists of two Ghost modules. The network first goes through a Ghost module to increase the number of channels, then a deep-wise convolution to re-integrate the features, and finally a Ghost module to match the number of channels with the shortcut paths. The two are

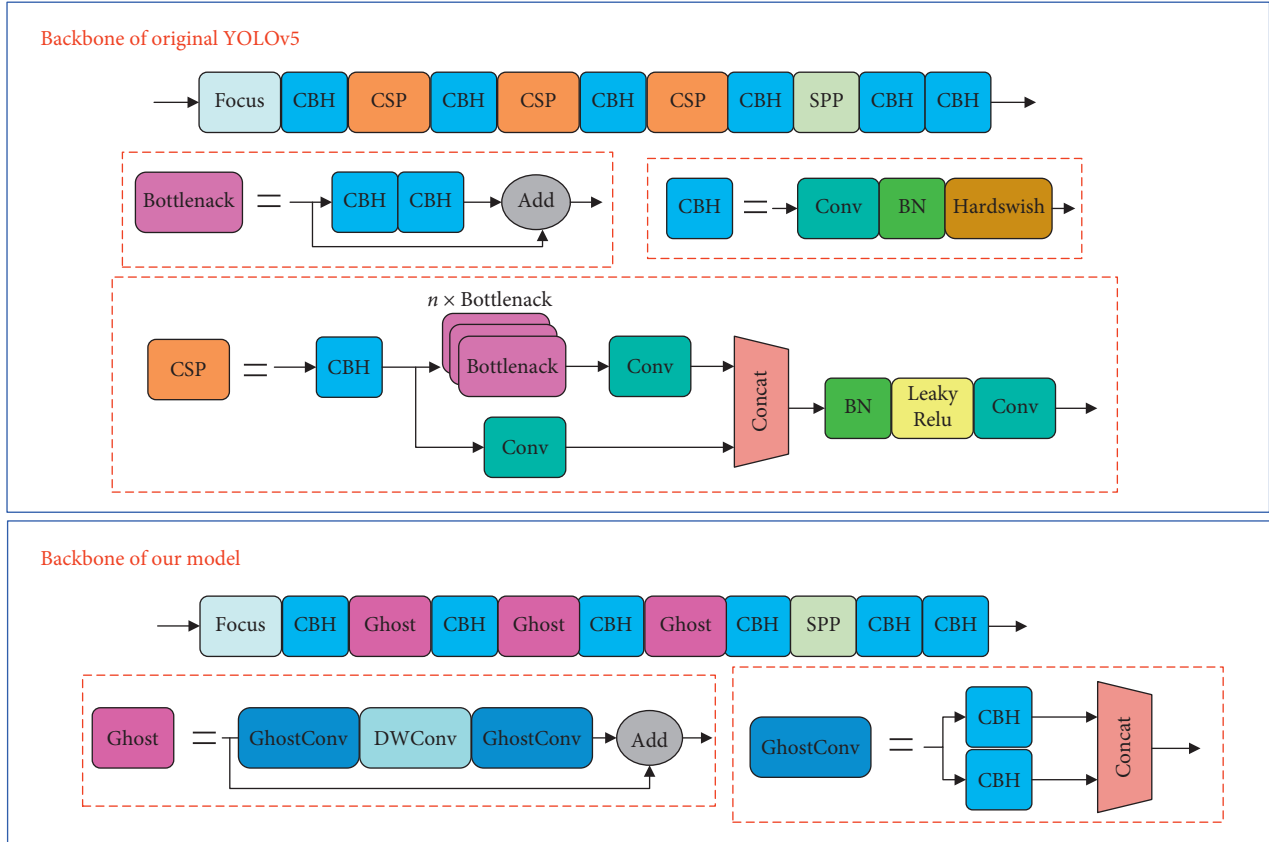


FIGURE 4: Backbone before and after improvements.

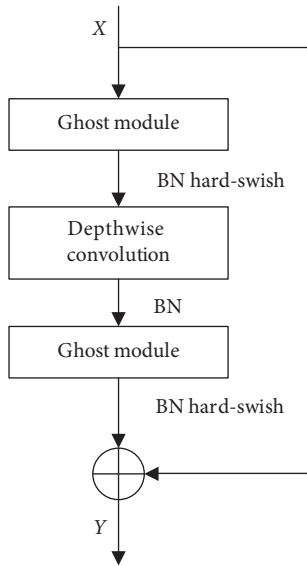


FIGURE 5: Ghost bottleneck.

added together to obtain the final output. Ghost module includes convolution operation and linear transform, and its calculation formula is as follows:

$$\begin{aligned} Y &= X * f, \\ y'_{(i,j)} &= \varphi_{(i,j)}(y_i), \end{aligned} \quad (1)$$

where  $X$  is the input data,  $*$  is the convolution operation, and  $Y = [y_1, y_2, \dots, y_i, \dots, y_m]$  is the output data, which means that the  $m$  channel feature map is obtained after the convolution operation,  $1 \leq i \leq m$ ,  $\varphi_{(i,j)}(y_i)$  in the aforementioned formula is the  $j$ -th linear transformation of the  $i$ -th feature map, and  $Y' = [y'_{(1,1)}, y'_{(1,2)}, \dots, y'_{(i,j)}, \dots, y'_{(m,s)}]$  represents the feature map of  $m \times s$  channels obtained by linear transformation,  $1 \leq j \leq s$ .

The Ghost module can flexibly define the number of convolution kernels and enlarge the number of channels of the input feature map by  $s$  times. Adding a deep-wise convolution between the two Ghost modules can effectively increase the tolerance to changes in the geometric features of the rail and fasteners and reduce the parameter redundancy. Batch normalization (BN) is added after the convolutional layer of each module, and the hard-Swish [50] activation function is added after the convolutional layer of the two Ghost modules to improve the expressive ability of the neural network.

### 3. Rail Surface Defect Detection

In this paper, the Mask R-CNN model is used to localize and segment the defects in the rail surface image.

Mask R-CNN is an improved two-stage object detection network based on the Faster R-CNN framework [51]. On the basis of Faster R-CNN [52], Mask R-CNN optimizes the architecture for bounding box regression and object

classification at the first stage and adds the FCN [53] branch for the second stage of predicting segmentation masks. The network structure is shown in Figure 6.

First, the rail surface image is input to the feature extraction network to generate a multiscale feature map. Second, the obtained feature map is input to the region proposal network (RPN) network to generate a region of interest (RoI). Then, the RoI of different dimensions generated by the RPN network is transformed to features of the same dimension by the RoI Align operation. Finally, the obtained features are, respectively, input to the fully connected layer and FCN for rail surface defect classification, bounding box regression, and segmentation mask prediction.

The rail surface defect detection model designed in this study uses Resnet50 [54] +FPN as the feature extraction network. Using Resnet50 can enable extraction of features at different scales on the rail surface image. However, if only Resnet50 is employed as a feature extraction network, there is the problem of weak detection ability of objects with small objects occurs, which can easily fail to detect small defects on the rail surface. Therefore, adding RPN to integrate the low-level and high-level features of Resnet50 can effectively improve the ability of small defect detection. Four different feature maps from P2 to P5 are used in FPN. Depending on the size of the RoI, different scales of feature maps should be selected. It is ensured that large RoIs are generated from high-semantic feature maps, which is conducive to the detection of large defects, and small RoIs are generated from high-resolution feature maps, which is conducive to the detection of small defects. The specific selection formula is

$$k = k_0 + \log_2 \left( \frac{\sqrt{wh}}{224} \right), \quad (2)$$

where  $k_0 = 4$ ,  $w$  and  $h$  are the width and height of RoI, and  $k$  is the number of layers of the feature map in FPN. To input RoIs of different dimensions to the fully connected layer for classification score calculation and bounding box regression, transforming RoIs of different dimensions to the same dimension is necessary. Mask R-CNN utilizes RoI Align instead of RoI Pooling in Faster R-CNN. RoI Align uses a bilinear interpolation to obtain the values of multiple sampling points and then uses the maximum pooling of the values of multiple sampling points to obtain the final value of the point. This method effectively solves the position mismatch problem caused by two quantization operations in RoI Pooling and can effectively improve the accuracy of detection or segmentation. Finally, the loss function of Mask R-CNN is

$$L = L_{\text{box}} + L_{\text{cls}} + L_{\text{mask}}, \quad (3)$$

where  $L_{\text{box}}$  and  $L_{\text{cls}}$  are the same as in Faster R-CNN [52], representing the bounding box regression loss and object classification loss, and  $L_{\text{mask}}$  is the mask loss. The mask branch in the network uses the Sigmoid function for each pixel on the mask, then feeds it into the cross-entropy loss, and defines the average of all pixel losses as the mask loss.

## 4. Fastener State Classification

*4.1. Judgment Criteria for Fastener State.* The track fasteners used in this experiment are Pandrol fast clip. The fasteners in the track images collected on the railway line are in three states, namely, normal, loosening, and broken, as shown in Figure 7. Currently, no set of criteria is available to classify the normal and loosening states of SFC-type fasteners. Therefore, this study divides the fastener area into the two parts shown in Figure 8 as the criteria for judging the state of the SFC-type fasteners based on the experience of the railway line inspection staff. When the clip is completely within area A, the fastener is fastened and is in a normal state. When the clip appears in area B, the fastener is in a loosening state.

*4.2. Classification Model.* ResNet [54] is a classical deep convolutional network that is widely used in image classification, detection, and segmentation. The core of ResNet is the residual block, as shown in Figure 9. By adding a shortcut branch to the residual block, the problem of gradient disappearance caused by the increase in the number of neural network layers is effectively solved, allowing ResNet to improve the network performance by increasing the number of network layers. The output function of the residual module is as follows:

$$y = F(x, \{w_i\}) + x, \quad (4)$$

where  $x$  and  $y$  are the input and output vectors of the residual block.  $F(x, \{w_i\})$  represents the feature vector obtained after the input vector passes through  $i$  convolutional layers. If the residual block has the same structure as that shown in Figure 9 and contains two weight layers, and then the formula of  $F(x, \{w_i\})$  is as follows:

$$F(x, \{w_i\}) = W_2 f_1 (W_1 x + b_1) + b_2, \quad (5)$$

where  $f_1$  is ReLU function.

Different depth ResNet models can be obtained by setting various channel numbers and residual blocks in the module. In this study, the ResNet101 model is used to detect the state of the Pandrol clip fasteners.

## 5. Experiments and Analysis

*5.1. Data Set.* The images are collected from the Shijiazhuang-Taiyuan high-speed railway line, as shown in Figure 10. The LQ-H3X industrial linear array camera, which is mounted on the special rail inspection vehicle, is used to collect the track images on the line. Through repeated image data acquisition experiments on site, high-resolution grayscale images of 2,572 track fasteners have been collected successfully, including 1,425 images of ballastless tracks and 1,147 images of ballast tracks, whose image resolutions are  $4096 \times 2048$  pixels.

In the localization experiment on rails and fasteners, 2,572 collected original images were selected as the data set. The data set of the rail surface defect detection experiment is composed of two parts: one is derived from the rail subimage obtained from the rail and fastener localization experiment

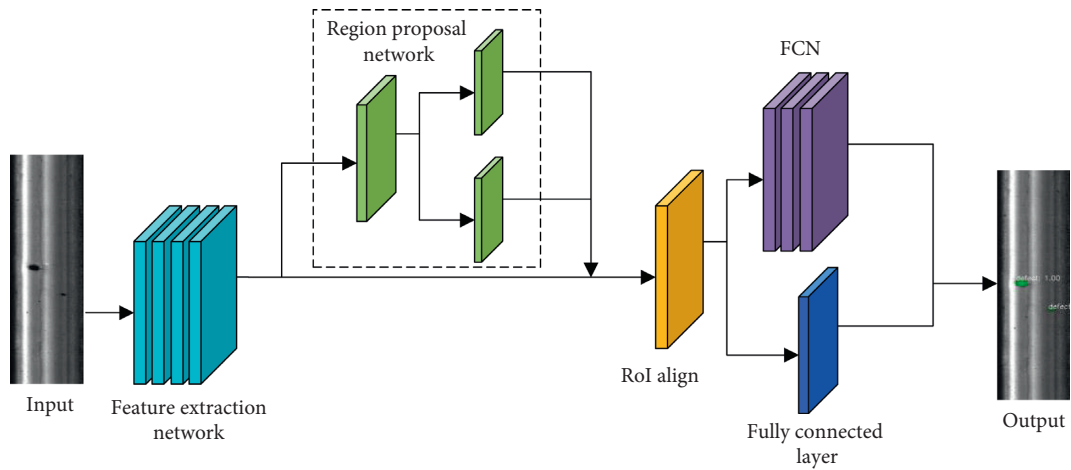


FIGURE 6: Rail surface defect detection model.

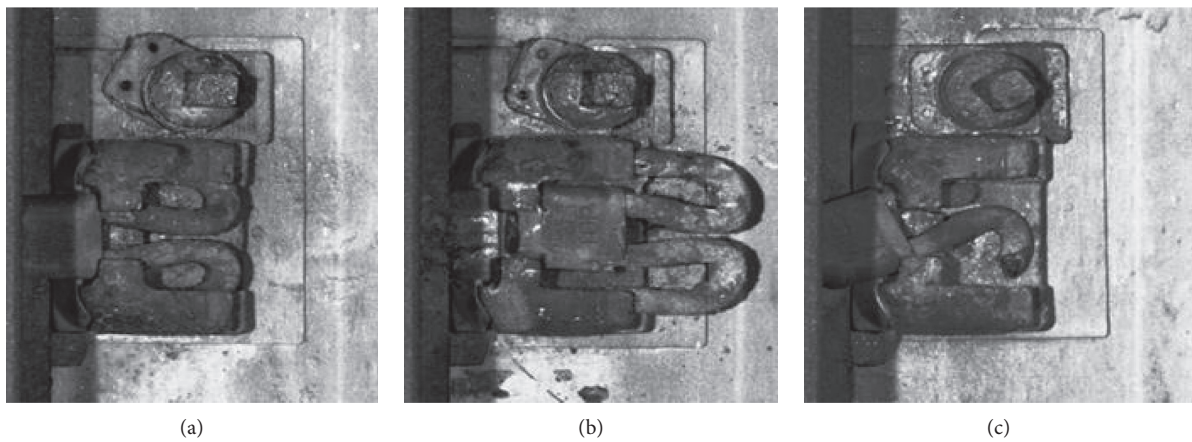


FIGURE 7: Different types of SFC fastener state. (a) Normal. (b) Loosening. (c) Broken.

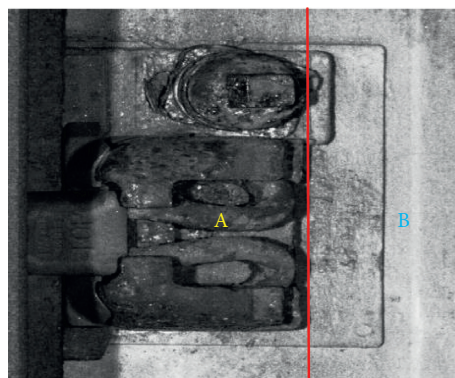


FIGURE 8: Division of the fastener state judgment area.

results, and the other is derived from the public rail surface discrete defect (RSDD) data set [32]. We obtained 526 images, of which rail surface has at least one defect, with width between 140 and 170 pixels and height between 600 and 700 pixels. We selected 825 subimages of fasteners from the experimental results of rail and fastener localization as the data set of fastener state detection, including 705 normal

fasteners, 71 loosening fasteners, and 49 broken fasteners. As the number of loosening fasteners and broken fasteners is relatively small, data augmentation methods such as rotation, Gaussian noise, and salt-and-pepper noise are used to expand the samples of defective fasteners. Then, 705 normal fasteners, 152 loosening fasteners, and 130 broken fasteners were obtained as the data set of this experiment ultimately.

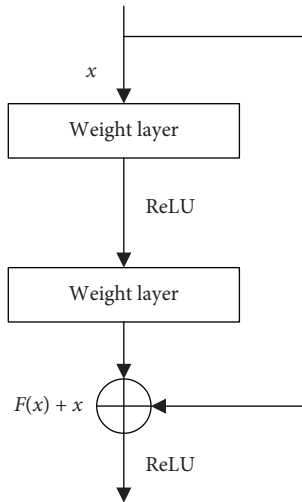


FIGURE 9: Residual block.

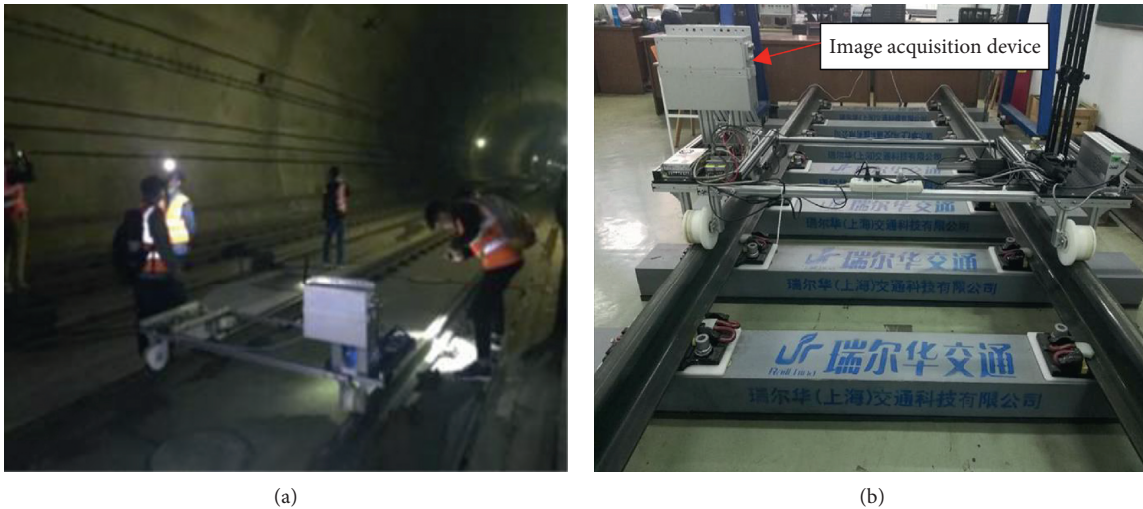


FIGURE 10: Image acquisition. (a) Picture of image acquisition in Shijiazhuang-Taiyuan high-speed railway line. (b) Special rail inspection vehicle.

Of the total number of images, 70% were randomly selected from the data set as the training set, including 494 normal fasteners, 106 loosening fasteners, and 91 broken fasteners. The remaining 30% of the images were used for testing, including 211 normal fasteners, 46 loosening fasteners, and 39 broken fasteners.

**5.2. Experimental Environment.** The experimental environment of this study is based on Windows 10, NVIDIA RTX 2080TI 11 GB GPU, Intel Xeon Silver 4214 2.2 GHz dual CPU and 64 GB RAM. The algorithm based on deep learning was developed using PyTorch framework.

**5.3. Training Process.** The overall training process of our method is shown in Figure 11, which is described as follows:

Step 1: use LabelImg to mark the rail and fastener area in the images of original data set for the training of the improved YOLOv5s to obtain the rail and fastener localization model.

Step 2: use the images of original data set as the input to the rail and localization model to obtain the rail sub-image and the fastener sub-image.

Step 3: combine the rail sub-image obtained in Step 2 with the public RSDD dataset as the rail dataset, and use LabelMe to mark the rail surface defect contours in the



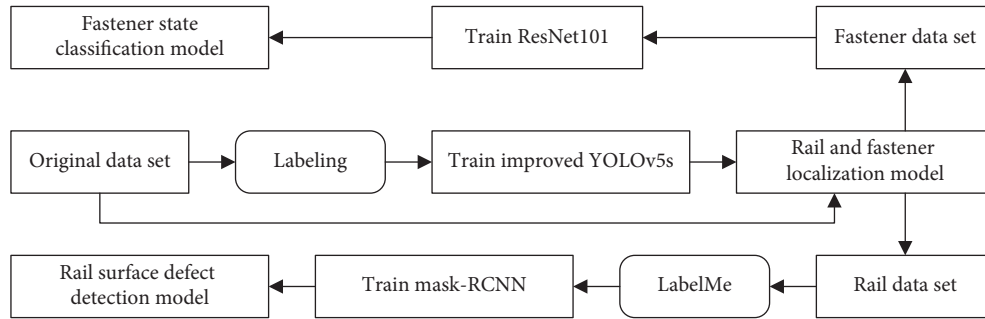


FIGURE 11: Overall training process.

dataset for Mask R-CNN training to obtain rail surface defect detection model.

Step 4: use the fastener subimage obtained in Step 2 as the fastener data set for the training of the ResNet101 model to obtain the fastener state classification model.

In the process of training the rail and fastener localization model, 2572 images were randomly assigned 1543 images as the training set, 2 257 images as the verification set, and the remaining 772 images as the test set. Due to the limitation of the performance of the GPU, the input image is resized to  $1024 \times 512$  pixels during the training. The specific parameter settings of the model are shown in Table 1 and the loss curve of the training process is shown in Figure 12. During the first 20 epochs, training loss converges rapidly, and the decline rate of the train loss value of the model decreases. After 100 epochs, the training efficiency of the model reaches saturation loss value, and the change of loss value is small.

During the training process of the rail surface defect detection model, 526 images were randomly assigned 368 as the training set, 52 as the verification set, and the remaining 106 images as the verification set. In this experiment, the size of the image input to the training model is resized to  $160 \times 650$  pixels. The threshold value of the intersection over union (IoU) in the RPN network was set as 0.6; that is, the IoU between the proposal and ground truth was greater than 0.6, which was retained as the positive sample. Other parameters of the model are shown in Table 2. The loss curve of the training process is shown in Figure 13. The training loss value decreases rapidly before 2500 iterations and tends to be stable after 20000 iterations, finally stabilizing at around 0.06.

#### 5.4. Localization Experiment of the Rail and Fastener

**5.4.1. Analysis of Experimental Results.** Figure 14 shows the visual detection results of two different types of track bed. According to the figure, the proposed model can realize the positioning of rails and fasteners on both ballastless and ballast railway track images.

To further verify the effectiveness of the proposed model, five object detection methods, namely, SSD [55], Faster R-CNN, YOLOv3 [56], Tiny-YOLOv3, and original YOLOv5s, were selected for comparison in this study.

TABLE 1: Parameters of rail and fastener localization model.

Parameters	Value
Input size	$1024 \times 512$
Initial learning rate	0.01
Class	2
Batch size	6
Epochs	120

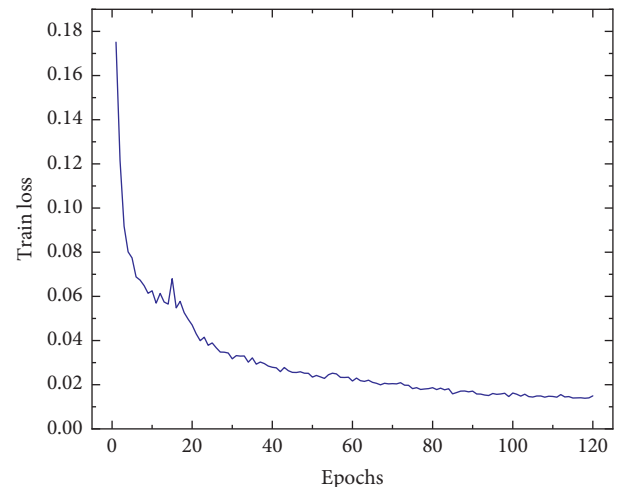


FIGURE 12: Training loss curve of the rail and fastener localization model.

TABLE 2: Parameters of the rail surface defect detection model.

Parameters	Value
Learning rate	0.001
Weight decay	0.0001
Batch size	4
Class	1
Iterations	30000

VGG16 [57] was used for SSD, Resnet50 was used for Faster R-CNN, and Darknet53 [56] was used for Yolov3. Precision (P), recall (R), mean average precision (mAP), and detection speed (FPS) were used as evaluation indexes for object detection:

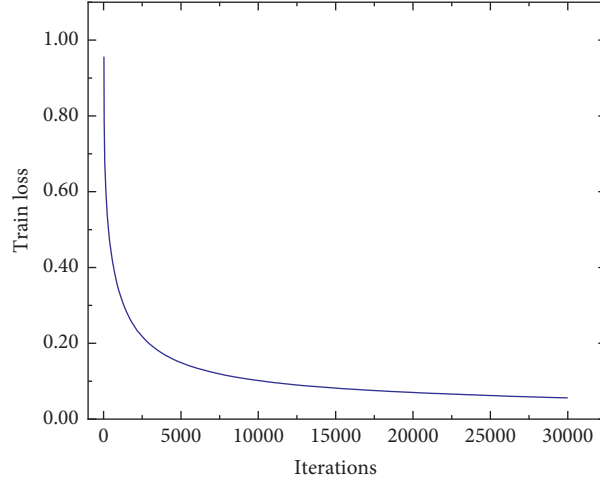


FIGURE 13: Training loss curve of the rail surface defect detection model.

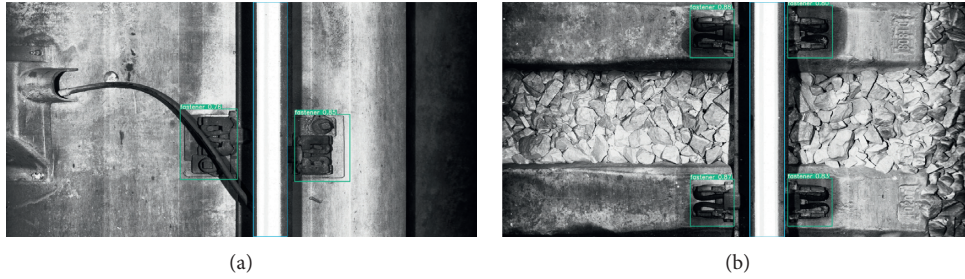


FIGURE 14: Visualization results of rail and fastener localization. (a) Ballastless track image. (b) Ballast track image.

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \\ \text{mAP} &= \frac{\sum_{d=1}^D \text{AP}(d)}{D}, \end{aligned} \quad (6)$$

where TP, FP, and FN represent true positive, false positive, and false negative cases, respectively. AP is the area covered under the P-R curve, and  $D$  represents the number of categories detected.  $D = 2$  was used in this experiment.

The results are shown in Table 3. Detection speed of Tiny-YOLOv3 is obviously faster than that of other methods, but its detection accuracy is only 76.52%. Faster R-CNN has the best detection performance but the lowest detection speed. The detection performance of the proposed model is similar to that of Faster R-CNN and Yolov5s, but the detection speed is significantly faster than that of Faster R-CNN, which is improved by 17.52% compared with the original Yolov5s. At the same time, our model is only 12.6 M in size and can be flexibly deployed on devices with small memory. Therefore, the performance of the object detection model proposed in this study is better than that of the other five methods in our data set.

TABLE 3: Comparison of different object detection methods.

Method	P (%)	R (%)	mAP (%)	Model size (MB)	FPS
SSD	94.72	99.73	98.96	181.2	61.3
Faster R-CNN	97.12	100	99.76	267.8	12.2
YOLOv3	96.81	99.73	99.74	117.2	62.5
Tiny-YOLOv3	76.52	98.04	92.92	16.6	168.4
YOLOv5s	96.41	100	99.71	14.1	83.3
Ours	96.23	100	99.68	12.6	97.9

5.4.2. *Experiment of Rail Surface Defect Detection.* Figure 15 shows the comparison results of the method proposed in this study and other methods for the detection of rail surface defects on different scales, where both PSPNet [58] and Deeplabv3+ [59] chose Resnet50 for the feature extraction network, and the boundary box was ignored for Mask R-CNN. Mask R-CNN has the best detection effect for slight defect because the addition of FPN in the backbone greatly improves the detection performance of small objects. In the three models of moderate and severe defects, the existence of defects can be detected well. However, the prediction of the defect edge by Mask R-CNN is significantly more accurate, and the defect contour can be segmented completely. In addition, the detection effect of Deeplabv3+ was also good, but the segmentation accuracy was inferior to

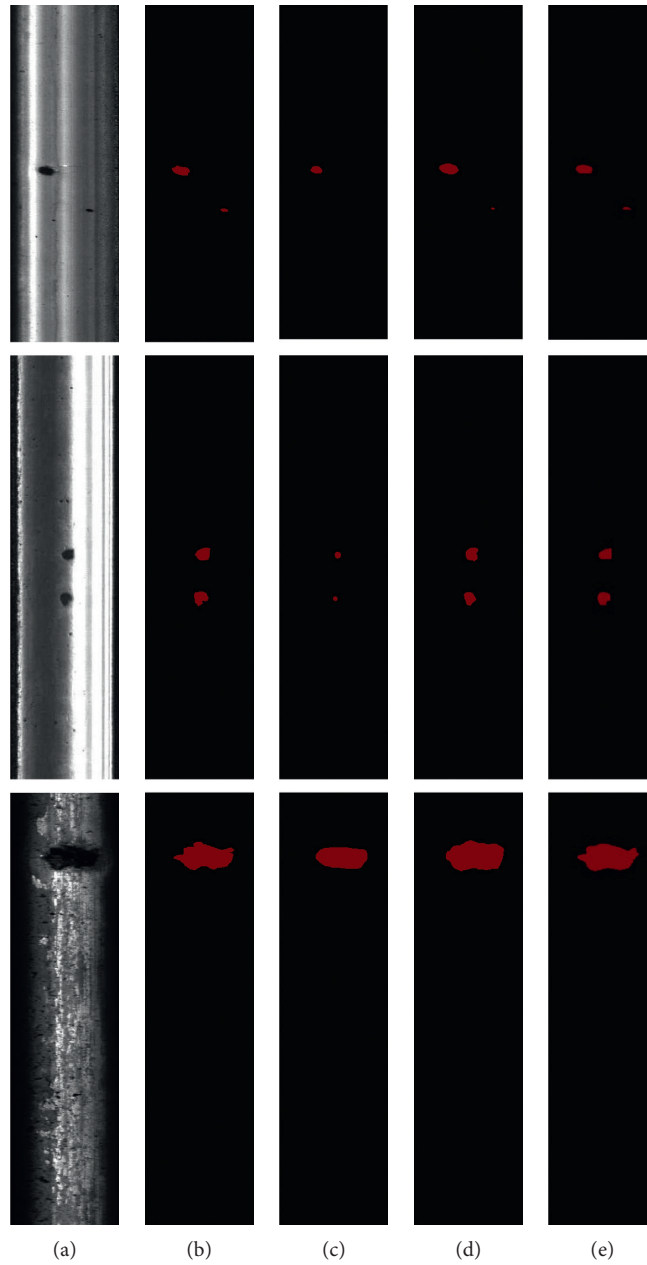


FIGURE 15: Comparison of detection results of rail surface defects with different methods: (a) original image, (b) ground truth, (c) PSPNet, (d) Deeplabv3+, and (e) Mask R-CNN.

that of Mask R-CNN. The segmentation accuracy of PSPNet was the worst, especially for the segmentation with slight and moderate defects. Therefore, the proposed method has high segmentation accuracy and robustness advantages compared with the other two methods.

To obtain quantitative experimental results, pixel accuracy (PA), mean pixel accuracy (MPA), mean intersection over union (MIoU), and frequency weighted intersection over union (FWIoU) were used as evaluation indexes in this experiment. Their specific expressions are as follows:

TABLE 4: Comparison of different segmentation models.

Method	PA (%)	MPA (%)	MIoU (%)	FWIoU (%)
PSPNet	99.41	74.48	73.65	98.84
Deeplabv3+	99.65	92.76	85.67	99.38
Mask R-CNN	99.72	94.37	87.52	99.51

TABLE 5: Classification results of different classification models.

Method	Normal fastener	Loosening fastener	Broken fastener
HOG + SVM	206/211	37/46	31/39
Canny + HOG + SVM	208/211	40/46	32/39
VGG16	211/211	44/46	34/39
ResNet101	211/211	45/46	36/39

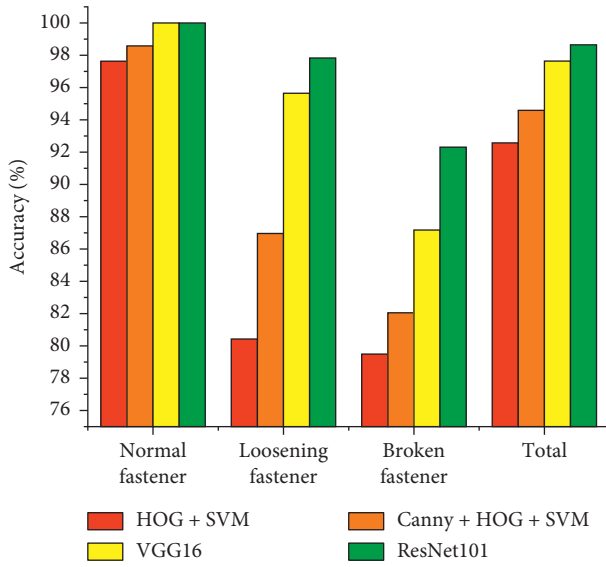


FIGURE 16: Accuracy of different fastener state classification models.

$$PA = \frac{\sum_{i=0}^c P_{ii}}{\sum_{i=0}^c \sum_{j=0}^c P_{ij}}$$

$$MPA = \frac{1}{c+1} \sum_{i=0}^c \frac{P_{ii}}{\sum_{j=0}^c P_{ij}}$$

$$MIoU = \frac{1}{c+1} \sum_{i=0}^c \frac{P_{ij}}{\sum_{j=0}^c P_{ij} + \sum_{j=0}^c (P_{ji} - P_{ii})}$$

$$FWIoU = \frac{1}{\sum_{i=0}^c \sum_{j=0}^c P_{ij}} \sum_{i=0}^c \frac{P_{ii} \sum_{j=0}^c P_{ij}}{\sum_{j=0}^c P_{ij} + \sum_{j=0}^c (P_{ji} - P_{ii})}, \quad (7)$$

where  $p_{ij}$  represents the total number of pixels that belong to the  $i$  class but are predicted to be in  $j$  class, and  $c$  represents the number of categories. Two categories are used in this experiment, namely, defects and background.

Table 4 records the specific quantitative experimental comparison results. As shown in the table, the performance

of PSPNet is significantly lower than that of Deeplabv3+ and Mask R-CNN in MPA and MIoU, with only 74.48% and 73.65%, respectively. The Mask R-CNN model used in this paper achieves the best results in all indicators. One of the main reasons is that Mask R-CNN is a two-stage object detection network and only segments candidate boxes generated in the first stage, which is conducive to the improvement of segmentation accuracy. Therefore, Mask R-CNN performs better in the test set of our dataset.

*5.4.3. Experiment of Fastener State Detection.* We selected some classification models based on deep learning algorithms and some classification models based on traditional shallow learning algorithms to compare our method:

- (1) VGG16: a classic deep learning framework is widely used in object classification and feature extraction networks.
- (2) HOG + SVM: HOG feature extraction is performed on the coupler image, and then the extracted HOG feature is input to SVM for coupler status classification.
- (3) Canny + HOG + SVM: Canny operator [60] first extracts the edge contour features of the coupler image to obtain the edge feature map. The HOG features are extracted from the edge feature map. The SVM algorithm is used for classification finally.

The results of different classification models are shown in Table 5. Figure 16 shows the accuracy comparison results of the various methods. The experiment shows that, compared with the other three methods, Resnet101 achieves the best detection results in our fastener data set. In addition, VGG16 and Resnet101 based on deep learning framework are significantly better than the other two methods in the detection accuracy for all types of coupler. One main reason is that VGG16 and Resnet101 extract advanced semantic features of coupler images by using the convolutional layer, while the other two methods only extract the low-level features of the image by using the artificially designed feature extraction method. Thus, they are better than the traditional machine learning method in terms of classification accuracy and robustness. Canny + HOG + SVM is better than

HOG + SVM because the former method first uses a Canny operator to extract the edge features of the coupler image before extracting HOG features, so that the interference of background and other useless information is reduced on classification, and the classification precision improves to a certain extent. Compared with VGG16, Resnet101 improved the detection accuracy of loosening fasteners by 5.13% and the overall detection accuracy by 1.01%, because Resnet101 uses residual blocks to increase the depth of the CNN. This feature enables Resnet101 to have stronger feature extraction capability.

## 6. Conclusions and Future Work

This study proposed a nondestructive detection method based on deep learning algorithms to implement rail surface and fasteners defect detection. At the object localization stage, part of the structure of the backbone based on the YOLOv5 framework is improved to achieve the localization of the rail and fastener rapidly. Compared with other object detection methods, our method has the highest detection accuracy and fastest detection speed, and the model size is only 12.6 M. At the defect detection stage, Mask R-CNN is used as the defect detection model of the rail surface. Experiments show that our method is more suitable for defect detection of rail surface compared with other advanced semantic segmentation methods. In the state detection of fasteners, a set of criteria for judging the state of SFC-type fasteners is given to judge whether the fasteners are in a normal or loosening state. A comparison between the classification models based on deep learning or traditional machine learning theory can show that ResNet is the most suitable classification method for the fasteners in this data set. In general, the proposed method can effectively detect rail surface defects and fastener states.

In the future, we intend to gain more advanced knowledge of deep learning and optimize the rail surface defects detection model to improve the accuracy of defect segmentation. In our data set, few samples of rail surface defects and fastener defects are available, so we will try to use more data augmentation methods to expand the defect samples and can further improve the robustness of our method.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 51975347, 51907117, and 12004240).

## References

- [1] X. Jin, Y. Wang, H. Zhang et al., "DeepRail: automatic visual detection system for railway surface defect using bayesian CNN and attention network," *Acta Automatica Sinica*, vol. 45, no. 12, pp. 2312–2327, 2019.
- [2] L. Xiao, B. Wu, and Y. Hu, "Surface defect detection using image pyramid," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7181–7188, 2020.
- [3] D. Huang, S. Liao, A. I. Sunny, and S. Yu, "A novel automatic surface scratch defect detection for fluid-conveying tube of Coriolis mass flow-meter based on 2D-direction filter," *Measurement*, vol. 126, pp. 332–341, 2018.
- [4] J. Wang, L. Luo, W. Ye, and S. Zhu, "A defect-detection method of split pins in the catenary fastening devices of high-speed railway based on deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9517–9525, 2020.
- [5] L. Peng, S. Zheng, P. Li, Y. Wang, and Q. Zhong, "A comprehensive detection system for track geometry using fused vision and inertia," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, Article ID 5004615, 2021.
- [6] W. Zhu, G. Fang, X. Meng et al., "Ultrasound SAFT imaging for HSR ballastless track using the multi-layer sound velocity mode," *Insight*, vol. 4, no. 63, pp. 199–208, 2021.
- [7] Y. Xia, F. Xie, and Z. Jiang, "Broken railway fastener detection based on adaboost algorithm," in *Proceedings of the 2010 International Conference on Optoelectronics and Image Processing*, pp. 313–316, Haikou, China, November 2010.
- [8] C. Aytekin, Y. Rezaeitabar, S. Dogru, and I. Ulusoy, "Railway fastener inspection by real-time machine vision," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1101–1107, 2015.
- [9] L. Jiajia, X. Ying, L. Bailin et al., "Research on automatic inspection algorithm for railway fastener defects based on computer vision," *Journal of the China Railway Society*, vol. 38, pp. 73–80, 2016.
- [10] Y. Qiu, X. Chen, and Z. Lv, "Rail fastener positioning based on double template matching," *Complexity*, vol. 2020, Article ID 8316969, 10 pages, 2020.
- [11] Y. Li, C. Otto, N. Haas et al., "Component-based track inspection using machine-vision technology," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, Trento, Italy, April 2011.
- [12] Y. Li, H. Hoang Trinh, N. Haas, C. Otto, and S. Pankanti, "Rail component detection, optimization, and assessment for automatic rail track inspection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 760–770, 2014.
- [13] X. Wei, Z. Yang, Y. Liu et al., "Railway track fastener defect detection based on image processing and deep learning techniques: a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 61–81, 2019.
- [14] H. Feng, Z. Jiang, F. Xie, P. Yang, J. Shi, and L. Chen, "Automatic fastener classification and defect detection in vision-based railway inspection systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 877–888, 2014.
- [15] J. Yang, W. Tao, M. Liu, Y. Zhang, H. Zhang, and H. Zhao, "An efficient direction field-based method for the detection of fasteners on high-speed railways," *Sensors*, vol. 11, no. 8, pp. 7364–7381, 2011.
- [16] P. Dollar and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1558–1570, 2015.

- [17] E. Resendiz, J. Hart, and N. Ahuja, "Automated visual inspection of railroad tracks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 751–760, 2013.
- [18] X. Gibert, V. M. Patel, and R. Chellappa, "Robust fastener detection for autonomous visual railway track inspection," in *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 694–701, IEEE, Waikoloa, HI, USA, January 2015.
- [19] H. Trinh, N. Haas, Y. Li, C. Otto, and S. Pankanti, "Enhanced rail component detection and consolidation for rail track inspection," in *Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision*, pp. 289–295, Breckenridge, CO, USA, January 2012.
- [20] L. Liu, F. Zhou, and Y. He, "Automated status inspection of fastening bolts on freight trains using a machine vision approach," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 230, no. 7, pp. 1629–1641, 2016.
- [21] P. D. Ruvo, A. Distanto, E. Stella, and F. Marino, "A GPU-based vision system for real time detection of fastening elements in railway inspection," in *Proceedings of the 2009 16th IEEE International Conference on Image Processing*, pp. 2309–2312, Cairo, Egypt, February 2010.
- [22] F. Marino, A. Distanto, P. L. Mazzeo, and E. Stella, "A real-time visual inspection system for railway maintenance: automatic hexagonal-headed bolts detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 7, pp. 418–428, 2007.
- [23] L. Li, R. Sun, S. Zhao et al., "Semantic-Segmentation-Based rail fastener state recognition algorithm," *Mathematical Problems in Engineering*, vol. 2021, Article ID 8956164, 15 pages, 2021.
- [24] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 153–164, 2017.
- [25] A. Ma, Z. Lv, X. Chen et al., "Pandrol track fastener defect detection based on local convolutional neural networks," *Proceedings of Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering*, 2020.
- [26] J. Liu, Y. Huang, Q. Zou et al., "Learning visual similarity for inspecting defective railway fasteners," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6844–6857, 2019.
- [27] J. Liu, Y. Teng, X. Ni, and H. Liu, "A fastener inspection method based on defective sample generation and deep convolutional neural network," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 12179–12188, 2021.
- [28] K. Ma, T. F. Y. Vicente, D. Samaras, M. Petrucci, and D. L. Magnus, "Texture classification for rail surface condition evaluation," in *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision*, pp. 1–9, Lake Placid, NY, USA, March 2016.
- [29] S. Hajizadeh, A. Núñez, and D. M. J. Tax, "Semi-supervised rail defect detection from imbalanced image data, IFAC-PapersOnLine, 49, 3," in *Proceedings of the 14th IFAC Symposium on Control in Transportation Systems*, pp. 78–83, Istanbul, Turkey, May 2016.
- [30] Q. Li and S. Ren, "A real-time visual inspection system for discrete surface defects of rail heads," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 8, pp. 2189–2199, 2012.
- [31] Q. Li and S. Ren, "A visual detection system for rail surface defects," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1531–1542, 2012.
- [32] J. Gan, Q. Li, J. Wang, and H. Yu, "A hierarchical extractor-based visual rail surface inspection system," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7935–7944, 2017.
- [33] Z. He, Y. Wang, F. Yin, and J. Liu, "Surface defect detection for high-speed rails using an inverse P-M diffusion model," *Sensor Review*, vol. 36, no. 1, pp. 86–97, 2016.
- [34] M. Nieniewski, "Morphological detection and extraction of rail surface defects," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6870–6879, 2020.
- [35] H. Yu, Q. Li, Y. Tan et al., "A coarse-to-fine model for rail surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 3, pp. 656–666, 2019.
- [36] M. Niu, K. Song, L. Huang et al., "Unsupervised saliency detection of rail surface defects using stereoscopic images," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2271–2281, 2021.
- [37] J. Gan, J. Wang, H. Yu, Q. Li, and Z. Shi, "Online rail surface inspection utilizing spatial consistency and continuity," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2741–2751, 2020.
- [38] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. D. Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *Proceedings of the 2016 International Joint Conference on Neural Networks*, pp. 2584–2589, Vancouver, Canada, July 2016.
- [39] L. Shang, Q. Yang, J. Wang, S. Li, and W. Li, "Detection of rail surface defects based on CNN image recognition and classification," in *Proceedings of the 2018 20th International Conference on Advanced Communication Technology*, pp. 45–51, Chuncheon, Republic of Korea, February 2018.
- [40] Y. Song, H. Zhang, L. Liu, and H. Zhong, "Rail surface defect detection method based on YOLOv3 deep learning networks," in *Proceedings of the 2018 Chinese Automation Congress*, pp. 1563–1568, Xi'an, China, December 2018.
- [41] Z. Liang, H. Zhang, L. Liu, Z. He, and K. Zheng, "Defect detection of rail surface with deep convolutional neural networks," in *Proceedings of the 2018 13th World Congress on Intelligent Control and Automation*, pp. 1317–1322, Changsha, China, July 2018.
- [42] A. James, J. Wang, X. Yang et al., "TrackNet - a deep learning based fault detection for railway track inspection," in *Proceedings of 2018 International Conference on Intelligent Rail Transportation*, pp. 1–5, Singapore, Singapore, December 2018.
- [43] X. Wei, D. Wei, D. Suo, L. Jia, and Y. Li, "Multi-target defect identification for railway track line based on image processing and improved YOLOv3 model," *IEEE Access*, vol. 8, no. 1, pp. 61973–61988, 2020.
- [44] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Computer and Electronics in Agriculture*, vol. 178, Article ID 105742, 2020.
- [45] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Salt Lake City, UT, USA, June 2018.
- [46] T. Lin, P. Dollár, R. Girshick et al., "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

- [47] Z. Zheng, P. Wang, W. Liu et al., “Distance-IoU Loss: faster and better learning for bounding box regression,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, Hilton New York Midtown, New York, NY, USA, February 2020.
- [48] C. Wang, H. M. Liao, Y. Wu et al., “CSPNet: a new backbone that can enhance learning capability of CNN,” in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1571–1580, Seattle, WA, USA, June 2020.
- [49] K. Han, Y. Wang, Q. Tian et al., “GhostNet: more features from cheap operations,” in *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1580–1589, Seattle, WA, USA, June 2020.
- [50] A. Howard, M. Sandler, B. Chen et al., “Searching for MobileNetV3,” in *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 1314–1324, Seoul, Republic of Korea, November 2019.
- [51] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask-RCNN,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pp. 2980–2988, Venice, Italy, October 2017.
- [52] S. Ren, K. He, R. Girshick, J. Sun, and R-CNN “Faster, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [53] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [55] W. Liu, D. Anguelov, D. Erhan et al., “SSD: single shot MultiBox detector, computer vision—ECCV 2016,” in *Proceedings of the 14th European Conference on Computer Vision*, pp. 21–37, Amsterdam, Netherlands, October 2016.
- [56] J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” 2018, <https://arxiv.org/abs/1804.02767>.
- [57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, <https://arxiv.org/abs/1409.1556>.
- [58] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6230–6239, Honolulu, HI, USA, July 2017.
- [59] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” 2018, <https://arxiv.org/abs/1802.02611>.
- [60] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern Analysis and machine intelligence*, vol. 8, no. 8, pp. 679–698, 1986.

## Research Article

# Indoor Acoustic Signals Enhanced Algorithm and Visualization Analysis

Suqing Yan <sup>1,2,3</sup> Xiaonan Luo,<sup>3</sup> Xiyan Sun <sup>2,4</sup> Jianming Xiao <sup>5</sup> and Jingyue Jiang<sup>3</sup>

<sup>1</sup>School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China

<sup>2</sup>Guangxi Key Laboratory of Precision Navigation Technology and Application, Guilin University of Electronic Technology, Guilin 541004, China

<sup>3</sup>Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China

<sup>4</sup>National & Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China

<sup>5</sup>Guilin University, Guilin 541004, China

Correspondence should be addressed to Xiyan Sun; sunxiyan1@163.com and Jianming Xiao; 26953411@qq.com

Received 20 May 2021; Accepted 23 July 2021; Published 31 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Suqing Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A pure acoustic signal can be easy to realize signal analysis and feature extraction. However, the surrounding noises will affect the content of acoustic signals as well as auditory fatigue to the audience. Therefore, it is vital to overcome the problem of noises that affect the acoustic signal. An indoor acoustic signal enhanced method based on image source (IS) method, filtered-x least mean square (FxLMS) algorithm, and the combination of Delaunay triangulation and fuzzy c-means (FCM) clustering algorithm is proposed. In the first stage of the proposed system, the IS method was used to simulate indoor impulse response. Next, the FxLMS algorithm was used to reduce the acoustic signals with noise. Lastly, the quiet areas are optimized and visualized by combining the Delaunay triangulation and FCM clustering algorithm. The experimental analysis results on the proposed system show that better noise reduction can be achieved than the most widely used least mean square algorithm. Visualization was validated with an intuitive understanding of the indoor sound field distribution and the quiet areas.

## 1. Introduction

An acoustic signal is the most widely used signal in real life. However, there is a lot of noise that disturbs the original acoustics signal. Excessive environmental noise harms people's physiological and psychological health [1]. Furthermore, long-term exposure to a high noisy environment will cause serious harm to people's health and affect their daily life [2]. Statistics show that more than 70 percent of the world's urban residents are affected by noise pollution [3]. And it is difficult to communicate with people in noisy environments. Even the phenomenon that you cannot hear or not hear clearly occurs. Therefore, acoustic enhancements have caused growing concern all over the world.

Acoustic enhancement algorithms include commonly spectral subtraction, wiener filtering, and adaptive filtering.

Boll proposed firstly the spectral subtraction algorithm with low computational complexity and easy implementation [4]. However, the music noise is caused for nonlinear processing of inaccurate amplitude estimation, and the speech roughness was produced for the lack of phase information of the pure signal. Then, Berouti et al. proposed nonlinear spectral subtraction [5], Gustafsson et al. proposed adaptive gain average spectral subtraction [6], and SIM et al. proposed minimum mean square error spectral subtraction [7]; these methods are not perfect. Lim and Oppenheim [8, 9] proposed the wiener filtering algorithm of speech enhancement. The premise of the wiener filtering is that the speech can be calculated by the AR model. Then, the noise can be reduced by estimating the AR parameters of pure speech. Compared with spectral subtraction and Wiener filtering methods which require prior knowledge of noise



and pure acoustic signals, adaptive filtering methods can dynamically adjust filter parameters using adaptive algorithms under unknown noise conditions to ensure optimal noise suppression performance. Therefore, noise reduction algorithms based on adaptive filtering have been widely used.

Active noise reduction methods eliminate the noise mixed in the useful signal using an adaptive algorithm, adjusting the parameters adaptively [10, 11]. This method is widely used due to its lower complexity and better controllability. Among the active noise reduction algorithms, the least mean square (LMS) algorithm is classical adaptive algorithms [12, 13]. However, due to the fact that its fixed-step manner slowly reaches the optimal coefficient of the whole system, the convergence speed of the LMS algorithm is relatively slow. As a result, the noise cannot be processed and analyzed in real time. Therefore, the filtered-x least mean square (FxLMS) algorithm can eliminate both high- and low-frequency noises [14, 15]. Furthermore, when the error between the received noise and the expected residual becomes more significant, the step is increased to accelerate its convergence to the wiener solution and vice versa.

There are three indoor acoustic simulation methods: wave acoustic method, statistical acoustic method, and geometric acoustic method [16]. The wave acoustic method focuses on studying the effect of standing wave resonance in the room by wave theory [17]. Craggs proposed the finite element method [18] based on the wave acoustic theory. Kopuz and Lalor proposed the boundary element method [19], and Botteldooren proposed the time-domain finite difference method [20]. The statistical acoustic approach focuses on measuring the energy, ignoring the acoustic wave characteristics [17]. Forssen et al. proposed the statistical energy analysis (SEA) to realize the sound field in the railway [21]. The geometric acoustic method ignores acoustic wave characteristics and uses sound lines to describe the sound propagation path when studying the free sound field's diffusion. Krokstad et al. proposed the ray tracing method (RTM) [22]. Allen and Berkley proposed an image source method based on geometric acoustics [23]. Finally, Vorlander combined the tracking method with the image source method [24] to improve the efficiency and accuracy of the indoor acoustic simulation.

The wave-based method is limited to some specific situations, which are used in a small room with uneven frequency distribution and less resonant frequency in low frequency. In addition, the statistical acoustic method is suitable for high frequency and large-sized space. The geometric acoustic method ignores the acoustic fluctuation and is applicable when the indoor sound propagates to an interface whose size is much larger than the sound wavelength. Among the methods mentioned above, the geometrical method has both high accuracy and more applications. The image source (IS) method is the most typical geometrical acoustic method, and it has been widely used in practical applications [25, 26].

In indoor environments, unreliable prior knowledge between noise and pure acoustic signals, and difficult-to-

estimate noise degrade the performance of acoustic enhancement and pose great challenges for attaining the pure acoustic signals. Aiming at a better performance on acoustic enhancement, we propose a novel indoor acoustic signals enhanced method. The basic idea of this method is to produce adaptively the reverse signal equal to the external noise, then to get pure signal by the addition of the received signal and the reverse signal. Finally, the quiet areas are optimized and visualized by combining the Delaunay triangulation and FCM clustering algorithm. The main contributions in this paper are as follows:

Noise reduction based on the FxLMS algorithm is presented for indoor spatial structure. The comparison between the FxLMS algorithm and the LMS algorithm has been researched for noise inhibition of indoor environments. The results demonstrate that the performance of the noise reduction based on the FxLMS algorithm has dramatically improved.

We propose to adopt the Delaunay triangulation and FCM clustering algorithm to analyze the acoustic signal and visualize noise inhibition in indoor environments. The visualization demonstration of noise inhibition is more conducive to examining the indoor effect and specific distribution of indoor noise reduction.

The remainder of this article is arranged as follows. In Section 2, we discuss noise reduction and the visualization of acoustic field distribution. The proposed method is introduced in Section 3 including the FxLMS algorithm and FCM clustering algorithm. Experimental results are depicted in Section 4. Finally, the conclusions are summarized in Section 5.

## 2. Related Work

*2.1. Noise Reduction.* Active noise reduction is realized with superposition and cancellation of the controlled acoustic wave and original noise. It can effectively suppress low-frequency noise that is difficult to reduce in the passive method.

The FxLMS algorithm is an active noise control method. The secondary channel composed of a loudspeaker and error sensor is used in the FxLMS algorithm [15, 27]. The input reference signal is processed to get the control signal. The weight vector of the FxLMS algorithm is modified by comparing the control signal with the error signal so that it can be adjusted at all the target frequency bands. Erkan completed the headset design of a single channel, which is realized by the FxLMS algorithm [28]. Liu analyzed the performance of a narrow-band active noise control system based on the FxLMS algorithm [29]. Kuo researched the FxLMS algorithm on an embedded platform [30]. Jordan and Elliott constructed a multichannel FxLMS active noise reduction system to suppress the multiline spectrum superimposed noise generated by the yacht engine and proposed a method to determine the convergence coefficient of each channel [31].

**2.2. Noise Inhibition Visualization.** Visualization is an intuitive method to help researchers know acoustic fields. However, visualizing acoustic fields is a complex problem in the acoustic simulation since sound will incur the reflection and absorption during the propagation. Oikawa et al. described the united visualization for acoustic field and the source fluctuation using the 3D laser [32]. Acoustical holography is the most widely used acoustic visualization technology. Wang and Bei applied an optimization method in the design of a microphone array [33]. Koprinkova-Hristova and Alexiev proposed a dynamic visual approach for acoustic camera perception [34] and created a 3D visualization of acoustic wave propagation in time. To visualize acoustic fields, the sound is typically estimated using active noise control (ANC) in the room at a given time in this paper; the sound field distribution during propagation and the quiet areas after noise reduction is visualized.

### 3. Proposed Method

In this section, a novel indoor acoustic signal enhanced method is proposed, aiming to realize a better performance for noise reduction and the conducive visualization of the acoustic signal distribution. The framework of our proposed method includes three stages. In the first stage, a reverberation acoustic signal is simulated by the IS method. Then, the reverse signals equal to the noises are produced by the FxLMS algorithm, and the denoised signal is gained by the addition of the reverberation signals and the reverse signal. After noise reduction, the data can be divided into different subsets. Then, all the quiet points are clustered through the FCM clustering algorithm. We adopt the Delaunay triangulation to subseparate the quiet points set. Lastly, visualization is developed for indoor acoustic signal distribution and the quiet areas in the room.

**3.1. Acoustic Signal Simulation.** In this paper, the source acoustic signal is recorded by the audiorecorder function of MATLAB, its sampling frequency  $f_s = 8000$  Hz, and the format is `audio1 = audiorecorder(8000, 16, 1)`. The IS method [23] is adopted to simulate the impulse response in indoor environments. Therefore, the acoustic simulation results of the received position in the space can be obtained.

In indoor environments, the sound may be reflected in each wall. Therefore, an image sound can be considered at each reflection. The distributions of the sound source and image source and the received position in 3D space are shown in Figure 1.  $S$  and  $R$  denote the sound source position and the received position, respectively.  $S_1, S_2, S_3,$  and  $S_4$  are the image source positions. In Figure 1(b), a solid arrow represents the direct path between the source position  $S$  and the received position  $R$ , and reflected paths between the image source positions and the received position  $R$  are represented by the dotted arrow.

Suppose the virtual room is  $a * b * c$ , the received position  $R = [R_x \ R_y \ R_z]$ , and the source position  $S = [S_x \ S_y \ S_z]$ . Only analyze two boundaries  $y=0$  and  $y=b$  for simplicity without loss of generality. The two image

positions will be  $S_1 = [S_x \ -S_y \ S_z]$  and  $S_2 = [S_x \ 2b - S_y \ S_z]$ , and the distances from  $S_1, S_2$  to  $R$  can be computed. We can also obtain the other image source positions and calculate the distances in the same way; the impulse response of the room is obtained by the image source (IS) method. Therefore, the total acoustic signals of the received position should be gained by the acoustic and all reflected acoustics in the received position.

As a result, the sound will be reflected in each boundary, and image sound can also be propagated and reflected at each border. The number of image sounds will increase exponentially, and the calculation will be much more complex with multiple reflections considered. However, the farther the image sounds are from the received position, the more the attenuation will be. It is crucial to analyze the reflected distribution for simulating indoor sound with more accuracy. Given  $K$  be the number of reflections, let  $k$  be  $[-K: K]$ .

Therefore, the array composed by image acoustic signal path can be expressed as follows:

$$M_k = k + 0.5 * [1 + (-1)^k]. \quad (1)$$

After determining all image positions, the distance from the source position  $S$  to the received position  $R$  is presented as

$$D_k = |(-1)^k * S + M_k * M - R|, \quad (2)$$

where  $M$  denotes room position. Sound signals of the received position are got if multiple acoustic signals arrive at the received position.

The acoustic signal must be convolved to get acoustic fluctuation at the received position. The convolution is represented as

$$G(k) = \sum_j Q(j)\varphi(k - j + 1), \quad (3)$$

where  $Q$  is the source data after normalization and  $\varphi$  is the spatial impact factor vector. Then, the indoor image acoustic signal simulation model can be expressed. The signal  $G$  can be obtained by the convolution of the function  $\varphi$  and the source signal  $Q$ . Hence, the final output  $G$  is the fixed-point acoustic simulation result of the received position under the condition of indoor space based on the image source method.

To simulate the acoustic signal in the indoor environment, we suppose that the parameters as follows: room size is  $5 \times 7 \times 3 \text{ m}^3$ , the boundary of the walls is not rigid, the absorption coefficient is 0.4, acoustic source position is  $S = [0.5 \ 0.5 \ 2.5] \text{ m}$ , and reflection coefficient  $K = [0 \ 5 \ 15]$ .

Figure 2 shows the simulation results with different received positions. The distance  $D_1$  from the source position  $S$  to received position  $R_1 = [3.5 \ 5.0 \ 1.3] \text{ m}$  is  $D_1 = 5.5399 \text{ m}$ , the distance  $D_2$  from the source position  $S$  to received position  $R_2 = [2.0 \ 3.0 \ 1.3] \text{ m}$  is  $D_2 = 3.1528 \text{ m}$ , and the distance  $D_3$  from the source position  $S$  to received position  $R_3 = [0.5 \ 5.0 \ 1.3] \text{ m}$  is  $D_3 = 4.6573 \text{ m}$ . Then, the signal

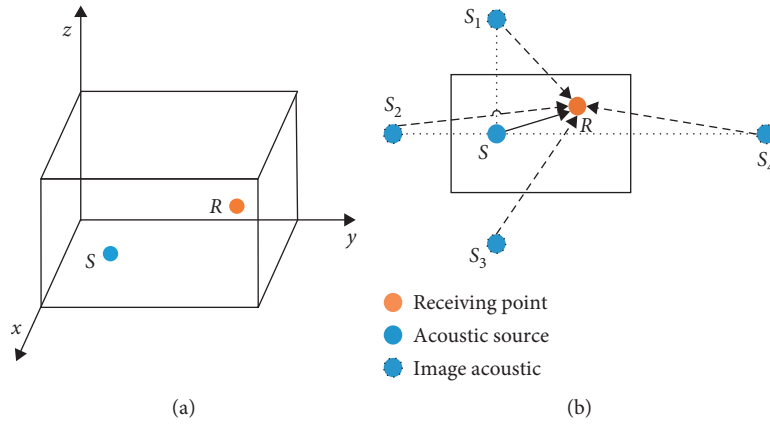


FIGURE 1: The distribution of sound source and image source and received position in 3D space: (a) space structure of sound source point and received position and (b) the distribution of direct path and reflected path.

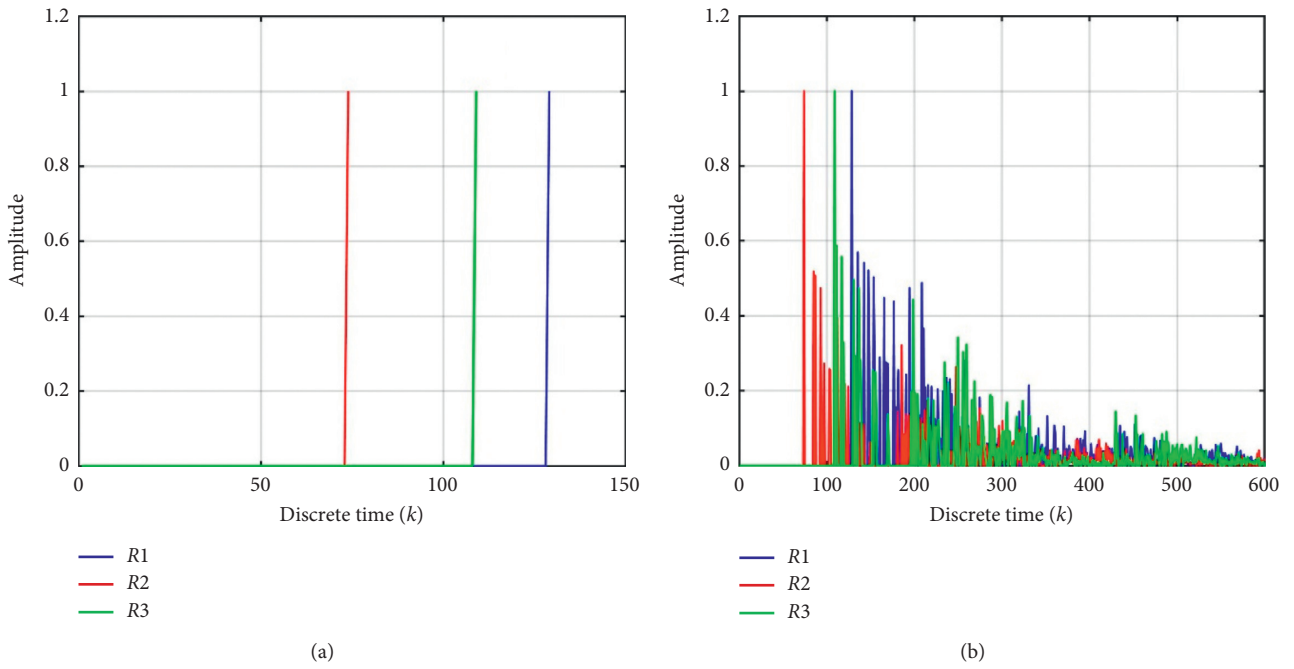


FIGURE 2: Continued.

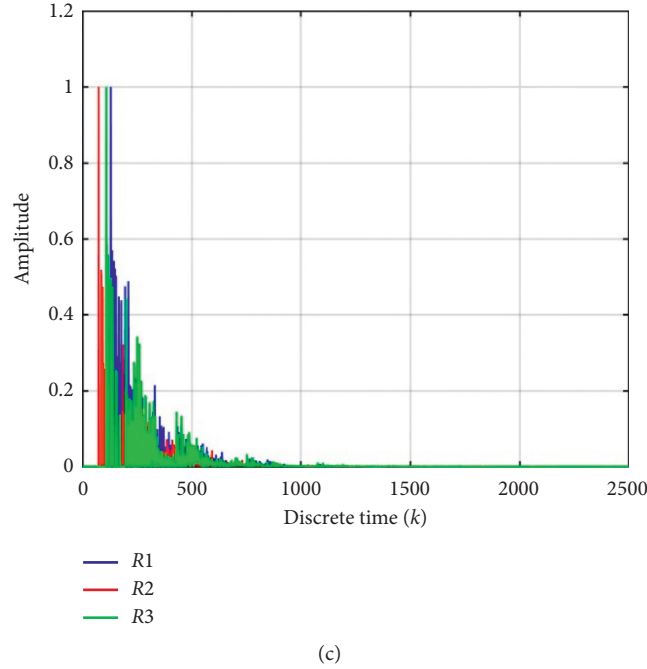


FIGURE 2: Impulse response on the different received positions R1, R2, R3: (a)  $K=0$ , (b)  $K=5$ , and (c)  $K=15$ .

reaches the received positions R1, R2, R3 at [129, 74, 109] as shown in Figure 2(a) at  $K=0$ .

**3.2. FxLMS Algorithm.** FxLMS algorithm [14] structure is shown in Figure 3. In Figure 3,  $P(z)$ ,  $S(z)$ , and  $\hat{S}(z)$  are the transfer function of the primary path, the secondary path, and the secondary path model, respectively. The desired signal  $d(n)$  is the output signal of the primary path. The coefficient of the secondary path is controlled by the residual noise or error signal  $e(n)$  that minimizes the noise.

If the filter  $W(z)$  has L-order transverse structure, therefore input signal  $X(n)$  of the filter  $w(n)$  can be described as

$$X(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T. \quad (4)$$

The residual noise or error signal  $e(n)$  is given by

$$e(n) = d(n) - s(n) * [w(n)^T X(n)], \quad (5)$$

where  $*$  is the convolution sum.

Assuming that  $M$  is the length of the secondary path, then  $E[e^2(n)]$  at the  $n$ th time is expressed by

$$E[e^2(n)] = E \left[ d(n) - \sum_{i=1}^{M-1} s_i(n) \sum_{j=1}^{N-1} w_j(n-i)(n-i-j) \right]^2. \quad (6)$$

We get the gradient of mean square error as follows:

$$\frac{\partial E[e^2(n)]}{\partial w(n)} = 2e(n) \sum_{i=1}^{M-1} s_i(n) \frac{\partial x(n-i)}{\partial w(n)}, \quad (7)$$

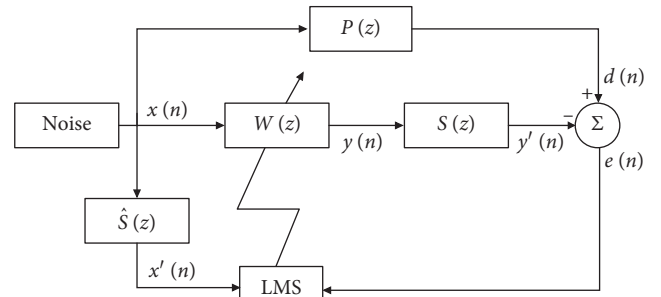


FIGURE 3: FxLMS algorithm block diagram.

If the update step of weight coefficient is small enough, then

$$\frac{\partial J(n)}{\partial W(n)} = 2e(n)x'(n). \quad (8)$$

The gradient descent algorithm of adaptive weighting coefficient is used in the ANC, so the weighting vector can be gained:

$$w(n+1) = w(n) + \mu e(n)X'(n), \quad (9)$$

where  $\mu$  is the convergence factor. It affects convergence speed and stability in the FxLMS algorithm. To ensure stability, the convergence factor must be less than the maximum eigenvalue of the autocorrelation function.

The coefficient of the secondary path is determined according to the error signal during the convergence procedure. A trial-and-error process is used to make sure the factor emerges stable response, and it is slowly decreased until it emerges durable response.

Initially, ANC was used for a single channel, and later, it was extended to the multichannel. In comparison with single-channel noise suppression, the multichannel noise suppression has better performance to gain large quiet regions. Therefore, multichannel noise suppression based on the FxLMS algorithm is designed in this paper.

In the FxLMS algorithm, the results of noise reduction under different parameters are obtained so as to further judge the best noise reduction performance. The antinoise signal is calculated as

$$G_m(k) = \sum_j Y'_m(j) \varphi(k-j+1), \quad (10)$$

where  $Y'_m$  is the control signal of the secondary path.

The signal received at the error microphone is

$$e_m(k) = d_K - G_m(k). \quad (11)$$

The implementation of the ANC is defined as follows:

Multichannel ANC includes one reference microphone, two control loudspeakers, and one error microphone

Choose one acoustic sound position and three control positions

Figure 4 shows the error waveform in different received positions after and before ANC. At the same time, it shows the error waveform in different influence factors. The signal at the received position is consistent with the source signal at different positions. Meanwhile, variations of the signal at the received position are almost compatible with the source signal, except for some differences.

**3.3. FCM Clustering Algorithm.** FCM clustering is a flexible algorithm [35]. By calculating the membership matrix of the sample, the FCM clustering algorithm divides the objects into same-sized clusters with the greatest similarity and the different clusters with minor similarity. Although, in actual most cases, the dataset cannot be divided into distinctly separate clusters, assigning an object to a particular cluster can be rigid and can be error-prone. Therefore, it is better to use fuzzy c-means with natural, nonprobability characteristics in the FCM clustering algorithm.

Supposing the data are divided into  $C$  subsets,  $C$  centers of the subset are gained. Then,  $u_{ij}$  is the degree of membership that data  $i$  belongs to subset  $j$ . FCM clustering algorithm aims to find minimum value as following function [36, 37]:

$$J(U, c_1, \dots, c_C) = \sum_{j=1}^C J_j = \sum_{j=1}^C \sum_{i=1}^M u_{ij} \|x_i - v_j\|^2, \quad (12)$$

with the constraints:

$$\begin{aligned} \sum_{j=1}^C u_{ij} &= 1, \quad \forall i; 0 \leq u_{ij} \leq 1, \forall j, \\ \sum_{i=1}^M u_{ij} &> 0, \quad \forall j, \end{aligned} \quad (13)$$

where  $\{c_1, \dots, c_C\}$  is the set of clustering centers,  $\|\cdot\|$  expresses the Euclidean distance, and  $M$  is the data length. Therefore, the equation can be solved by

$$\begin{aligned} u_{ij} &= \frac{1}{\sum_{l=1}^C (x_i - v_j / x_i - v_l)^{(2/m-1)}}, \quad j = 1, 2, \dots, C; i = 1, 2, \dots, n, \\ v_{ij} &= \frac{\sum_{i=1}^M u_{ij}^m x_i}{\sum_{i=1}^M u_{ij}^m}, \quad j = 1, 2, \dots, C, \end{aligned} \quad (14)$$

where  $m$  is the weighting exponent.

## 4. Experimental Results

To validate the proposed method, we conducted an experiment to compare it with a baseline based on the LMS algorithm. Visualizations of sound field distributions are also presented to help to understand sound propagation. In addition, the FCM clustering algorithm is adopted to optimize the quiet points after indoor noise suppression.

**4.1. Noise Inhibition.** The two-way speaker noise control based on the FxLMS algorithm includes one noise source, one reference microphone, two antinoise loudspeakers, and one error microphone.

The ANC transfer function  $P(z)$  is

$$\begin{aligned} P(z) &= 0.01 + 0.25z^{-1} + 0.5z^{-2} + z^{-3} \\ &\quad + 0.5z^{-4} + 0.25z^{-5} + 0.01z^{-6}. \end{aligned} \quad (15)$$

The secondary-path transfer function is defined as

$$\begin{aligned} S_1(z) &= 0.05 - 0.01z^{-1} + 0.95z^{-2} + z^{-3} + 0.9z^{-4}, \\ S_2(z) &= 1 + 0.44z^{-1} - 0.95z^{-2} + 0.01z^{-3} + 0.9z^{-4}. \end{aligned} \quad (16)$$

The  $5 \times 7 \times 3 \text{ m}^3$  room is the border of the indoor sound field. Its walls are not rigid in which absorption coefficient is 0.4. Figures 5(b)–5(c) show the noise inhibition results with LMS and FxLMS algorithms in the time domain, respectively. As we can see, both methods can reduce noise. In the beginning, the noise inhibition effect based on the FxLMS algorithm does not meet expectations. It is more significant with time increasing. The noise suppression based on the FxLMS algorithm is better than the system based on the LMS algorithm in the time domain. Figure 5(d) shows the spectrum of original noise in the frequency domain. Figure 5(e) describes the residual noise spectrum of the system based on the LMS algorithm. Figure 5(f) describes the residual noise spectrum based on the FxLMS algorithm. The vertical axis denotes the noise amplitude after suppression in dB. Figure 5 shows that the noise in the whole frequency band has been well suppressed. The system based on the FxLMS algorithm has a perfect suppression effect than based on the LMS algorithm.

Figure 6 depicts the experiment result of an  $8 \times 10 \times 4 \text{ m}^3$  medium-sized room, of which the impact factor  $K = 10$  and the source position is [7.9, 9.9, 3.9]. The experiment result shows that the acoustic inhibition of the medium-sized room

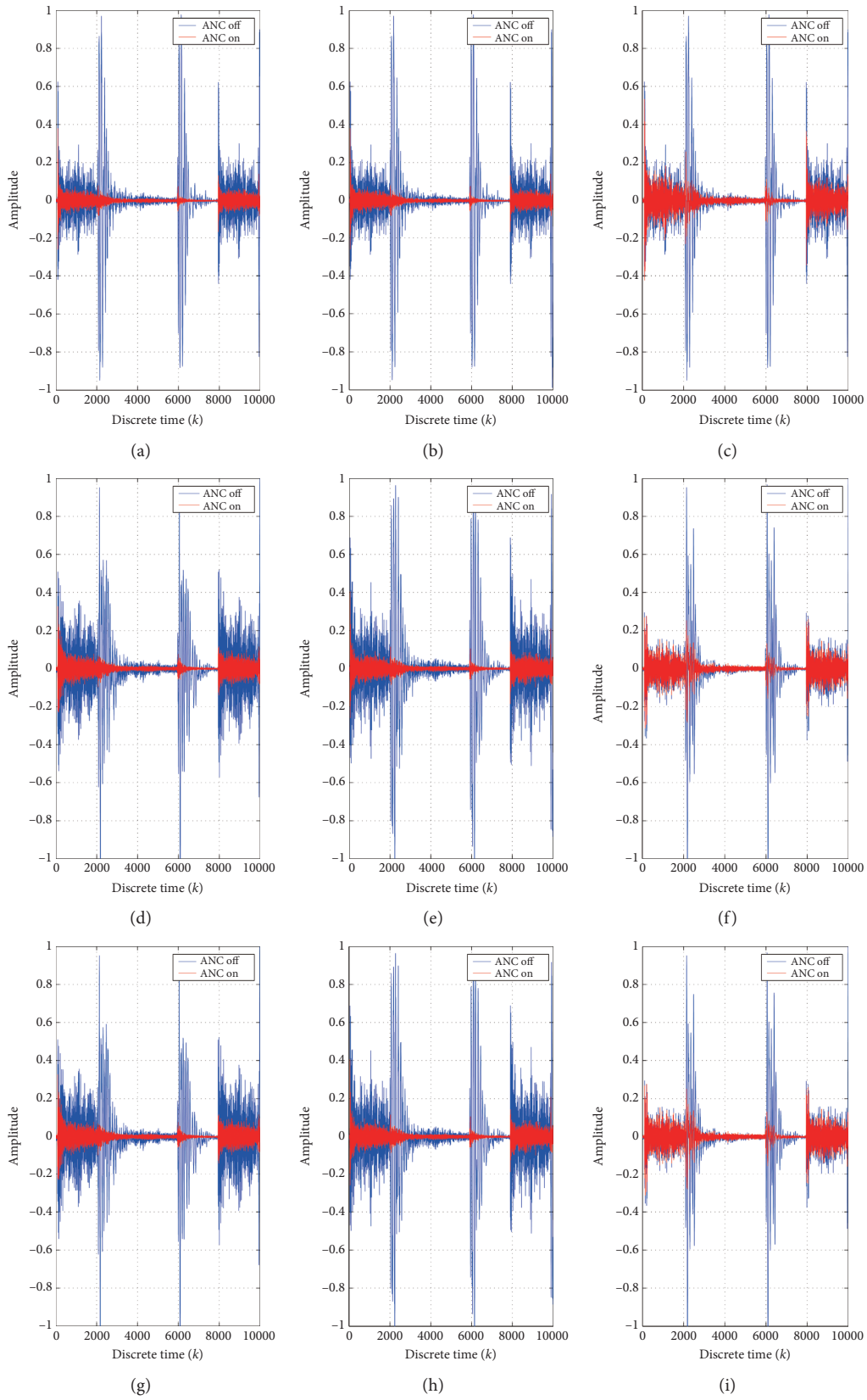


FIGURE 4: The noise reduction waveform of ANC at different received positions with  $K=0, 5$ , and  $15$  in different rows, respectively. (a, d, g) The received position R1. (b, e, h) The received position R2. (c, f, i) The received position R3.

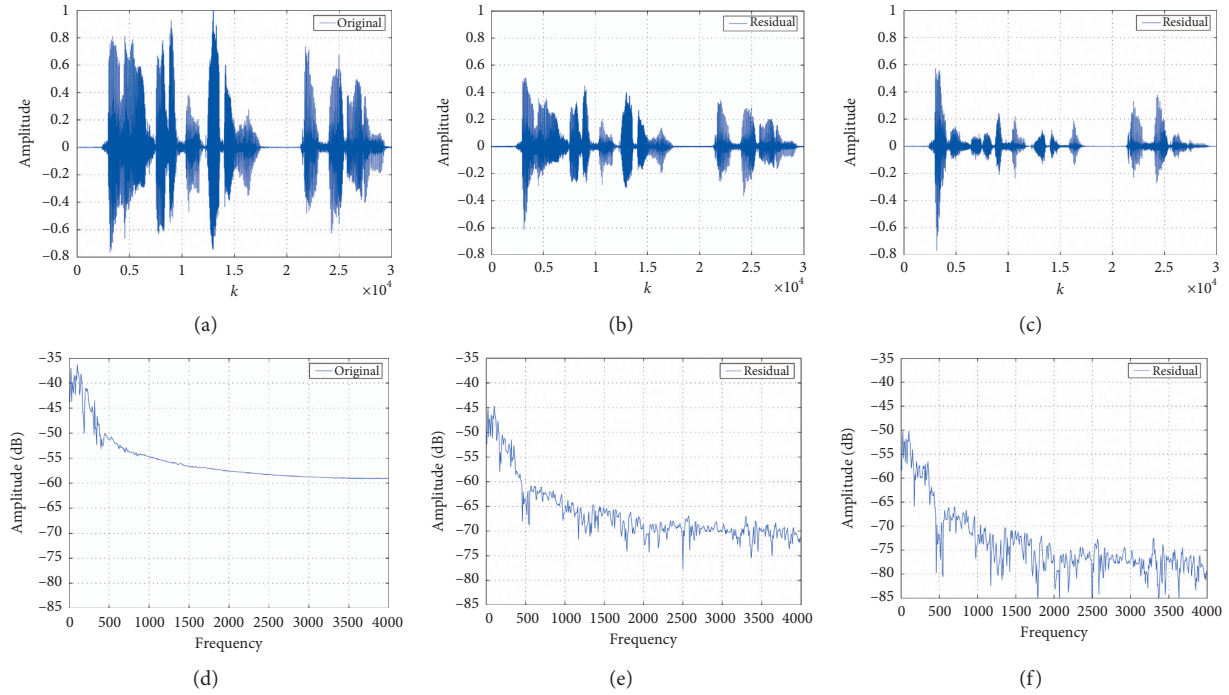


FIGURE 5: Acoustic inhibition comparison of the LMS and FxLMS algorithm. (a) Original noise in the time domain. (b) Residual noise in the time domain using the LMS algorithm. (c) Residual noise in the time domain using the FxLMS algorithm. (d) Original noise in the frequency domain. (e) Residual noise in the frequency domain using the LMS algorithm. (f) Residual noise in the frequency domain using the FxLMS algorithm.

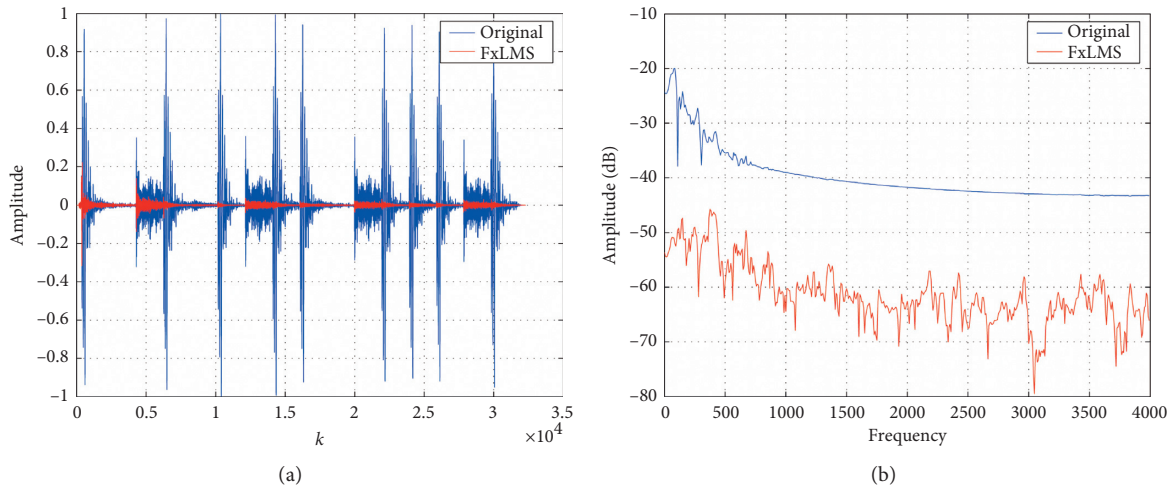


FIGURE 6: Acoustic inhibition comparison based on FxLMS algorithm. (a) Original noise and residual noise in the time domain. (b) Original noise and residual noise in the frequency domain.

can reach 30 dB. Besides, the correlation between signal and interference is weak, and the reflection signal and refraction signal are not obvious for the large-sized room. The noise inhibition for the small-sized room is much more challenging, so the  $5 \times 7 \times 3 \text{ m}^3$  room is adopted in this paper.

**4.2. Distribution of Sound Field.** According to the sound field of the room, the acoustic vibration of each position at different times and spaces can be obtained. However, in

the visualization stage, a large room will lead to difficulties for sound field simulation. Therefore, the room is divided into small units with 10 cm. Image sounds are used to simulate sound information at all the received positions.

In the experiment, the positions of the noise source  $r_{\text{src}}$  and the reference microphone  $r_{\text{mic}}$  are identical; both are [2.5 0.5 2] m. It is considered that the measuring height of the building is between 1.2 m and 1.5 m from the ground in the acoustic environment quality standard. Therefore 1.3 m

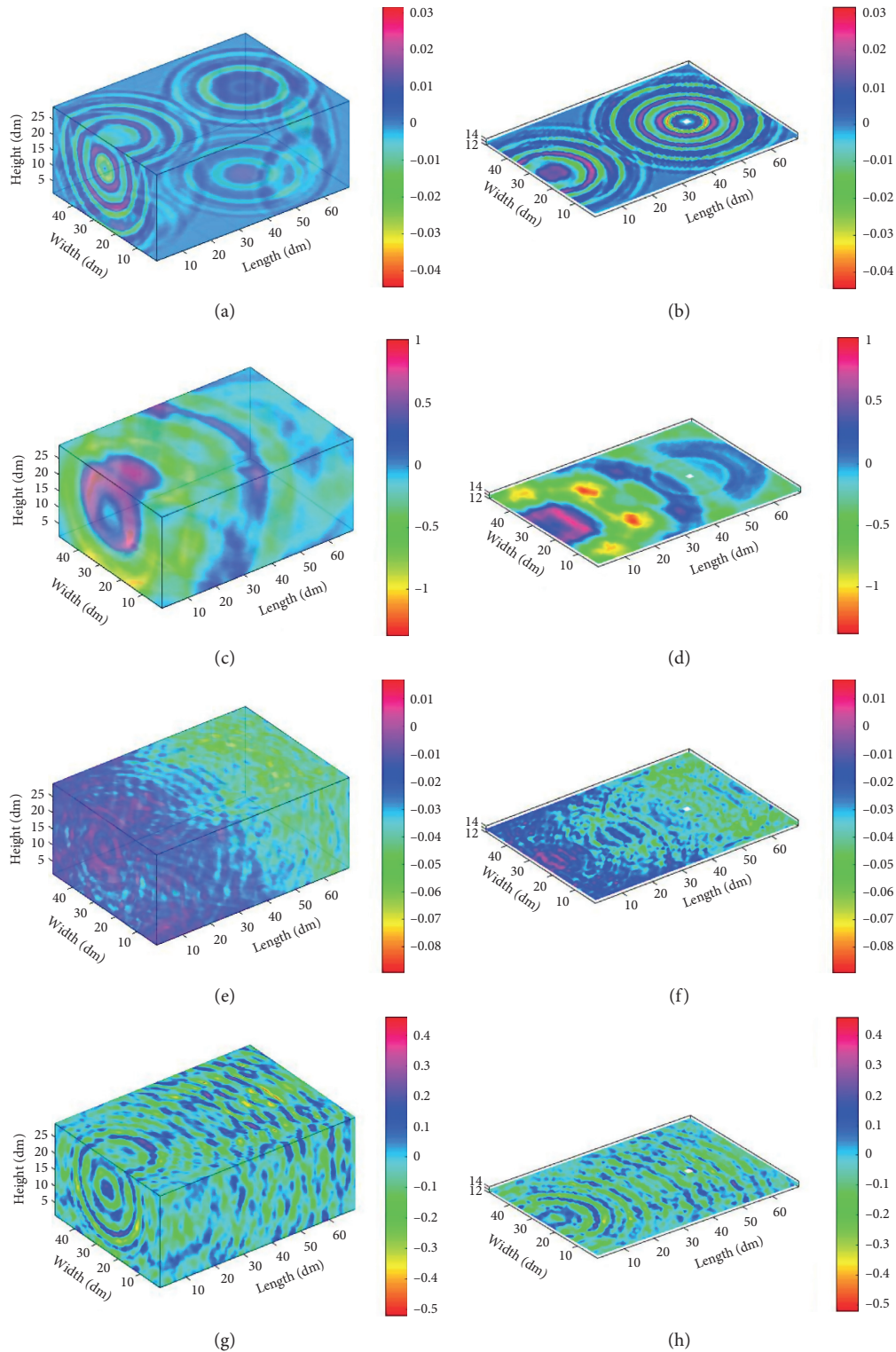


FIGURE 7: Indoor acoustic distributions in 3D space with different times: (a)  $t=0.03$  s, (c)  $t=0.07$  s, (e)  $t=0.4$  s, and (g)  $t=0.6$  s. Acoustic distributions at the height  $h=1.3$  m with different times of (b) 0.03 s, (d) 0.07 s, (f) 0.4 s, and (h) 0.6 s.

is selected as the height in this paper, and the received position  $r_{mic}$  is  $[2.5 \ 5 \ 1.3]$  m.

Considering the areas of human movement, we detect the areas between 1.0 m and 2.1 m in a vertical orientation. The amplitude ranges of  $[-0.001, 0.001]$  are defined as the

quiet points. The acoustic distributions at times  $t=0.03$  s, 0.07 s, 0.4 s, and 0.6 s were examined to observe the sound propagation more intuitively. Figure 7 shows the experiment results. Figures 7(b), 7(d), 7(f), and 7(h) show sound propagation at different times when the height  $h$  is 1.3 m.



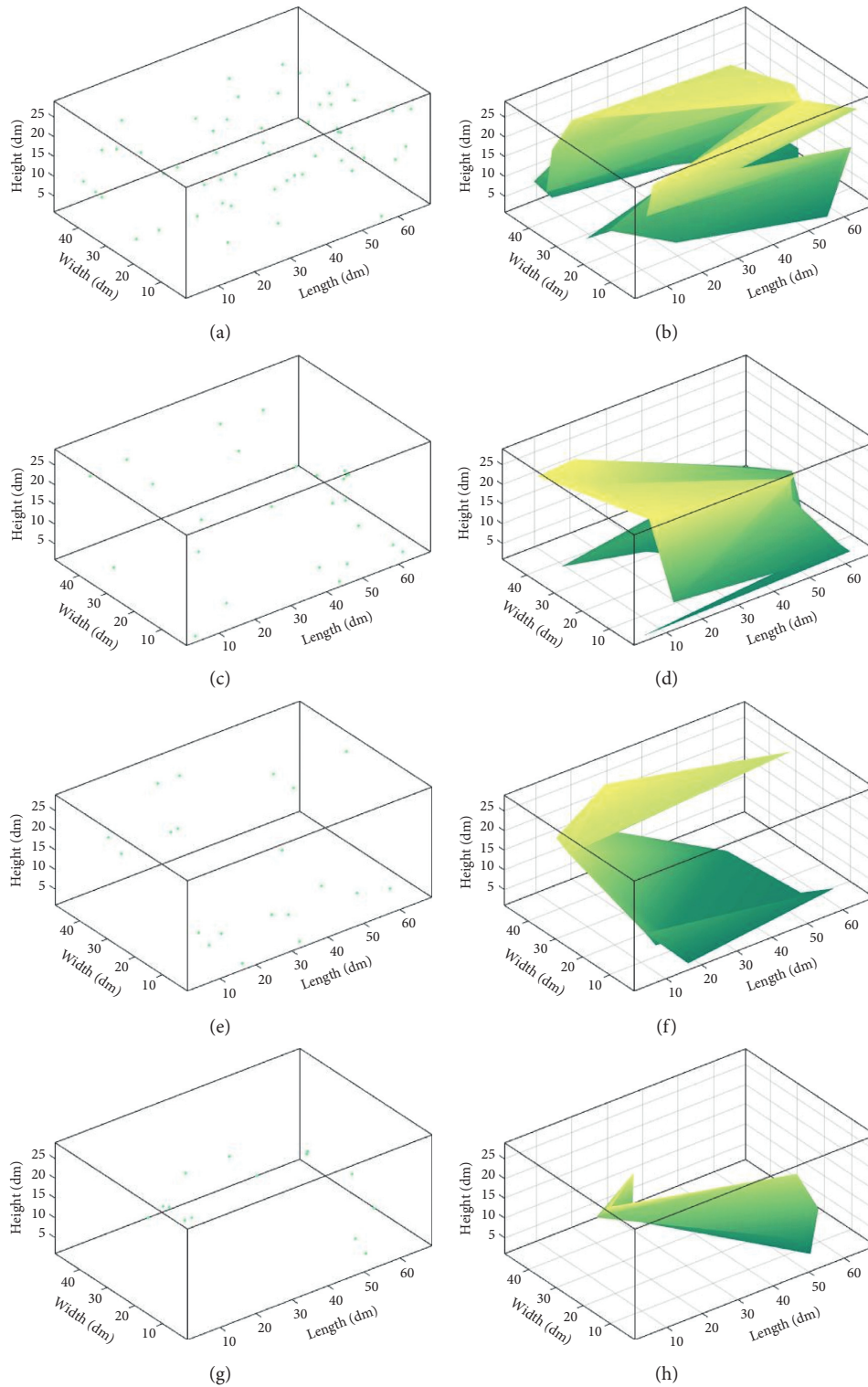


FIGURE 8: The distributions of quiet points at (a)  $t = 0.03$  s, (c)  $t = 0.07$  s, (e)  $t = 0.4$  s, and (g)  $t = 0.6$  s and their corresponding results after subseparation are shown in (b), (d), (f), and (h).

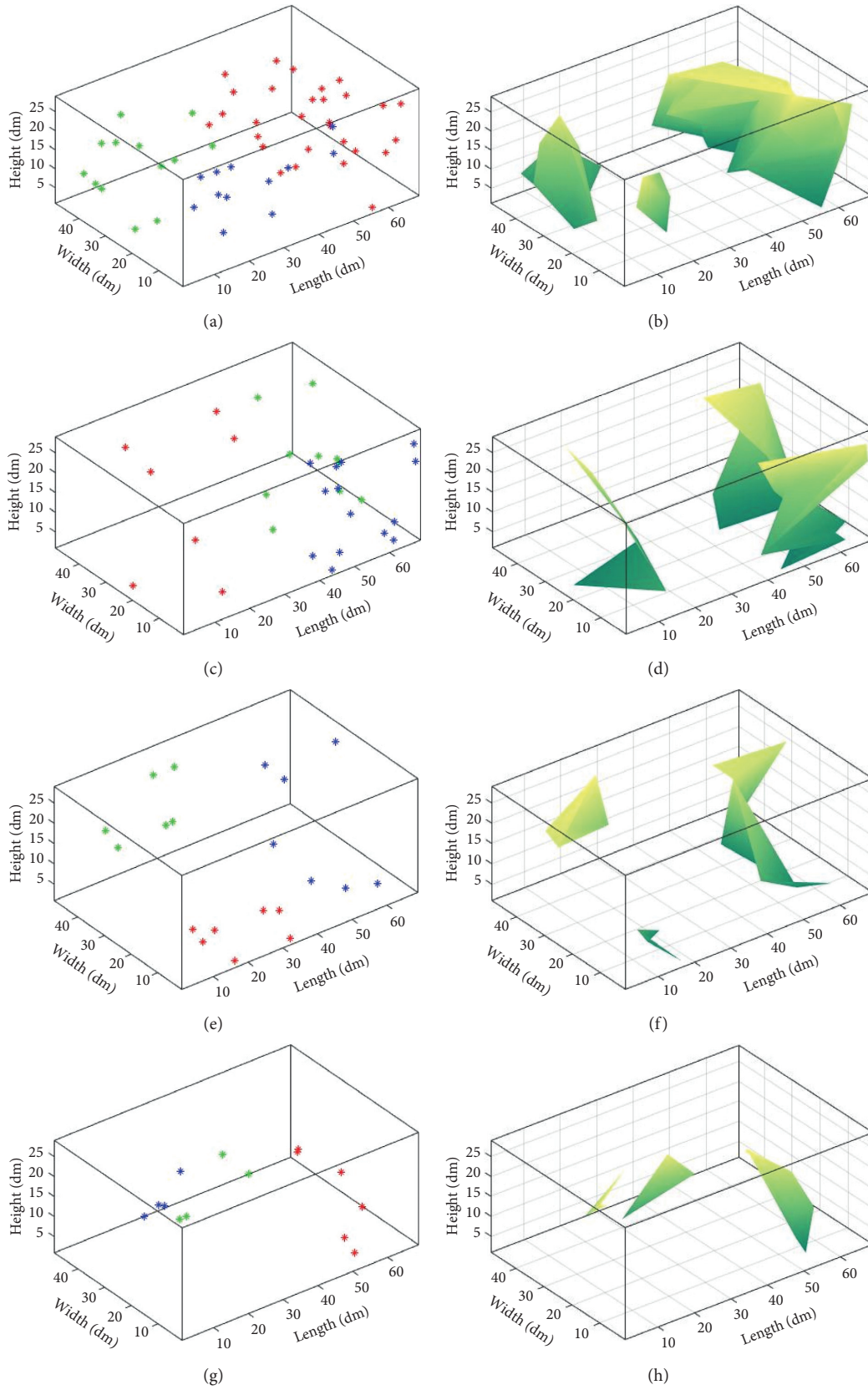


FIGURE 9: The quiet area distribution for the indoor environment after FCM algorithm with different times of (a) 0.03 s, (c) 0.07 s, (e) 0.4 s, and (g) 0.6 s at different rows, respectively, and their visualization results of using the FCM algorithm are shown in (b), (d), (f), and (h).

Figure 7 depicts the direct sound signal that has not reached the received position at  $t = 0.03$  s, which means there are many quiet regions in the room. At  $t = 0.07$  s, the direct signal has nearly reached the received position. Some image signals have reached the received end at  $t = 0.4$  s while the reverberation becomes more severe at  $t = 0.6$  s.

**4.3. Quiet Area Integration.** The distribution density of quiet indoor points represents the comfort of certain areas in space. Because of the line of sight occlusion in three-dimensional space, it is difficult to distinguish the quiet local area with a concentration of points. The quiet local area can be represented intuitively through the quiet points' subdivision and the integration of the quiet areas.

After obtaining the quiet points, we adopt the Delaunay triangulation to subseparate the quiet points set. Firstly, it is integrated into two-dimensional space. During integration, specific spatial points can be integrated into the same point on the plane. As a result, not all the quiet points can be vertices of the Delaunay triangle. Figure 8 shows the outcomes adopted by the Delaunay triangle and the quiet points at  $t = 0.03$  s,  $0.07$  s,  $0.4$  s, and  $0.6$  s separately. It can be seen from the figures that there are a few quiet points in the space, but they still show a particular regional distribution. However, the quiet areas are larger than the others in Figure 8(a). Therefore, it would be convenient to integrate the quiet points set in the space directly. It is necessary to optimize the quiet point set to facilitate the description of the mute area.

**4.4. Optimization of the Quiet Areas.** The quiet points obtained in the discrete acquisitions can either be distributed sparsely throughout the space or be grouped into distinct distribution groups. If we use the data to integrate the area directly, we can obtain nearly the whole area. However, as shown in Figure 8(g), there are only small quiet points. To tackle this issue, we use a FCM clustering algorithm to optimize all the quiet points before we integrate the Delaunay triangulation into the quiet area.

After the data are divided into  $C$  different subsets, the data are divided into different subsets to obtain an accurate quiet area. Quiet points are clustered with the FCM algorithm.

The quiet area in Figure 8 is not perfect when specific points are far away from the others. Furthermore, certain small areas contain many quiet points, whereas other regions have few quiet points. For this case, data with precise subcluster characteristics and continuity of volatility, the triangulation is obtained after separating the quiet points, and we segment the quiet points by the FCM clustering algorithm.

Figure 9 depicts the optimization results of the quiet area at different visual time points after the noise suppression through the combination of clustering algorithm and Delaunay triangulation. In Figure 9, the quiet area will be less as time increases, and it achieved good noise inhibition and has better than that without subset optimization.

## 5. Conclusions

In this paper, a multichannel ANC noise reduction method based on the FxLMS algorithm is realized in small-sized and medium-sized rooms. In addition, to illustrate and optimize the quiet areas in 3D indoor spaces, the combination of the FCM algorithm and Delaunay triangulation is also employed. The experimental results show that the proposed method of signal enhancement performs better than the system based on the LMS algorithm in noise inhibition. This is more conducive to examining the indoor effect and specific distribution of indoor noise reduction through visualization demonstration.

## Data Availability

The data that support the findings of this study are openly available and the details of source acoustic signal are provided in Section 3.1.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61772149, 61936002, 2018AAA0100300, 6202780103, 62033001, 61861008, 11603041, and 62061010) and Guangxi Science and Technology Project (nos. AD18216004, AA18118039, AA19182007, AA19254029, AA20302022, AD18281079, 2018GXNSFAA294054, 2019GXNSFFA245014, and 2019GXNSFBA245072).

## References

- [1] X. Shi, L. Liang, and Y. Liu, "Review of research on noise map application of sound environment," *Architecture & Culture*, vol. 9, pp. 197–198, 2020.
- [2] K. KyooSang, "Source effects and control of noise in indoor/outdoor living environments," *Journal of the Ergonomics Society of Korea*, vol. 34, no. 3, pp. 265–278, 2015.
- [3] Z. Tang, "Status quo, harm and control of noise pollution," *Ecological Economy*, vol. 33, pp. 6–9, 2017.
- [4] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.
- [5] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 208–211, Washington, DC, USA, April 1979.
- [6] H. Gustafsson, S. E. Nordholm, and I. Claesson, "Spectral subtraction using reduced delay convolution and adaptive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 8, pp. 799–807, 2001.
- [7] B. L. Sim, Y. C. Tong, J. S. Chang, and C. T. Tan, "A parametric formulation of the generalized spectral subtraction method," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 4, pp. 328–337, 1998.

- [8] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.
- [9] M. A. Abd El-Fattah, M. I. Dessouky, A. M. Abbas et al., "Speech enhancement with an adaptive Wiener filter," *International Journal of Speech Technology*, vol. 17, no. 1, pp. 53–64, 2014.
- [10] S. J. Elliott and P. A. Nelson, "Active noise control," *IEEE Signal Processing Magazine*, vol. 10, no. 4, pp. 12–35, 1993.
- [11] D. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 454–467, 2003.
- [12] P. Cheng, *Digital Signal Processing*, Tsinghua university press, Beijing, China, 2017.
- [13] C. Sookpuwong and C. Chompoo-Inwai, "Performance comparisons between a single channel feedforward ANC system and a single channel feedback ANC system in a noisy-environment classroom," in *Proceedings of the 2017 International Symposium on Electrical Insulating Materials (ISEIM)*, pp. 11–15, Toyohashi, Japan, September 2017.
- [14] S.-C. Chan and Y. Chu, "Performance analysis and design of FxLMS algorithm in broadband ANC system with online secondary-path modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 982–993, 2012.
- [15] M. Singhal, M. Trikha, A. Pandey, and P. Bhardwaj, "High order X-LMS filter applied for active noise control system," *MIT International Journal of Electrical and Instrumentation Engineering*, vol. 2, no. 2, pp. 94–97, 2012.
- [16] L. Cremer and H. Moller, *Principle and Application of Room Acoustic Design*, Tongji University Press, Shanghai, China, 1995.
- [17] G. Du, Z. Zhu, and X. Gong, *The Basics of Acoustics*, Nanjing University Press, Nanjing, China, 2012.
- [18] A. Craggs, "A finite element method for the free vibration of air in ducts and rooms with absorbing walls," *Journal of Sound and Vibration*, vol. 173, no. 4, pp. 568–576, 1994.
- [19] S. Kopuz and N. Lalor, "Analysis of interior acoustic fields using the finite element method and the boundary element method," *Applied Acoustics*, vol. 45, no. 3, pp. 193–210, 1995.
- [20] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *Journal of the Acoustical Society of America*, vol. 98, no. 98, pp. 3302–3308, 1995.
- [21] J. Forssén, S. Tober, A. C. Corakci, A. Frid, and W. Kropp, "Modelling the interior sound field of a railway vehicle using statistical energy analysis," *Applied Acoustics*, vol. 73, no. 4, pp. 307–311, 2012.
- [22] A. Krokstad, S. Strom, and S. Sørsdal, "Calculating the acoustical room response by the use of a ray tracing technique," *Journal of Sound and Vibration*, vol. 8, no. 1, pp. 118–125, 1968.
- [23] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal Acoustical Society of America*, vol. 65, no. 4, pp. 934–950, 1979.
- [24] M. Vorländer, "Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm," *The Journal of the Acoustical Society of America*, vol. 86, no. 1, pp. 172–178, 1989.
- [25] S. Yan, X. Luo, X. Sun, Z. Tang, J. Jiang, and Y. Li, "Efficient signal distribution of indoor acoustic inhibition using Delaunay triangulation," in *The Eleventh International Conference on Advanced Computational Intelligence*, pp. 217–222, Guilin, China, June 2019.
- [26] Y. Tong, G. U. Yaping, X. Yang, J. Zhang, and S. A. Laboratory, "Design and performance research of reverberation filter system based on source image method," *Journal of Network New Media*, vol. 4, pp. 24–27, 2015.
- [27] E. Y. Kim, B. H. Kim, and S. K. Lee, "Active noise control in a duct system based on a frequency-estimation algorithm and the FX-LMS algorithm," *International Journal of Automotive Technology*, vol. 14, no. 2, pp. 291–299, 2013.
- [28] F. Erkan, *Design and Implementation of a Fixed Point Digital Active Noise Controller Headphone*, Middle East Technical University, Ankara, Turkey, 2009.
- [29] J. Liu, *Performance Analysis of Narrowband Active Noise Control System Based on FxLMS Algorithm*, Harbin Institute of Technology, Harbin, China, 2011.
- [30] S. M. Kuo, *Design of Active Noise Control Systems with the Tms320 Family*, Texas Instruments, Dallas, TX, USA, 2014.
- [31] C. Jordan and S. J. Elliott, "Active noise control of a diesel generator in a luxury yacht," *Applied Acoustic*, vol. 105, pp. 209–214, 2016.
- [32] Y. Oikawa, T. Hasegawa, Y. Ouchi, Y. Yamasaki, and Y. Ikeda, "Visualization of sound field and sound source vibration using laser measurement method," in *Proceedings of the 20th International Congress on Acoustics 2010, ICA 2010 - Incorporating Proceedings of the 2010 Annual Conference*, Sydney, Australia, August 2010.
- [33] R. Wang and S. Bei, "Optimization of fixed microphone array in high speed train noises identification based on far-field acoustic holography," *Advances in Acoustics and Vibration*, vol. 2017, Article ID 1894918, 11 pages, 2017.
- [34] P. Koprinkova-Hristova and K. Alexiev, "Dynamic sound fields clusterization using neuro-fuzzy approach," *Artificial Intelligence: Methodology, Systems, and Applications*, vol. 8722, pp. 194–205, 2014.
- [35] X.-Y. Wang and J. Bu, "A fast and robust image segmentation using FCM with spatial information," *Digital Signal Processing*, vol. 20, no. 4, pp. 1173–1182, 2010.
- [36] L. Hall, "Exploring big data with scalable soft clustering," *Synergies of Soft Computing and Statistics for Intelligent Data Analysis Advances in Intelligent Systems and Computing*, vol. 190, pp. 11–15, 2013.
- [37] O. Tony and R. Imad, "Efficient clustering-based source code plagiarism detection using PIY," *Knowledge and Information Systems*, vol. 43, no. 2, pp. 445–472, 2015.

## Research Article

# LSM-SEC: Tongue Segmentation by the Level Set Model with Symmetry and Edge Constraints

Shanshan Gao <sup>1,2,3</sup> Ningning Guo,<sup>1</sup> and Deqian Mao<sup>1</sup>

<sup>1</sup>Department of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China

<sup>2</sup>Shandong Provincial Key Laboratory of Digital Media Technology, Jinan 250014, China

<sup>3</sup>Shandong China-U.S. Digital Media International Cooperation Research Center, Jinan 250014, China

Correspondence should be addressed to Shanshan Gao; [gss\\_sdufe@sdufe.edu.cn](mailto:gss_sdufe@sdufe.edu.cn)

Received 25 April 2021; Accepted 30 June 2021; Published 29 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Shanshan Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate segmentation of the tongue body is an important prerequisite for computer-aided tongue diagnosis. In general, the size and shape of the tongue are very different, the color of the tongue is similar to the surrounding tissue, the edge of the tongue is fuzzy, and some of the tongue is interfered by pathological details. The existing segmentation methods are often not ideal for tongue image processing. To solve these problems, this paper proposes a symmetry and edge-constrained level set model combined with the geometric features of the tongue for tongue segmentation. Based on the symmetry geometry of the tongue, a novel level set initialization method is proposed to improve the accuracy of subsequent model evolution. In order to increase the evolution force of the energy function, symmetry detection constraints are added to the evolution model. Combined with the latest convolution neural network, the edge probability input of the tongue image is obtained to guide the evolution of the edge stop function, so as to achieve accurate and automatic tongue segmentation. The experimental results show that the input tongue image is not subject to the external capturing facility or environment, and it is suitable for tongue segmentation under most realistic conditions. Qualitative and quantitative comparisons show that the proposed method is superior to the other methods in terms of robustness and accuracy.

## 1. Introduction

Tongue diagnosis is one of the important diagnostic methods of traditional Chinese medicine, while for a long time, tongue diagnosis relied on the doctor's clinical experiences by short-term visual observation, which causes the subjective and uncertain diagnosis results. With the development of image processing and machine learning technology, the research about intelligent assistant diagnosis of tongue manifestation in Chinese medicine has attracted more and more attention. Tongue segmentation from the background with teeth, lips, and face is the important step in the process of computer-aided tongue diagnosis and also an important premise to extract and analyze the color, texture, and morphology features of tongue quality and fur character. However, due to the limitation of the image acquisition process, the tongue with its surrounding tissue in the

tongue image is similar in color and blurred of the outline; it is a challenge to propose an automatic, accurate, and universal tongue segmentation method.

Tongue segmentation is also an image segmentation task. Image segmentation is a process of dividing an image into several homogenous regions that do not overlap each other. It is an important part of the image processing and is of great significance for image analysis, pattern recognition, and computer vision. From the classical image processing methods [1–3] to the deep learning [4–9], image segmentation has been widely concerned and applied. Traditional methods often focus on segmentation based on image features and variable models, and the level set model is one of the most representative methods of the active contour model. In 1987, the active contour models (ACM) were proposed by Kass et al. [10] firstly, which treated the image segmentation as an energy optimization problem and

opened up a new view of image segmentation [10–13]. In recent years, the deep learning method can better perform automatic segmentation and can improve segmentation speed and robustness because of its excellent feature learning and representation ability [6–9]. SegNet [6] provides a full convolution network for pixel level image segmentation, while DFN [7] constructs a smooth network and a border network to form a discriminative feature network. LEDnet [9] adopts an asymmetric codec structure to solve the segmentation task in real-time scenes. These learning-based segmentation methods are widely used in various fields such as medical diagnosis [14].

In fact, in the process of tongue image acquisition, due to the influence of external conditions such as light and temperature, the tongue image is prone to the problems of tongue body position error and spot noise. And, due to the low-contrast characteristic of the tongue image, tongue segmentation is more difficult than conventional image segmentation. Since 1990s, many scholars have made relevant research studies on accurate tongue segmentation [15–22]. In the early stage of research, it is difficult to get accurate results only by using the underlying information of the image. Many subsequent methods improve the accuracy and robustness of segmentation. For example, Huang et al. [19] used the mean shift to smooth the edge and the maximum between-class variance method to classify the image and then merged the regions to extract the tongue. In [20], tongue extraction was based on color building blocks, and sparse representation was used to calculate pixel probability. The method based on deep learning can acquire more image features and has better performance. Huang et al. [21] proposed an automatic tongue image segmentation based on an enhanced full convolutional network. Qu et al. [22] proposed an image quality evaluation method based on brightness statistics to determine whether the input image needs to be segmented and used SegNet to train the tongue dataset. These methods avoid the complicated process of manually extracting features and have obvious advantages in segmentation performance.

It is worth mentioning that the active contour model and some variants began to be applied in the field. For instance, Yang [23] presented a gradient vector active contour model based on the original tongue edge detection method and color gradient and obtained a good segmentation effect. In [24], the original tongue contour was obtained by extracting the ROI of the tongue and using the color similarity of the histogram, and then, the tongue segmentation combining region model and edge model were proposed. From the perspective of transforming the color space model, researchers proposed some effective algorithms based on color information [25–30].

However, it should be noted that the above algorithms usually have certain restrictions and requirements on the tongue image collection environment and the tongue image itself. Therefore, the results of tongue image segmentation with incorrect tongue body position and large noise influence are often not ideal. At the same time, there are other tissues such as peri lip in the image, and the color features of these parts are very close to the tongue itself, which results in

the slow change of the gradient of the tongue edge. This leads to problems such as incomplete segmentation and boundary leakage in level set methods that rely on active contour models or gradient information to extract edges, and the accuracy of segmentation results is difficult to guarantee. Moreover, the active contour model is sensitive to the initial position; then, the adaptability is not satisfied.

In view of the above problems, we propose a symmetry and edge-constrained level set model for tongue segmentation. Different from the traditional level set model, the edge probability value is calculated using the latest convolutional neural network, and the obtained edge probability map is used as the gradient input of the level set. Considering the symmetry characteristics of the tongue, we add a symmetry detection constraint to the level set evolution model to test the symmetry feature of the zero level set contour. A novel level set initialization method is also proposed. It is proved by experiments that this method can complete automatic precise tongue segmentation suitable for most real situations.

## 2. Related Work

Osher and Sethian [31, 32] proposed a level set method based on the important idea of fluid, which solved the problem that the topological structure is not easy to change during image segmentation. The level set method implicitly represents the closed active contour as a zero level set of a higher dimensional level set function and uses the curve evolution to locate the edge of the target. A lot of improvement work related to this appeared later. For example, Li et al. [33] proposed the distance regularized level set evolution (DRLSE) based on distance reinitialization in the process of level set evolution. Zhong et al. [34] proposed a level set method based on region consistency detection by considering the consistency of image region information and achieved good experimental results.

The main idea of the level set method is to regard the physical section moving with time  $t$  as the zero iso-surfaces of the level set function and transform the contour transformation of the  $n$ -dimensional surface into the evolution of the  $n + 1$  dimensional level set function; the boundary of it is expressed by the zero level set of the higher dimensional level set function. The active contour  $C$  is represented as a zero level set of the higher dimensional level set function  $\varphi(x, y, t)$ , denoted as  $C(t) = \{(x, y) | \varphi(x, y, t) = 0\}$ . The purpose of the level set method is to make the zero level set  $C$  meet the partial differential equation of curve evolution:

$$\frac{\partial C}{\partial t} = V(k)N. \quad (1)$$

For the above formula, the evolution equation of the zero level set  $\varphi$  under the velocity function  $F$  is

$$\frac{\partial \varphi}{\partial t} + F|\nabla \varphi| = 0. \quad (2)$$

The velocity function  $F$  depends on the image data and the level set function  $\varphi$ . In the image segmentation,  $F$

generally contains the curvature  $k$  of the evolution curve  $C$ . The curvature is calculated as follows:

$$k = \operatorname{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right) = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_x\varphi_y\varphi_{xy} + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}}. \quad (3)$$

If it is under the average curvature, the evolution equation can be written as

$$\frac{\partial\varphi}{\partial t} = |\nabla\varphi| \operatorname{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right). \quad (4)$$

One advantage of the level set method is that the calculation of curves and surfaces can be performed on a fixed Cartesian grid, and the evolution of the curve is independent of parameters. However, in the conventional level set methods, with the evolution of the curve, the level set function will no longer remain as the signed distance function. Therefore, it is necessary to periodically initialize the level set function to the signed distance function during the curve evolution. The process of reinitialization can affect the accuracy of the calculation, and it is time-consuming. To solve this problem, Li et al. added a distance regularization term to the conventional level set model and proposed a DRLSE [33] model without reinitialization.

The energy function of the DRLSE model is as follows:

$$\begin{aligned} \varepsilon_{\text{DRLSE}}(\varphi) &= \mu R_p(\varphi) + \lambda L(\varphi) + \alpha A(\varphi) = \mu \int_{\Omega} p(|\nabla\varphi|) dx \\ &+ \lambda \int_{\Omega} g\delta(\varphi)|\nabla\varphi| dx + \alpha \int_{\Omega} gH(-\varphi) dx, \end{aligned} \quad (5)$$

where  $\mu, \lambda$ , and  $\alpha$  are constant with positive values, representing the weight of each energy term.

The first term is a regularization term that constrains the deformation of the curve by guaranteeing the signed distance property  $|\nabla\varphi| = 1$ . It is not necessary to reinitialize the level set function acyclically after adding the regularization term.

The second term is used to drive the zero level set to evolve towards the target edge. The function  $g$  is a boundary stop function based on the image gradient. Once the zero level set curve arrives at the target boundary, the energy function of the length term is the smallest.

The third term is the area term, which can accelerate the convergence of the zero level contour during the evolution of the level set. When initializing the level set, if the target is completely inside the initial curve,  $\alpha$  should take a positive number to ensure that the curve converges inward; on the contrary, it should take the negative number.

The edge stop function  $g$  is defined as

$$g = \frac{1}{1 + |\nabla G_{\sigma} * I|^2}, \quad (6)$$

where  $I$  is the image to be segmented and  $G_{\sigma}$  is the standard deviation of the Gaussian filter.

According to the variational theory, in order to solve the gradient descent flow of energy functional, the following level set evolution equation is obtained as

$$\begin{aligned} \frac{\partial\varphi}{\partial t} &= -\frac{\partial\varepsilon_{\text{DRLSE}}(\varphi)}{\partial\varphi} = \mu \operatorname{div}(d_p(|\nabla\varphi|)\nabla\varphi) \\ &+ \lambda\delta_{\varepsilon}(\varphi) \operatorname{div}\left(g\frac{\nabla\varphi}{|\nabla\varphi|}\right) + \alpha g\delta_{\varepsilon}(\varphi), \end{aligned} \quad (7)$$

where the Heaviside function  $H_{\varepsilon_U}(x)$  (see the third term in formula (5)) is used to divide the evolution region in the level set evolution, and the Dirac function  $\delta_{\varepsilon_U}(x)$  is the derivative function of the Heaviside function, which is used to constrain the evolution value. They are formulated by the following smooth functions:

$$\begin{aligned} H_{\varepsilon}(x) &= \begin{cases} \frac{1}{2}\left(1 + \frac{x}{\varepsilon} + \frac{1}{\pi}\sin(\varepsilon)\right), & |x| \leq \varepsilon, \\ 1, & x > \varepsilon, \\ 0, x & x < -\varepsilon, \end{cases} \quad (8) \\ \delta_{\varepsilon}(x) &= \begin{cases} \frac{1}{2\varepsilon}\left(1 + \cos\left(\frac{\pi x}{\varepsilon}\right)\right), & |x| \leq \varepsilon, \\ 0, & |x| > \varepsilon. \end{cases} \end{aligned}$$

In recent years, the active contour model, level set, and some variants have been applied to tongue segmentation; Li [25] added the prior information of the difference between tongue and other parts in HSV color gamut to the level set model and proposed a new region-based bounded pressure function. Shi et al. [28] combined the geometric snake model with the parameterized GVFSnake [27] model and built the C2G2FSnake model, which improved the segmentation accuracy.

### 3. The Proposed New Method

Due to the speciality of the pathological tongue and the limitation of image acquisition equipment, the difficulties of tongue segmentation are mainly reflected in the following aspects:

(1) The color of the tongue is similar to the surrounding tissues in the image background, so the color contrast is low. (2) The position of the tongue is not correct, and the spot noise is common in tongue segmentation. (3) The surface of the tongue has a thick coating or the tongue is cracked in the middle of the tongue. These factors lead to small differences in gradient values; then, the gradient map of the tongue is blurred. Therefore, the segmentation curve usually cannot be accurately stopped at the edge of the target contour, which greatly increases the difficulty of tongue segmentation. Figure 1 shows the segmentation results of the DRLSE method for low-contrast, speckle noise, and thick coating images.

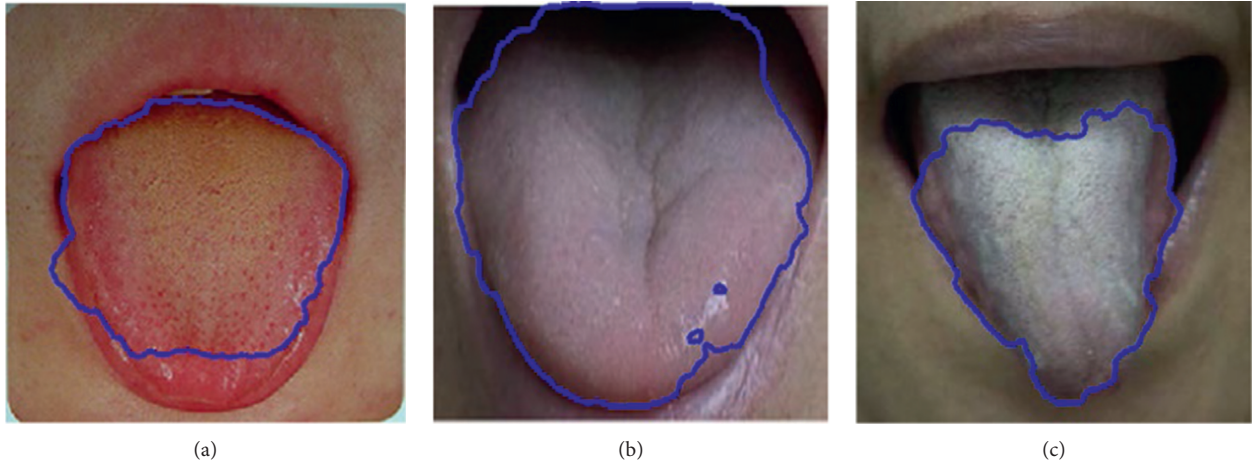


FIGURE 1: (a) Low-contrast tongue. (b) Speckle noise tongue. (c) Thick coating tongue.

The level set method can calculate curves and surfaces on fixed Cartesian meshes and can deal with various topological changes as well, which is very suitable for medical image segmentation. As well known, the geometric characteristics of an image can clearly reflect the structural and content characteristics of the image and can prevent the image texture from being easily affected by interference factors such as light and noise. As the basic shape characteristic of an object, symmetry is ubiquitous in the nature, and the tongue as a part of the human body has the characteristics of mirror symmetry and rotation invariance. Therefore, the extraction of symmetry geometric features of the tongue image has a good guiding effect on tongue segmentation. Based on the above findings, the symmetry information is added to the level set model as a constraint by using the symmetry characteristics of the tongue image. At the same time, in recent years, the convolutional neural network has shown its unique advantages in complex or low-contrast image processing. Therefore, this paper combines the convolutional neural network model with the level set model, by taking the gradient map of neural network training as the gradient input of the level set to guide the curve evolution, and then proposes a symmetry and edge-constrained level set model for tongue segmentation.

The principle of the symmetry detection constraint is that, during the process of level set segmentation, if the zero level set curve evolves to a weak gradient or strong noise, at that time, the zero level set function does not maintain symmetry; then, the constraints on the incomplete side of the segmentation and the energy will increase under the combined action of internal energy and external force of image symmetry. Conversely, if the zero level set function remains symmetrical during the segmentation process, the constraint term does not participate in the evolution process. Meanwhile, the evolution process of the level set function is to solve the DRLSE energy function of the minimized closed curve.

The initial contour is usually fixed to a rectangular area at an arbitrary position in traditional methods. In fact, the selection of the initial contour has a great impact on the segmentation results. Inappropriate position of the initial curve will cause the level set function to fall into a local minimum position. In this paper, we also extract the initial contour of the level set in a novel way. According to the characteristics of the constructed symmetry detection energy item, we first obtain the symmetry axis of the tongue body and set the initial contour curve as a circular region.

The key steps of the algorithm include the following: the convolution neural network is used to train the color tongue image, and the edge probability map is obtained as the gradient image input of the level set model. Then, we use the mirror symmetry of the tongue image to select the symmetry axis of the tongue automatically and take the symmetrical axis as the centre of the circle to get the initial contour curve located in the centre of the tongue. During the process of evolution, the evolutionary image and the gradient image are reflected and transformed, and the symmetry detection energy term is constructed to constrain the level set evolution. Finally, we use the variational method to solve the gradient descent flow of the energy function as to obtain the target boundary. The specific flow of the algorithm is shown in Figure 2.

#### 4. The Symmetry and Edge-Constrained Level Set Model for Tongue Segmentation

Compared with the traditional model, our improvements are as follows: adding the symmetry detection constraints, putting forward a symmetry and edge-constrained level set model, determining the symmetry axis of the tongue and changing the initial contour curve accordingly to match the functional characteristics of the symmetry constraint item, and combining the deep learning method to train the gradient input to improve edge accuracy. Next, the technical



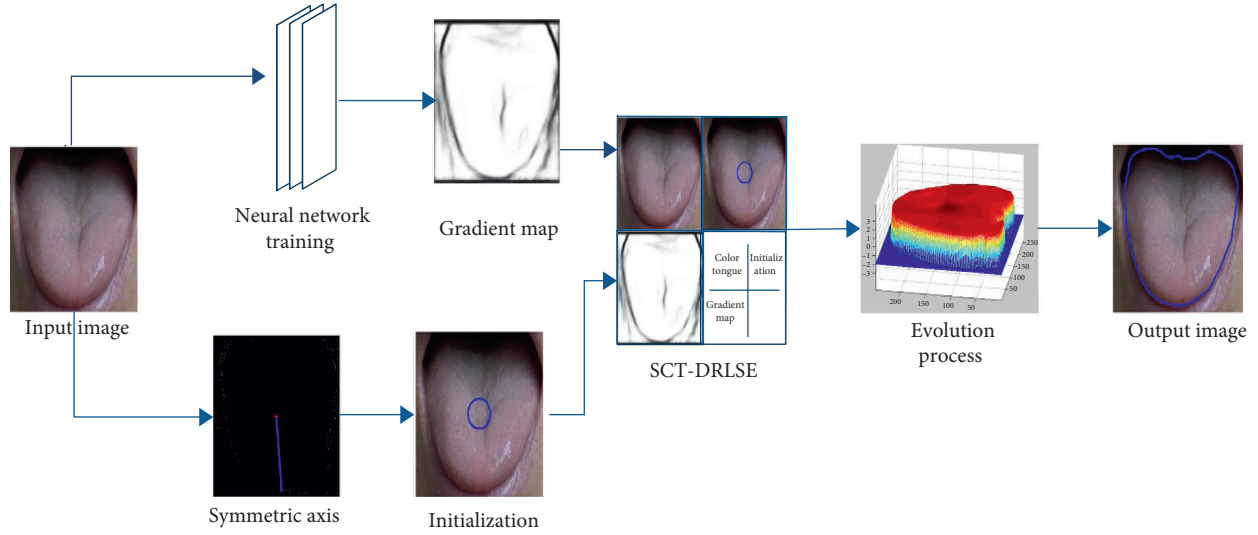


FIGURE 2: Flowchart of symmetry and edge-constrained level set model for tongue segmentation.

details will be described in the above order, not the order of the algorithm implementation steps.

**4.1. Symmetry Detection Constraint.** As mentioned above, the tongue has obvious symmetry; the symmetry detection constraint is used to detect the symmetry of level set function  $\varphi$  under the gradient image on both sides of the symmetry axis. The proposed detection constraint is based on a simple but important fact: if a plane geometry figure is approximately symmetric about the major axis, there must be a reflection transformation that minimizes the error of the transformed figure aligned with the original figure. Then, the difference between the energy of the DRLSE level set function and the energy after reflection transformation to the energy function is added as a symmetry detection constraint. The essence of the constraint is to evaluate the approximate symmetry of the target contour in the segmentation process.

Axis reflection transformation on the Euclidean plane and mirror reflection transformation in the Euclidean space are called reflection transformation. Reflection transformation is an important transformation in the Euclidean geometry. In this paper, the reflection transform is a horizontal mirror transform. Specifically, the symmetry axis of the image is used as the transformation axis to swap the pixels of the image. The original level set and the transformed level set in the evolution process are shown in Figure 3. The matrix  $M$  of the reflection transformation is expressed as

$$\frac{1}{A^2 + B^2} \begin{bmatrix} B^2 - A^2 & -2AB & -2AC \\ -2AB & A^2 - B^2 & -2BC \\ 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

where  $A$ ,  $B$ , and  $C$  are the coefficients of the general formula of the straight line  $l$ , and the calculation formulas of the coefficients  $A$ ,  $B$ , and  $C$  are as follows:

$$\begin{cases} A = (y_1 - y_2), \\ B = (x_2 - x_1), \\ C = (x_1 * y_2 - x_2 * y_1). \end{cases} \quad (10)$$

In the evolution process of the level set function, if  $Q(x, y)$  is a point on the zero level set, the calculation formula of the new coordinate  $Q(x', y')$  after the reflection transformation with  $l$  as the symmetry axis is as follows:

$$Q' = M * Q. \quad (11)$$

If an image  $I$  is symmetrical,  $\hat{I}$  is defined as a symmetrical complementary image of the source image, and the position of the pixels on  $\hat{I}$  is derived from equation (11). The coordinates of point  $Q(x, y)$  for reflection transformation are as follows:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \frac{1}{A^2 + B^2} \begin{pmatrix} B^2 - A^2 & -2AB & -2AC \\ -2AB & A^2 - B^2 & -2BC \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (12)$$

For any given signed distance function, a transformed signed distance function that preserves shape invariance can be obtained by the above transformation. In the evolution process,  $\varphi$  is used to represent the priori shape, and the Heaviside function of  $\varphi$  in the gradient image is denoted as  $H_\varepsilon(-\varphi)g$ . According to the above reflection transformation formula, the symmetric complementary term is  $\hat{H}_\varepsilon(-\varphi)\hat{g}$ . For solving the symmetric complementary term, the value of matrix  $M$  will be updated with the iteration of level set function  $\varphi$ . The definition of the symmetry detection constraint is shown in the formula:

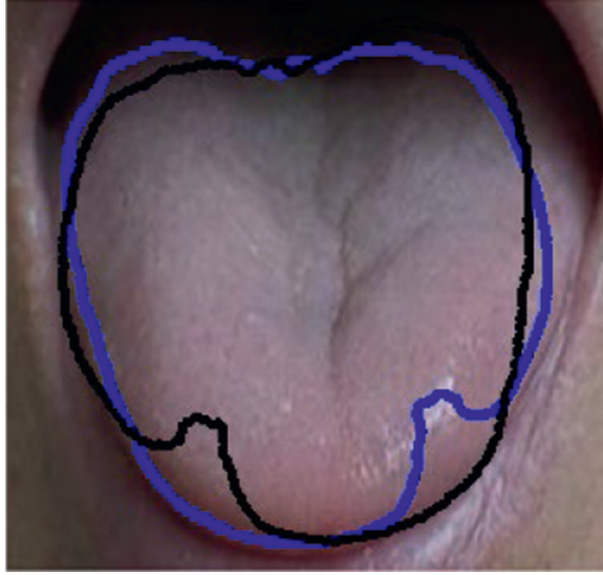


FIGURE 3: Schematic diagram of horizontal set reflection transformation.

$$S_g(\varphi) = \eta \int_{\Omega} (H_{\varepsilon}(-\varphi)g - \widehat{H}_{\varepsilon}(-\varphi)\widehat{g})^2 dx, \quad (13)$$

where  $\eta$  is a positive number and  $H_{\varepsilon}(-\varphi)$  is a Heavyside function of the level set  $\varphi$ .

In image domain  $\Omega$ , we measure the approximate symmetry of the target's own contour by the difference

between the original level set function and the reflected transformed level set function. The computation on difference is completed by the symmetry detection constraint.

The energy function of the symmetry detection level set model is expressed as follows:

$$\begin{aligned} \varepsilon_{\text{SCT-DRLSE}}(\varphi) &= \mu R_p(\varphi) + \lambda L_g(\varphi) + \alpha A_g(\varphi) + \eta S_g(\varphi) \\ &= \int_{\Omega} p(|\nabla\varphi|)dx + \lambda \int_{\Omega} g\delta_{\varepsilon}(\varphi)|\nabla\varphi|dx \\ &\quad + \alpha \int_{\Omega} gH_{\varepsilon}(-\varphi) + \eta \int_{\Omega} (H_{\varepsilon}(-\varphi)g - \widehat{H}_{\varepsilon}(-\varphi)\widehat{g})^2 dx, \end{aligned} \quad (14)$$

where  $\lambda$ ,  $\alpha$ , and  $\eta$  are the coefficients of each energy term, the first three terms of the formula belong to the DRLSE model, and the last one is the symmetry detection constraint term (SCT).

The optimization of this energy function can be obtained with the following gradient flow descent method:

$$\begin{aligned} \frac{\partial\varphi}{\partial t} &= -\frac{\partial\varepsilon_{\text{SCT-DRLSE}}(\varphi)}{\partial\varphi} = -\frac{\partial\varepsilon_{\text{DRLSE}}(\varphi)}{\partial\varphi} - \eta \frac{\partial S}{\partial\varphi} \\ &= \mu \text{div}(d_p(|\nabla\varphi|)\nabla\varphi) + \lambda \delta_{\varepsilon}(\varphi) \text{div}\left(g \frac{\nabla\varphi}{|\nabla\varphi|}\right) \\ &\quad + \alpha g \delta_{\varepsilon}(\varphi) + 2\eta \delta_{\varepsilon}(\varphi) (H(\varphi) - H(\widehat{\varphi})). \end{aligned} \quad (15)$$

Obviously, the higher the symmetry of level set function  $\varphi$ , the smaller the value of SCT and the less the energy of constraints on evolution. If the image information is asymmetric, such as when the curve evolves to weak edges or tongue noise and tongue cracks, the symmetry of level set

function  $\varphi$  decreases, and then, the value of SCT increases, which promotes the evolution of the side curve under the effect symmetry detection constraints.

A schematic diagram of LSM-SEC level set evolution is given in Figure 4.

**4.2. Automatic Determination of Initial Contour.** In the actual process of image acquisition, the tongue is usually captured in the middle of the image. Considering this prior knowledge, the symmetry axis from the tongue gradient image is first extracted, and it is used as the reflection transformation reference line of the symmetry detection constraint (SCT). Considering the feature of the symmetry detection constraint (SCT), in order to maintain the original image force of the level set function under the initial condition, we set the initial level set function as a circular contour fixed in the target area.

The extraction of the symmetrical axis can be divided into two steps: fixed axis and direction, that is, determining

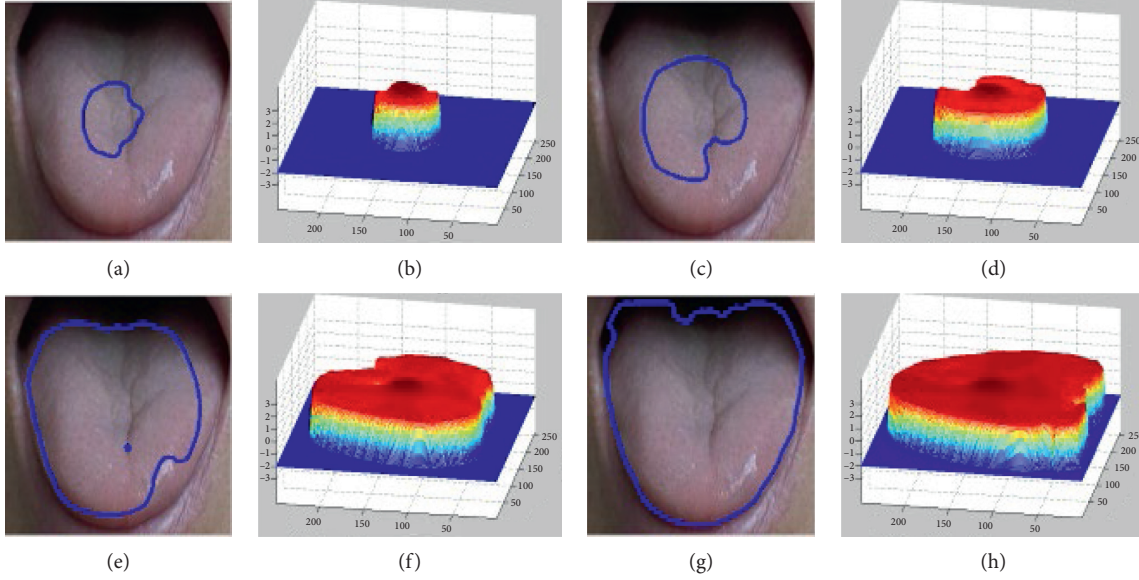


FIGURE 4: The behavior of level set evolution in the LSM-SEC: (a) the contour of the level set after 10 iterations, (b) the level set function after 10 iterations, (c) the contour of the level set after 80 iterations, (d) the level set function after 80 iterations, (e) the contour of the level set after 300 iterations, (f) the level set function after 300 iterations, (g) the contour of the level set after 600 iterations, and (h) the level set function after 600 iterations.

the position of the symmetrical axis and the direction of the symmetrical axis. Since the centre of gravity of an axis-symmetric figure must be on the symmetrical axis, the position of the symmetrical axis is determined by calculating the centre of gravity  $w_1(x_1, y_1)$  of the gradient image:

$$\begin{aligned} X_1 &= \frac{\sum P_i x_i}{\sum P_i}, \\ Y_1 &= \frac{\sum P_i y_i}{\sum P_i}, \end{aligned} \quad (16)$$

where  $(x_i, y_i)$  is the coordinates of the pixel and  $p_i$  is the pixel values.

The corner point is generally considered to be the point at which the brightness of the image changes abruptly or the point at which the curvature of the edge curve is maximum. The Harris corner detection algorithm is used to find the potential tip point in the middle position of the tongue. The Harris corner detection [35] algorithm defines a corner as a point whose gray value can be greatly changed by micro offset in any direction. The Harris corner detection algorithm assumes that the pixel gray value of point  $(x, y)$  is  $I(x, y)$ , and the change of gray intensity of each pixel  $(x, y)$  moving  $(u, v)$  in the image is expressed as a differential operator:

$$E_{(x,y)} = \sum_{u,v} w(u,v) |I(x+u, y+v) - I(u,v)|^2, \quad (17)$$

where  $w(u, v)$  is the coefficient of the filter window.

The rule of tracking the tip of the tongue with the Harris corner detector is searching for  $k$  pixels on the left and right sides of the middle position of the image data. For each

current point projection to the  $y$ -axis, set the  $y$ -axis threshold and find its mean coordinate  $w_2(x_2, y_2)$ . Determine the axial direction by finding the position average of the potential tip point. The acquisition of the symmetry axis is shown in Figure 5.

It is known that the barycentric coordinates are  $v_1(x_1, y_1)$  and the tongue tip coordinates are  $v_2(x_2, y_2)$ . According to the general equation  $Ax + By + C = 0$  of the straight line, a straight line equation that can obtain two points passing  $v_1$  and  $v_2$  can be expressed as

$$l: (y_1 - y_2) * x + (x_2 - x_1) * y + (x_1 * y_2 - x_2 * y_1) = 0. \quad (18)$$

Taking the line  $l$  as the axis of symmetry, the point on the zero level set function is transformed.

The initial contour of level set evolution is fixed in the target area, by choosing the axes of the symmetry axis. The initial contour shape is set as a circular region with the axes as the centre, which ensures that the symmetry detection constraint does not act on the level set function  $\phi$  in the initial segmentation state and maintains the image force of the original evolution process.

With the source image  $M$  and symmetry axis  $l$ , the intersection of the line  $l$  and the source image  $M$  is denoted by  $(x_a, y_a), (x_b, y_b)$ , and the calculation formula of the axis coordinate  $O$  of the symmetry axis is

$$O(X, Y) = \left( \frac{(x_a + x_b)}{2}, \frac{(y_a + y_b)}{2} \right). \quad (19)$$

In general, the initial level set function is set as the signed distance function (SDF), which is defined as follows:

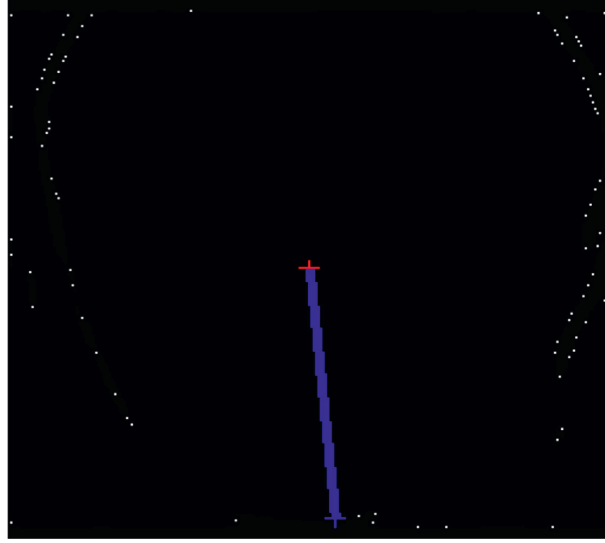


FIGURE 5: Center of gravity detection and corner detection.

$$\varphi(x, y) = \begin{cases} -d((x, y), C), & (x, y) \in \text{inside}(C), \\ 0, & (x, y) \in C, \\ +d((x, y), C), & (x, y) \in \text{outside}(C). \end{cases} \quad (20)$$

The signed distance function satisfies  $|\nabla\varphi| = 1$ , where  $d$  is the Euclidean distance from the point  $(x, y)$  to the zero level set.

In this paper, the symbol distance function is defined as a circular initialization level set function with the axis  $O$  as the centre and  $R$  as the radius. The expression is as follows:

$$d(X, Y) = \sqrt{X^2 + Y^2} - R. \quad (21)$$

#### 4.3. Gradient Image Based on Edge Probability Prior.

From the definition of the boundary stopping function, one can see that the accuracy of the tongue gradient map is very important to the segmentation results. The traditional level set method directly calculates the partial derivative of the original image in the horizontal and vertical directions to obtain the gradient, but at the fuzzy boundary or the discrete edge, the segmentation result is limited with small gradient change of the target tongue.

Convolution neural networks (CNNs), as a kind of deep network, have been widely used in image processing and pattern recognition in recent years. The basic structure of a convolutional neural network generally includes a convolutional layer, pooling layer, and fully connected layer. Given by that the traditional CNN edge detection method only uses the features of the last convolutional layer as the output, many features and details are lost in the convolution process. Liu et al. [36] proposed an edge detector using a richer convolution feature (RCF). The RCF network makes full use of multiscale and multilevel information, combines all meaningful convolution features in a holistic manner to perform edge detection, and obtains a clear probability

boundary. The network achieved the best detection results on the BSDS500 database.

In this paper, the multilayer network structure features of the RCF network model are used to obtain the edge probability map, which is used as the gradient image input of the level set to guide the evolution of the edge stop function.

The RCF network model uses the characteristics of the multilayer network structure to obtain an edge probability map. RCF is based on the VGG16 network, which consists of five modules, alternating convolutional and pooled layers and three fully connected layers. The first two modules contain two subconvolution layers with the same parameters, and the last three modules contain three subconvolution layers with the same parameters. The subconvolution layer features of each module are added pixel by pixel using else layer, and the results are fused. Different scale features can be obtained by sampling under the maximum pooling layer.

Different from the traditional VGG16 network structure, the RCF replaces the pooled layer and the fully connected layer of the fifth module with a convolutional layer of size  $1 \times 1$  so that the training result retains spatial information. RCF also proposes a new loss function for each module, avoiding the gradient disappearance problem during network training. The loss function is defined as follows:

$$l(X_i; W) = \begin{cases} \alpha \cdot \log(1 - P(X_i; W)), & \text{if } y_i = 0, \\ 0, & \text{if } 0 < y_i \leq n, \\ \beta \cdot \log P(X_i; W), & \text{otherwise,} \end{cases}$$

$$\text{in which } \alpha = \lambda \cdot \frac{|Y^+|}{|Y^+| + |Y^-|},$$

$$\beta = \frac{|Y^-|}{|Y^+| + |Y^-|}.$$

(22)

where  $Y^+$  and  $Y^-$  represent a positive sample set and a negative sample set, respectively. The superparameter  $\lambda$  is used to balance the positive and negative samples.  $P(X)$  is a standard sigmoid function. RCF uses the Caffe deep learning framework, the other parameters are the same as the Caffe model, and the training experiments were performed using the NVIDIA TITAN X GPU. The RCF network structure diagram is shown in Figure 6.

The edge probability map trained by the RCF network is used as the gradient image input of LSM-SEC, replacing the original gradient input in the edge stop function. The Gaussian filtering of the RCF gradient image is carried into equation (14), and the evolution of the edge curve is guided by iteratively calculating the edge stop function in the process of level set evolution. As shown in Figure 7,  $a$  is the original image and  $b$  is the tongue gradient image acquired by the RCF.

## 5. Experimental Results and Analyses

In the experiment, the tongue image dataset contains 550 tongue images, part of which is from GitHub's open-source dataset, with a total of 300 tongue images; the other part is provided by the teachers of the University of traditional Chinese medicine, with a total of 250 tongue images. The images in the dataset are different in size, shape, angle, and position, but they all contain the complete tongue body, which is suitable for this experiment. Due to the need of the follow-up experiments, the tongue images were flipped, randomly cropped, rotated, and other operations were performed to expand the dataset, and finally, 1100 images were obtained. The "ground truth" of each tongue image is manually marked by experts. In this section, we will make qualitative and quantitative analysis of the experimental results.

The experimental environment of the algorithm is MATLAB R2010b; the machine system: win7; memory: 4 GB. In RCF training, the weight of the  $1 \times 1$  convolution layer in stages 1–5 is subject to a zero-mean Gaussian distribution, the standard deviation is initialized to 0.01, and the deviation is initialized to 0. Because the dataset is relatively small, the ratio of training data and test data is 7:3. All parts of the neural network in this paper are completed by NVIDIA TITAN X GPU.

The parameters of the experiment are set as follows: the time step of the level set is  $\Delta t = 1$ , the regularization parameter is  $\varepsilon = 1.5$ , the length penalty term parameter is  $\lambda = 2$ , the weighted area term is  $\alpha = -2$ , the distance regularization coefficient is  $\mu = 0.2$ , and the convolution calculation window size is  $\sigma = 1$ . The above parameters all maintain the original DRLSE method parameter settings, and the symmetry detection constraints' parameter is  $\eta = 1$ .

**5.1. Qualitative Analysis.** We compare the proposed method with three other classical methods, including distance rule level set evolution (DRLSE) method [33], maximal similarity-based region merging (MSRM) [37], automatic tongue image segmentation utilizing prior knowledge (C2G2FSnake) [28],

and SegNet-based method proposed in [22]. The results of tongue segmentation are shown in Figure 8. It can be analyzed that the MSRM method is not effective for most tongue segmentation, the contour curve is not completely consistent with the tongue boundary, and the segmentation accuracy is low. As shown in the third row (c) of Figure 8, the tongue and upper lip portions are not identified, and the thick coated tongue of the row (3) and column (e) differs greatly from the true boundary. It can be seen from the row (4) that SegNet can hardly distinguish the background around the tongue, especially in the row (4) and the columns (e) and (f), teeth and lips are not recognized. The edge of the DRLSE-divided tongue is smooth, but since the DRLSE method only uses the gradient information and does not combine high-level features such as color information, the result of segmentation between the low-contrast and low-gradient portions of the tip and the lip is poor. As shown in the low-contrast tongue images in columns (a) and (b) of row (5) of Figure 8, the DRLSE method does not accurately segment the portion of the tongue that is similar in color to the circumference of the lips. From the segmentation result of row (6), we can see that the C2G2FSnake method preserves the main contour of the tongue better, but this method still cannot solve the noise interference on the edge of the tongue, such as the thick coated tongue image of the fifth line. In addition, due to the low robustness of the C2G2FSnake tipping point finding method, these results in the segmentation extraction results are often not obtained during the segmentation process. In the experimental dataset, other results of the method of this paper are shown in Figure 9.

Combining the experimental results of Figures 8 and 9, we can conclude that the edge of the target tongue extracted by the method is smooth and can effectively copy with the tongue crack of the pathological tongue, such as the third line (d) and (f) image above. At the same time, by observing the pictures in the second row (d) column and the fourth row (b) column, it can be noted that the method in this paper is insensitive to the spot noise appearing on the surface of the tongue, which solves the problem of spot segmentation caused by the DRLSE method. On the qualitative point of view, the LSM-SEC method is superior to the other three methods in processing low-contrast tongue images, which greatly improves the segmentation accuracy. The accuracy is partly due to the fact that the level set method is more suitable for the change of the tongue contour topology, and the gradient image input makes the segmentation result insensitive to the cracks and thick coating on the surface of the tongue, and the contour is more stable. On the contrary, the symmetry detection constraint enables the segmentation curve to maintain a good symmetry characteristic of the original tongue at a weak gradient. In a word, we can see from the results that our method is relatively universal and can extract accurate tongue from the surrounding environment.

**5.2. Quantitative Analysis.** In order to quantitatively measure the segmentation performance of the proposed method, we use the reca, prec, IoU, and  $F1$ -measure indicators to

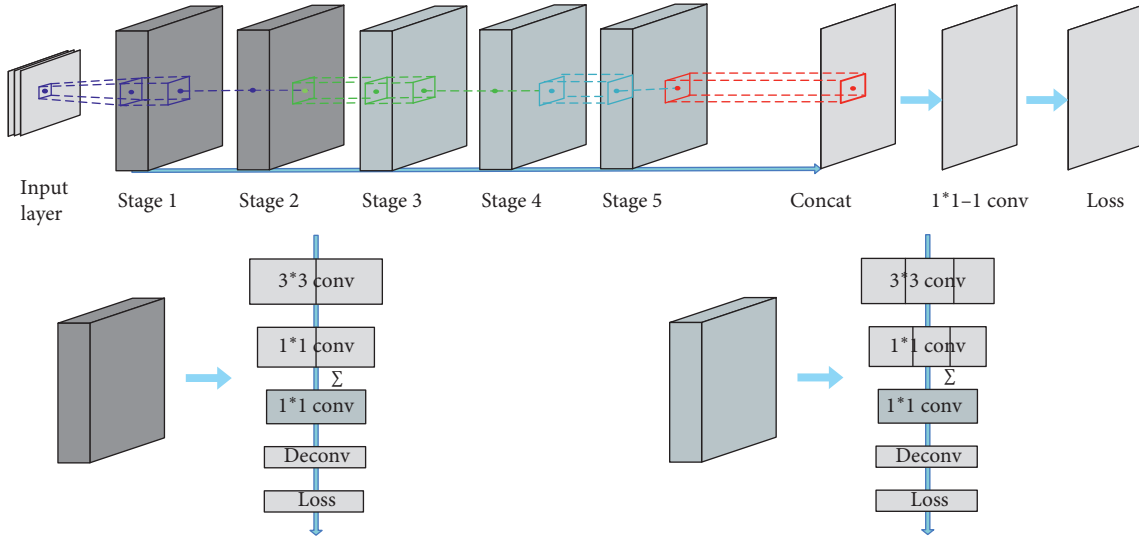


FIGURE 6: RCF network structure diagram.

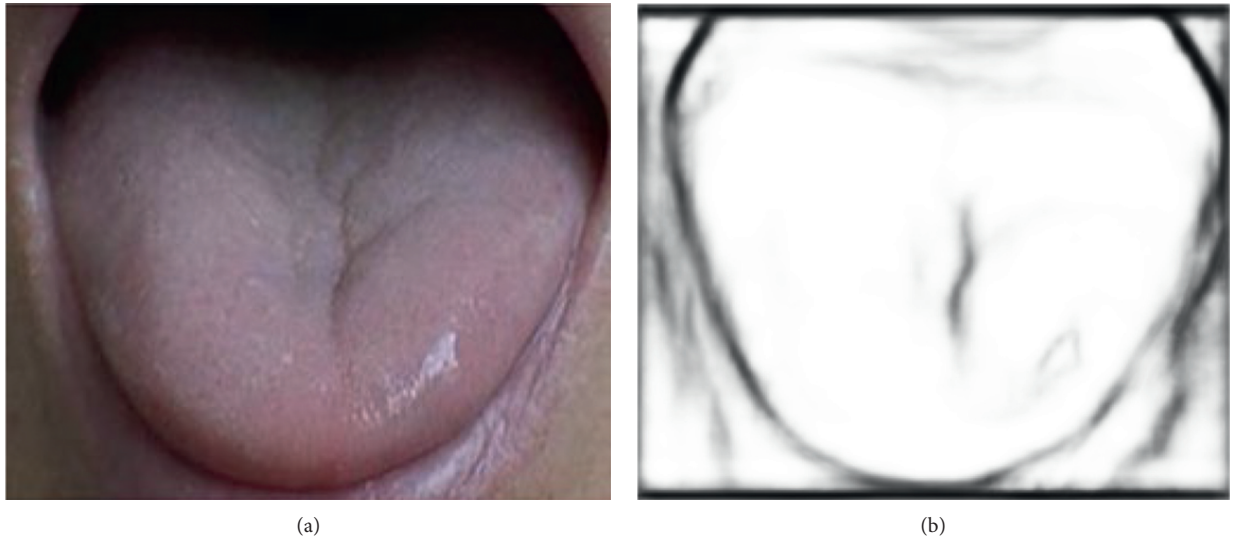


FIGURE 7: (a) Original image. (b) Gradient image.

compare and analyze the segmentation accuracy of the four methods. Prec, reca, and IoU are the precision, recall, and cross ratio, respectively, and  $F1$ -measure is the harmonic mean of the accuracy and recall. The accuracy of segmentation represents the proportion of the real target region in the segmentation result, and the recall ratio represents the proportion of the segmentation result in the real target region.  $F1$ -measure is the weighted harmonic average of accuracy and recall, while IOU is the intersection and parallelism ratio of the real area and segmentation area. The four indicators reflect the accuracy of the segmentation method, which are defined as follows:

$$\begin{aligned}
 P &= \frac{TP}{TP + FP}, \\
 R &= \frac{TP}{TP + FN}, \\
 F_1 &= \frac{2 * PR}{P + R}, \\
 IOU &= \frac{A_a \cap A_b}{A_a \cup A_b}.
 \end{aligned} \tag{23}$$

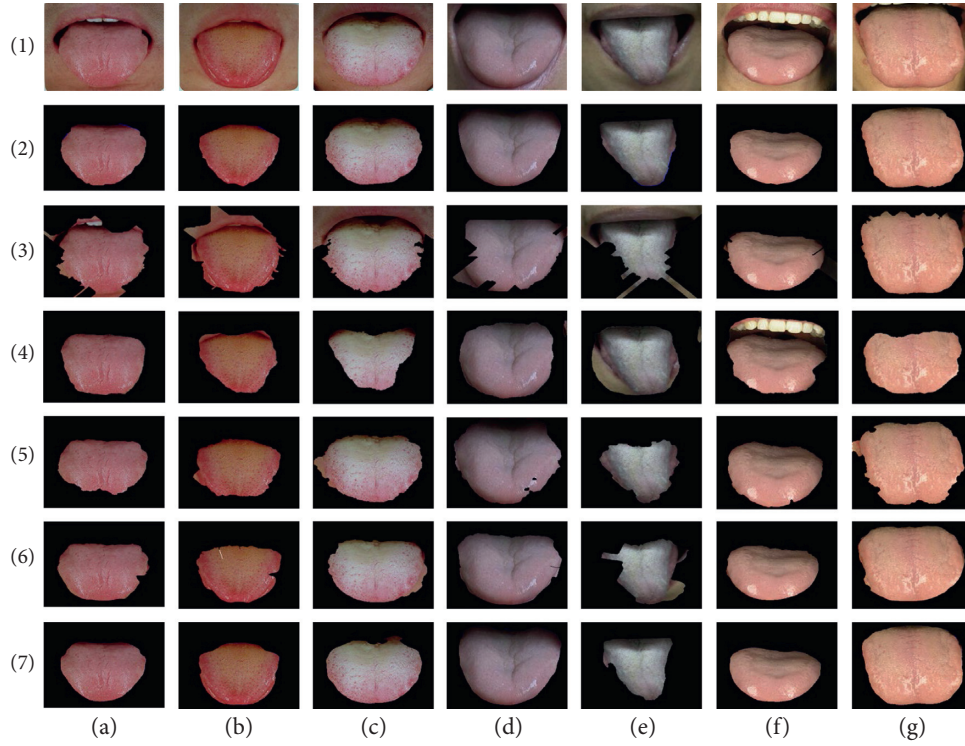


FIGURE 8: Comparison of the results of tongue image segmentation method. Column (a)-(b) are seven types of tongue image. Row (1) is the original image, row (2) is the ground truth, and rows (3)–(7) are the segmentation results of MSR, SegNet, DRLSE, C2G2FSnake, and our LSM-SEC, respectively.

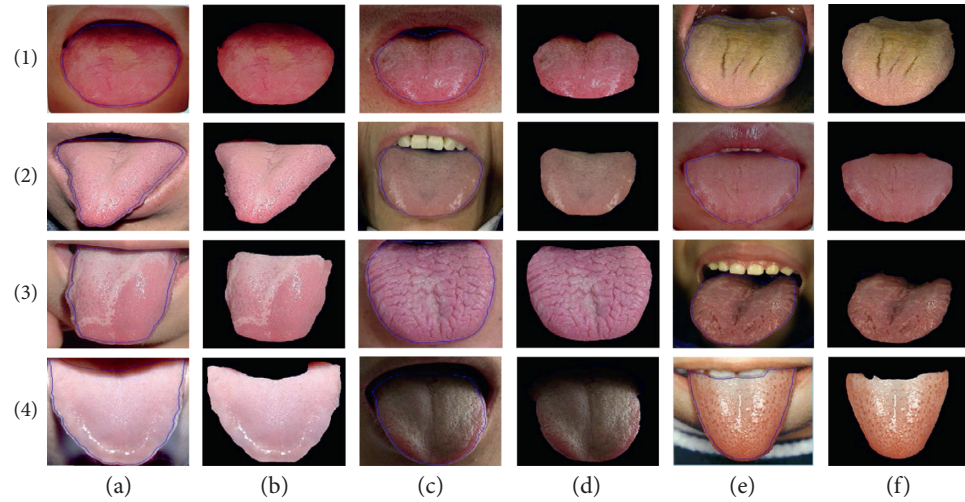


FIGURE 9: LSM-SEC segmentation result. Items (a), (c), and (e) are listed as real images, and items (b), (d), and (f) are listed as LSM-SEC segmentation results.

TABLE 1: Quantitative results.

	<i>F</i> -measure	IOU	Prec	Reca
MSRM	0.856	0.759	0.842	0.885
DRLSE	0.909	0.834	0.925	0.897
C2G2FSnake	0.899	0.820	0.899	0.904
SegNet	0.838	0.727	0.905	0.802
LSM-SEC	0.963	0.930	0.972	0.956

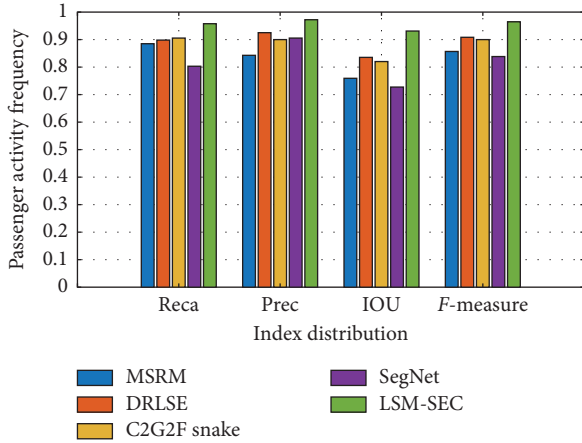


FIGURE 10: Comparison of tongue image segmentation results.

The variables  $A_a$  and  $B_b$  represent the segmentation results of the model and the divisions given by the medical experts. FP, FN, and TP are false positive volume fractions, false negative volume fractions, and true positive volume fractions, respectively, as defined below:

$$\begin{aligned}
 TP &= \frac{A_a \cap A_b}{A_b}, \\
 FN &= \frac{A_b - (A_a \cap A_b)}{A_b}, \\
 FP &= \frac{A_a - (A_a \cap A_b)}{A_b}.
 \end{aligned} \tag{24}$$

The closer the values of prec, reca, IoU, and  $F1$ -measure are to 1, the better the segmentation results are. The average reca of LSM-SEC is 95.6%, the prec is 97.2%, the IOU is 93%, and the  $F1$ -measure is 96.3%. Refer to Table 1 for index values of other methods.

Figure 10 is a comparison of the index results of the four methods. The abscissa is the mean value of the indicators of reca, prec, IOU, and  $F1$ -measure from left to right. Through the quantitative analysis of the four indicators, it can be concluded that the LSM-SEC algorithm represented by the yellow column is superior to the other methods in the segmentation effect of the tongue. The proposed method achieves accurate segmentation results on all clinical tongue images and has high robustness. In the MATLAB experimental environment, the average processing time of each image in this algorithm is about 49.2 seconds.

## 6. Conclusions

As aforementioned, tongue segmentation is an important basic step in the informatization of tongue diagnosis in traditional Chinese medicine. In this paper, we first introduce a symmetry and edge-constrained level set model, which combines the latest neural network model and level set segmentation method to improve the gradient accuracy. With the symmetry constraint and adjustment of the initialization position, the proposed approach realizes

intelligent segmentation. As the basic of the expert system, the symmetry and edge-constrained level set model for tongue segmentation can realize automatic tongue segmentation without manual intervention and achieve the goal of intellectualization. Finally, we provide detailed experimental tests. The experimental results demonstrate the segmentation accuracy and robustness of the proposed algorithm.

Machine learning has been widely used in the medical field and plays an important role in disease diagnosis. In assisted tongue diagnosis, the method based on deep learning can achieve end-to-end tongue segmentation, which greatly simplifies the tedious steps of the traditional segmentation method and runs faster, with higher accuracy and better robustness. But different learning methods have different efficiency and segmentation accuracy. Different training samples and different size data will also affect the segmentation accuracy. Therefore, in the future research, we can improve the segmentation effect by improving the network structure and training strategy. At the same time, in view of the small dataset of the tongue image, few-shot learning is also considered as the research direction in the future.

## Data Availability

In the experiment, the tongue image dataset contains 550 tongue images, part of which is from GitHub's open-source dataset, with a total of 300 tongue images: <https://github.com/BioHit/TongueImageDataset>; the other part is provided by the project collaborator of the University of Traditional Chinese Medicine, with a total of 250 tongue images. The images in the dataset are different in size, shape, angle, and position, but they all contain the complete tongue body, which is suitable for this experiment. Due to the need of the follow-up experiments, the tongue images were flipped, randomly cropped, rotated, and other operations were performed to expand the dataset, and finally, 1100 images were obtained. The "ground truth" of each tongue image is manually marked by experts in traditional Chinese medicine.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was helped by Yao Qin (Associate Research Librarian) of Shandong University of Traditional Chinese Medicine, and the authors thank her for providing them with some research data and professional guidance. This work was supported in part by National Natural Science Foundation of China (U1909210 and 61772309), Natural Science Foundation of Shandong Province (ZR2020MF037, ZR2019MF016, and ZR2019MF051), Key Research and Development Project of Shandong Province



(2019GGX101007), Planning Foundation of Education Ministry (20YJA870013), and Introduction and Education Plan of Young Creative Talents in Colleges and Universities of Shandong Province.

## References

- [1] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.
- [2] C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia, "A variational model for image classification and restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 460–472, 2000.
- [3] X. Song, M. Cheng, B. L. Wang, S. Huang, X. Huang, and J. Yang, "Adaptive fast marching method for automatic liver segmentation from CT images," *Medical Physics*, vol. 40, no. 9, pp. 897–900, 2013.
- [4] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2016.
- [5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2014.
- [6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [7] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1857–1866, Salt Lake City, UT, USA, June 2018.
- [8] P. A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: a deep neural network architecture for real-time semantic segmentation," 2016, <https://arxiv.org/abs/1606.02147>.
- [9] Y. Wang, Q. Zhou, J. Liu et al., "Lednet: a lightweight encoder-decoder network for real-time semantic segmentation," in *Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1860–1864, Taipei, Taiwan, September 2019.
- [10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [11] C. Li, C. Kao, J. C. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1940–1949, 2008.
- [12] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [13] M. A. Savelonas, D. K. Iakovidis, and D. Maroulis, "LBP-guided active contours," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1404–1415, 2008.
- [14] R. Gu, G. Wang, T. Song et al., "CA-net: comprehensive attention convolutional neural networks for explainable medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 699–711, 2021.
- [15] K. H. Kim, J. H. Do, H. Ryu, and J. Y. Kim, "Tongue diagnosis method for extraction of effective region and classification of tongue coating," in *Proceedings of the 2008 First Workshops on Image Processing Theory, Tools and Applications*, pp. 1–7, IEEE, Sousse, Tunisia, November 2008.
- [16] X. Zhai, H. Lu, and L. Zhang, "Application of image segmentation technique in tongue diagnosis," in *Proceedings of the 2009 International Forum on Information Technology and Applications*, pp. 768–771, IEEE, Chengdu, China, May 2009.
- [17] D. Zhang, B. Pang, N. Li, K. Wang, and H. Zhang, "Computerized diagnosis from tongue appearance using quantitative feature classification," *American Journal of Chinese Medicine*, vol. 33, no. 6, pp. 859–866, 2005.
- [18] M.-J. Shi, G.-Z. Li, F.-F. Li, and C. Xu, "Computerized tongue image segmentation via the double geo-vector flow," *Chinese Medicine*, vol. 9, no. 1, p. 7, 2014.
- [19] Z. P. Huang, Y. S. Huagn, F. L. Yi et al., "An automatic tongue segmentation algorithm based on OTSU and region growing," *LiShiZhen Medicine and Materia Medica Research*, vol. 28, no. 12, pp. 3062–3064, 2017, in Chinese.
- [20] W. Liu, C. Zhou, Z. Li, and Z. Hu, "Patch-driven tongue image segmentation using sparse representation," *IEEE Access*, vol. 8, pp. 41372–41383, 2020.
- [21] X. Huang, H. Zhang, L. Zhuo, X. Li, and J. Zhang, "TISNet-enhanced fully convolutional network with encoder-decoder structure for tongue image segmentation in traditional Chinese medicine," *Computational and Mathematical Methods in Medicine*, vol. 202013 pages, 2020.
- [22] P. L. Qu, H. Zhang, L. Zhuo, J. Zhang, and G. Chen, "Automatic tongue image segmentation for traditional Chinese medicine using deep neural network," in *Proceedings of the 13th International Conference on Intelligent Computing Theories and Application ICIC 2017*, pp. 247–259, Liverpool, UK, August 2017.
- [23] S. Yu, J. Yang, Y. Wang, and Y. Zhang, "Color active contour models based tongue segmentation in traditional Chinese medicine," in *Proceedings of the International Conference on Bioinformatics and Biomedical Engineering*, pp. 1065–1068, Wuhan, China, July 2007.
- [24] K. Wu and D. Zhang, "Robust tongue segmentation by fusing region-based and edge-based approaches," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8027–8038, 2015.
- [25] W. S. Li, J. F. Yao, L. Yuan, and Q. Zhou, "The segmentation of the body of tongue based on the improved level set in TCM," in *Proceedings of the International Conference on Life System Modeling and Simulation & LSMS 2010 and International Conference on Intelligent Computing for Sustainable Energy and Environment ICSEE 2010*, pp. 220–229, Wuxi, China, September 2010.
- [26] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.
- [27] C. Xu and J. L. Prince, "Gradient vector flow: a new external force for snakes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 66–71, San Juan, PR, USA, June 1997.
- [28] M. Shi, G. Li, and F. Li, "C2G2FSnake: automatic tongue image segmentation utilizing prior knowledge," *Science China Information Sciences*, vol. 56, no. 9, pp. 1–14, 2013.
- [29] M. Zhu, J. Du, and C. Ding, "A comparative study of contemporary color tongue image extraction methods based on HIS," *International Journal of Biomedical Imaging*, vol. 2014, no. 6, 10 pages, Article ID 534507, 2014.
- [30] X. Sun and C. Pang, "An improved Snake model method on tongue segmentation," *Journal of Changchun University of Science & Technology*, vol. 36, no. 5, pp. 154–156, 2013.

- [31] J. A. Sethian, "Curvature and the evolution of fronts," *Communications in Mathematical Physics*, vol. 101, no. 4, pp. 487–499, 1985.
- [32] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [33] C. Li, C. Xu, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Trans Image Process*, vol. 19, no. 12, pp. 3243–3254, 2010.
- [34] L. Zhong, Y.-f. Zhou, X.-f. Zhang, Q. Guo, and C.-m. Zhang, "Image segmentation by level set evolution with region consistency constraint," *Applied Mathematics-A Journal of Chinese Universities*, vol. 32, no. 4, pp. 422–442, 2017.
- [35] Y. Zhang and D. Ji, "Adaptive Harris corner detection algorithm based on B-spline function," in *Proceedings of the 2010 2nd International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, August 2010.
- [36] Y. Liu, M. M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Las Vegas, NV, USA, June 2016.
- [37] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.

## Research Article

# A Multiattention-Based Supervised Feature Selection Method for Multivariate Time Series

Li Cao <sup>1</sup>, Yanting Chen <sup>2</sup>, Zhiyang Zhang <sup>2</sup>, and Ning Gui <sup>2</sup>

<sup>1</sup>School of Information, Zhejiang Sci-Tech University, Hangzhou, China

<sup>2</sup>School of Computer Science and Engineering, Central South University, Changsha, China

Correspondence should be addressed to Ning Gui; ninggui@gmail.com

Received 24 May 2021; Revised 25 June 2021; Accepted 8 July 2021; Published 21 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Li Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feature selection is a known technique to preprocess the data before performing any data mining task. In multivariate time series (MTS) prediction, feature selection needs to find both the most related variables and their corresponding delays. Both aspects, to a certain extent, represent essential characteristics of system dynamics. However, the variable and delay selection for MTS is a challenging task when the system is nonlinear and noisy. In this paper, a multiattention-based supervised feature selection method is proposed. It translates the feature weight generation problem into a bidirectional attention generation problem with two parallel placed attention modules. The input 2D data are sliced into 1D data from two orthogonal directions, and each attention module generates attention weights from their respective dimensions. To facilitate the feature selection from the global perspective, we proposed a global weight generation method that calculates a dot product operation on the weight values of the two dimensions. To avoid the disturbance of attention weights due to noise and duplicated features, the final feature weight matrix is calculated based on the statistics of the entire training set. Experimental results show that this proposed method achieves the best performance on compared synthesized, small, medium, and practical industrial datasets, compared to several state-of-the-art baseline feature selection methods.

## 1. Introduction

With the development of IoT, more and more domains, e.g., social media and industries, have accumulated a large amount of high-dimensional data with temporal orders, so-called multivariate time series (MTS), which contain valuable information. MTS data containing a large number of features become more and more common in various applications, such as in biology [1], multimedia [2], social networks [3], energy [4], and industries [5, 6]. It has brought the curse of dimensionality and volume. Excessive numbers of features may greatly slow down the quality of the classifiers because irrelevant, redundant, and noninformative features are highly confusing in the learning process [7–9], while also increasing computational overhead. Thus, it is important to fully exploit the complex relationship from both temporal and variate dimensions and identify the most related variates and their most appropriate feature time stamps in respect to the supervision target. Figure 1 shows

the two different requirements for the feature selection in MTS. Finding those variables and their time lags is often of great importance in understanding physical/chemical models of the underlying systems.

Feature selection, by removing irrelevant and/or redundant features/variables, has been seen as an essential and crucial data preprocessing step for machine learning [10]. The supervised feature selection methods are normally categorized as the wrapper, filter, and embedded methods [7, 11]. Different feature selection algorithms exploit various types of criteria to define the relevance of features: similarity-based methods, e.g., SPEC [12] and Fisher's score [13], feature discriminative capability, e.g., ReliefF [14], information-theory based methods, e.g., mRmR [15], and statistics-based methods, e.g.,  $T$ -score [16]. However, those feature selection methods normally suffer major problems: varying from computation scalability to stability. Recently, advances in tree-based solutions and deep learning-based feature selection and many deep learning-based feature selection methods have

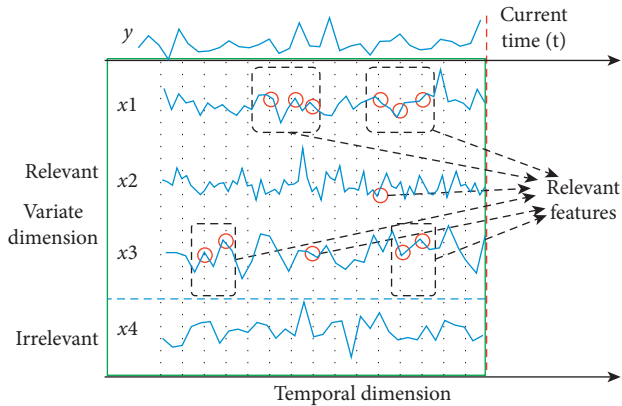


FIGURE 1: Two-dimensional feature selection in MTS: temporal feature selection and variate selection, only partial variates and certain time lags of those variates relevant towards the label  $y$ .

been proposed due to their effectiveness in processing massive data and rich modeling capability. Random Forest [17] calculates feature importance as the sum over number of splits. The extreme popularity of the gradient boosting methods also provides feature selection capabilities, e.g., the Xgboost [18] and LightGBM [19] calculate feature weight basically according to the numbers of times the feature is used. Li et al. [20] proposed a deep feature selection (DFS) by adding a sparse one-to-one linear layer. Roy et al. [21] use the activation potentials contributed by each of the individual input dimensions, as the metric for feature selection. Gui et al. [22] in their recent work use an attention mechanism for the general feature selection task as both attention mechanism and feature selection focus on selecting partial data from the high-dimensional dataset. However, those feature selection algorithms are designed for general data and treating the two-dimensional MTS data indiscriminately.

For MTS feature selection, partially due to its complexity, most research studies are optimized for certain domains, e.g., Wong et al. [23] propose the feature selection method based on the adaptive resonance theory for financial time series forecasting. Jimenez et al. [24] define a wrapper feature selection method based on multi-objective evolutionary algorithms for antibiotic resistance outbreak prediction. González-Vidal et al. [25] design a feature selection method for smart buildings. Those approaches generally limit in their respective domains and cannot easily be extended to other domains. Few feature selection methods have been proposed for general multivariate time series. Most of them have major limitations. For instance, Hido and Morimura [26] find the most appropriate time stamps for the whole set of variates. Some keep, e.g., Wong et al. 2012, the time windows invariant or the same for all features [23]. Sun et al. [27] used the Granger causality [28] discovery to identify causal features as well as the effective sliding window sizes in multivariate numerical time series. However, these approaches face the same limitation of Granger causality and may produce misleading results when the true relationship involves three or more variables and is incapable of the nonlinear causal relationship.

In this paper, a novel multiattention-based supervised feature selection (m-AFS) method is proposed to explicitly tackle the two different correlations. It translates the feature weight generation problem into a bidirectional attention generation problem with two parallel placed attention modules. The input 2D data are sliced into 1D data from two orthogonal directions, and each attention module generates attention weights from their respective dimensions.

The major contributions of our work are as listed as follows:

- (i) An innovative biattention-based feature selection architecture is proposed to make dimension-specific feature selection methods with neural network-based solutions. This method proposes a systematic structure to generate two different feature weights from a different perspective with one coherent neural network structure. By reusing existing neural network computation advances, this architecture supports fast and scalable feature weight generation.
- (ii) Two different attention-based modules are proposed that formulate dimension-specific feature weight generation problems into attention-based attention weight generation problems: attention over time (AoT) and attention over variates (AoV). Those two modules are designed according to the different characteristics of two-dimensional features.
- (iii) A feature weight generation mechanism is proposed to generate a final feature weight matrix to unify two different feature weights across two dimensions with simple dot product operation. As the attention weight might have a huge disturbance during the training, the final feature weight matrix is calculated based on the statistics of the entire training set.

A set of experiments are designed on a set of datasets including both regression and classification problems. The highest predicting and classification accuracy, compared with existing popularly used baseline algorithms, has been observed on all tested datasets. To the best of our knowledge, m-AFS is the first attention-based neural network solution for MTS feature selection tasks.

## 2. Multiattention-Based Feature Selection

In this section, the overall architecture of m-AFS is illustrated and analyzed. Then, the major components of this architecture are illustrated.

**2.1. Notation.** For the clarity of symbol usage, this paper presents matrices as a bold uppercase character (e.g.,  $A$ ), vectors as a bold lowercase (e.g.,  $a$ ), and normal lowercase character for numerical values (e.g.,  $a$ ). For instance, a time series is a series of observations,  $x_i(t)$ ;  $i = 1, 2, \dots, m$ ;  $t = 1, 2, \dots, d$ , which is made sequentially through time, where  $i$  denotes the index of the measurements made at each time step  $t$  and  $t$  denotes the index of the time. Matrix  $X = \{x_i(t_k) | i = 1, 2, \dots, m; k = 1, \dots, d\}$  is used to indicate the feature selection space with  $n$  features and

$d$  time points before time  $t$ . Here,  $d$  represents the maximum time interval in respect to the current time  $t$ . For the feature selection task, our goal is to find the appropriate feature and time step with respect to the output  $y(t)$ . Here,  $y(t)$  presents the value for the label at time point  $t$ . When  $n$  is equal to or greater than 2, it is called MTS.

**2.2. Architecture.** As discussed in Introduction, for MTS data, two different feature selection dimensions coexist: time dimension selection and variate dimension selection. Those two dimensions have respective characteristics and have to be handled differently. In the time dimension, the sequence of a single feature's correlation with the target at different time steps generally is of close characteristics: (1) same unit: the unit of value is uniform for the same feature; (2) continuity in values: the values in time sequence are generally continuous. Normally, the smaller the time interval, the smaller the difference between the front and back of the sequence of features. However, in the variate dimension, different features are heterogeneous in most cases. Therefore, the ways in which features are correlated with the label normally are quite different.

Similar to the embedded feature selection methods, m-AFS generates feature weight during a learning process. As shown in Figure 2, m-AFS consists of three connected modules, namely, the AoT module, the AoV module, and the learning module. The AoT and AoV modules are parallel arranged in the upper of m-AFS. AoT is responsible for computing the time dimensional weights with transformed one-dimensional data instead of the original data. Each variate has an AoT module and a set of attention weights  $a_T^i$  is generated. Similarly, the AoV takes all variates at the same time step as its inputs and tries to find the correlation between variates and label. The two attention modules are placed in parallel to avoid convergence problem which exists in the sequential structure. The mutual influence between two modules hampers the learning module. The learning module aims to find the optimal correlation between the weighted features and the supervision target by solving the optimization problem. It connects the supervision target and features by the backpropagation mechanism and continuously corrects the feature weights during the training process. The AoT, AoV, and the learning module build the correlation that best describes the degree of relevance of the target and features together.

As shown in Figure 2, m-AFS is a loosely coupled and stacked structure. Thus, it is quite similar to extend the feature selection to data with more dimensions, e.g., temporal, spatial, and variable dimensions. Furthermore, the learning module can also be customized according to specific learning tasks, e.g., CNN or RNN.

**2.3. Design of the Attention Module.** The AoV unit, as shown in Figure 3, slices the sample along the time dimension and uses the variate vector on a single time step  $t_j = \{x_1(j), x_2(j), \dots, x_m(j)\}$  as input. Firstly, a dense layer (denoted as  $E$ ) is used to extract the intrinsic relationship to eliminate certain noise or outliers. The introduced dense network  $E$  compresses the original feature domain into a vector with a smaller size (adjustable according to specific problems),

while keeping the major part of the information. As the size of  $E$  is normally much smaller than the size of variables, certain redundant variables will be discarded during this process.

Secondly, by using the extracted  $E$  as input, each  $U$  is assigned with a shallow neural network corresponding to the number of variables. The output of  $U$  represents the  $j$ th time step's variable attention distribution. To widen the difference between variables and avoid to take an effect on the time dimension, the softmax activation is used and the selection possibility of feature  $j$ ,  $p^j$  is calculated with equation (1) and the output  $a_V^j$  is calculated with (2):

$$p^j = w_p^j t_j + b_p^j, \quad (1)$$

$$a_V^j = \text{softmax}(\tanh(w_n^j p^j + b_n^j)). \quad (2)$$

For each input  $X$  with  $m$  feature and  $n$  time steps, the AoV modules generate  $n$  different attention vectors  $a_V^j$  for different time stamps  $j$ . Thus, it creates a weight matrix  $A_V = \{a_V^j | j = 1, 2, \dots, d\}$ . Note that the parameters of AoV and AoT modules are summarized as  $\theta_a$ .

While the AoV unit calculates the variable attention, the AoT unit integrates the input information of all moments in the form of soft attention. It uses the time step vector of a single variable  $x_i = \{x_i(1), x_i(2), \dots, x_i(d)\}$  as input and calculates the  $i$ th variable's corresponding attention vector  $a_T^i | i = 1, 2, \dots, m$  and matrix  $A_T = a_T^i | i = 1, 2, \dots, m$  with a series of transformations which are similar to the AoV unit. For each variable, one AoT is used.

This design has two major functions: (1) the separation of two dimensions avoids mutual influence and accelerates convergence and (2) each component  $a_T^i$  and  $a_V^j$  in the interval (0,1) can force many feature coefficients to be small, or exactly zero to facilitate feature selection. Attention here is similar to some sparse regularization terms used in many sparse-learning-based feature selection methods.

**2.4. Learning Module.** The feature weights generated from m-AFS are from AoT and AoV, respectively. Therefore, it is important to merge the two sets of weights to facilitate global feature selection. Two dimensions of the original data have their different characteristics and cannot directly be used for selection. But after the transformations of the attention module, weights of both dimensions are unified within [0,1] and can be directly used to identify the importance of variables. Here, we contact the two attention weight matrices  $A_V$  and  $A_T$  by a pairwise multiplication operation  $\odot$  and the global dimension attention weight is as follows:

$$A = A_V \odot A_T. \quad (3)$$

The 2D weighted inputs of the learning module  $G$  can be accessed by the following equation:

$$G = A \odot X. \quad (4)$$

$A$  is constantly adjusted during the learning process with backpropagation by solving the objection function as follows:

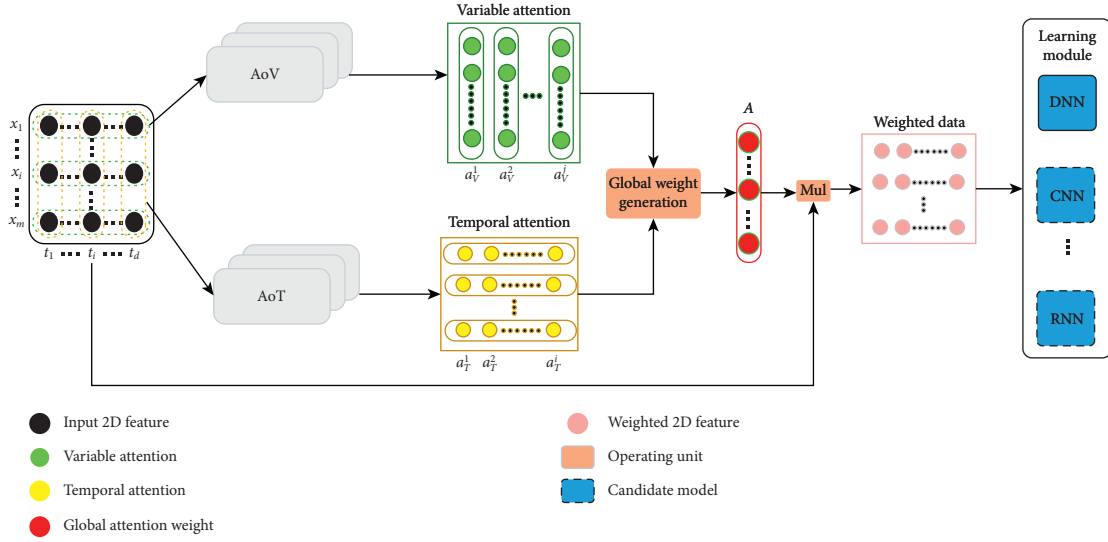


FIGURE 2: Conceptual structure of m-AFS.

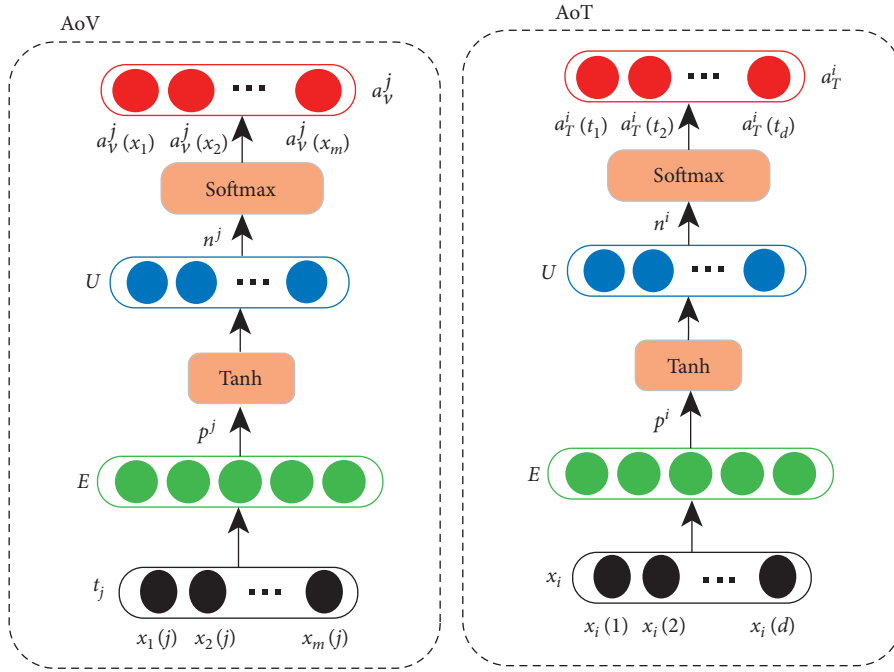


FIGURE 3: Attention structure for temporal dimension.

$$\arg \min_A \mathcal{L}[f_{\theta_l}(A \odot X) - Y] + \lambda R(\theta), \quad (5)$$

where  $\theta = \langle \theta_a, \theta_l \rangle$  and  $R(\cdot)$  is often an  $L2$ -norm that helps to speed up the optimization process and prevent overfitting. Here,  $\lambda$  controls the strength of regularization. The loss function depends on the type of prediction task. For the classification tasks, the cross-entropy loss functions are usually used. For regression tasks, the mean absolute error (MAE) is normally used. Note that  $f_{\theta_l}(\cdot)$  is a neural network with parameters  $\theta_l$ .

For a specific learning problem, m-AFS can use a network structure that best fits the particular task. For general

value-based regression and classification tasks, we adopt the fully connected network for task learning. Other structures, e.g., LSTM and CNN, are also adopted.

**2.5. Feature Score Generation.** Considering the much larger amount of data and limited computing resources in the real scenario, as well as the risk of trapping into local optimum, the training of network is processed in batch. This limits us to getting global attention weights of only one batch inputted, resulting in degraded performance. To have a better understanding of the attention distribution, we use the trained model to evaluate the whole dataset, get each

sample's global weight  $w_s$ , and calculate the statistical feature score using the following equation:

$$F = \frac{\sum_{i=1}^D A_i}{D}, \quad (6)$$

where  $D$  is the size of the dataset and  $A_i$  is the attention matrix generated by the trained model for the sample  $i$ . The average weight matrix  $F$  across the whole sample is used as the basis for the feature selection.

### 3. Results

In this section, we will conduct experiments to answer the following research questions:

- (i) Q1: Does the selection achieve good accuracy or a small error in those datasets?
- (ii) Q2: Does it capable to select the most appropriate features from both the temporal and variate dimensions?

In the following section, we introduce the basic experiment settings and the comparisons of different methods on both synthetic and real-world datasets.

**3.1. Experiment Settings.** This section is divided into two main experiments. The first experiment verifies the feasibility of m-AFS on a synthetic data. Then, experiments on several real-world datasets from the UCI Machine Learning Data Repository are conducted.

**3.1.1. Evaluation Setting.** The ratio of training data to test data is 8:2. m-AFS adopts the normalization method introduced in Section 2.5 to generate global feature weight from the weights of variable and the temporal dimensions. Other feature selection methods do not have the concept of hierarchically generating weights. Thus, other baseline algorithms select feature directly via their feature weights across all features.

**3.1.2. Baselines.** The implementation of the feature selection methods compared in this experiment is from the open-source library [7] (<https://github.com/jundongl/scikit-feature>). This experiment compares the m-AFS with the following representative methods:

Similarity-based methods: Fisher's score [29] and ReliefF [30] select features by finding the near-hit and near-miss instances using the l1-norm: FS\_l2l1 (feature selection with l2, l1-norm) [31]

Embedded method: RF (Random Forest) is a tree-based feature selection method provided by scikit-learn package

**3.1.3. Predictive Model Settings.** The RF (Random Forest) is used as the classifier for the experiments to avoid using the same methods for feature selection and testing. Other classifiers are also tested, e.g., support vector machine (SVM) is too slow to be used in the large dataset, and KNN is also much slower than RF and displays no significant advantages over RF

in most of the tested datasets. Since the feature subsets selected by different feature selection methods are different, it is not appropriate to use the same hyperparameters for prediction. Therefore, we use the grid search to find the optimal parameters for the prediction model and use these parameters to set the model and then test the prediction accuracy on the reconstructed feature set. For the regression tasks, the mean absolute error (MAE) is adopted while the percentage of classification accuracy is used for classification tasks.

Model parameters are initialized with the truncated normal distribution with a mean of 0 and a standard deviation of 0.1. The model is optimized by Adam. The batch size is set according to the size of samples, 100 for small datasets and 1000 for MNIST and noisy MNIST. The learning rate is the default value of Adam optimizer in Keras framework (0.002). Here, all trainable parameters are constrained by L2 regularization. The network setting of AoT is one hidden layer and AoV is with two hidden layers: the first layer  $E$  with 32 units and the second layer  $U$  with the length of time steps and the number of the variables, respectively. The  $E$  layer is with 512 units. As the structure is loosely coupled, the learning module can be easily replaced. The max training epoch is set at 100 and early stopping is adopted to avoid overfitting.

**3.2. Experiments on the Synthetic Data.** In order to verify whether m-ATP can accurately identify the related features, we performed feature selection in a synthesized nonlinear system with known dynamics. There are six variables that are uniformly random distributed. The output  $y$  is generated with the following function:

$$Y = X_1(t-1) * X_2(t-2) + X_3(t-5) + X_4(t-1) + X_4(t-4) + X_4(t-5) + X_4(t-7) + X_4(t-8) + \sigma(0, 0.1), \quad (7)$$

where  $x_1 \in [2, 5]$ ,  $x_2 \in [10, 30]$ ,  $x_3 \in [5, 10]$ ,  $x_4 \in [30, 70]$ ,  $x_5 \in [100, 200]$ ,  $x_6 \in [65, 85]$ , and uniformly distributed. As can be seen from this equation, only  $x_1 \sim x_4$  variates are related to  $y$  at certain time stamps. At the same time, in order to simulate the noisy environment, Gaussian white noise  $\sigma(0, 0.1)$  is added. The total number of samples of the simulation dataset generated according to the above principles is 5000. Here,  $T$  is set to 10, and the total number of samples becomes 4991.

We tested various feature selection algorithms on the datasets. Here, the major focus is to check whether those algorithms can effectively identify the correct time stamps. Thus, the feature weights generated by different methods are illustrated in Figure 4. Note that as other methods generate weight in ranges other than  $[0, 1]$ , in order to have straightforward comparisons, those weights are normalized to the same range. Of course, the order of feature weights for feature selection is kept unchanged. The darker the feature, the more likely it should be chosen. This figure clearly shows that m-AFS can correctly find all the most relevant time stamps. In contrast, none of the other methods can correctly identify both variates and time stamps, or even some of

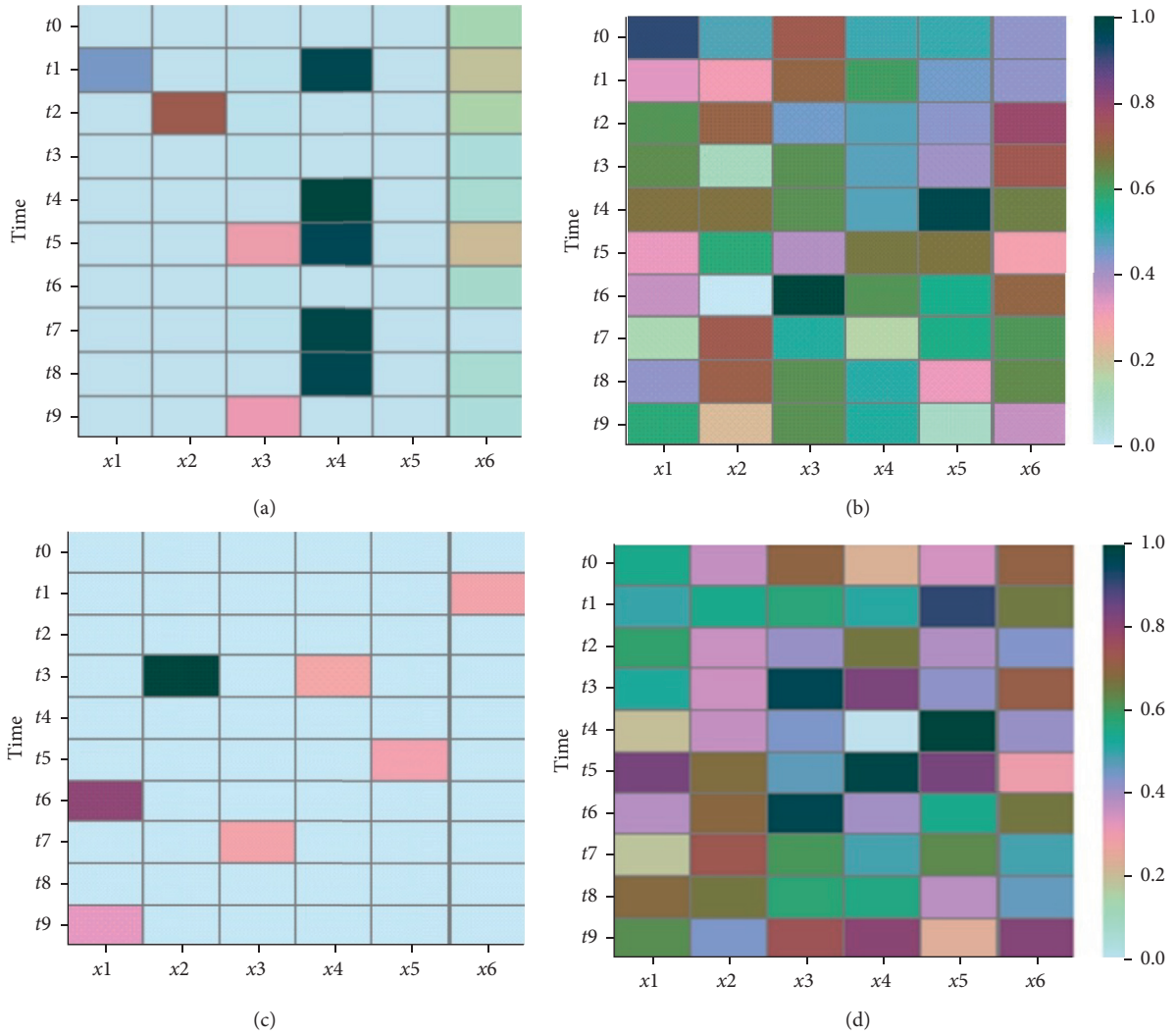


FIGURE 4: Feature weight distribution in the synthetic data. (a) m-AFS feature weight; (b) trace ratio; (c) RF; (d) ReliefF.

them. For instance, although RF achieves very sparse feature weight distribution, this distribution deviates significantly from the real system dynamics. Thus, their results might give misguidance towards the system’s characteristics.

**3.3. Experiments on Real-World Datasets.** To further demonstrate the effectiveness of m-AFS in real-world cases, we conducted experiments in six publicly available time series datasets from UCI (<https://archive.ics.uci.edu/ml/index.php>), including three regression datasets and three classification datasets. Details about the dataset are shown in Table 1. The size of the data is calculated with the product of sample instances, maximum time window, and the number of variates to represent how many inputs are needed to be calculated.

Table 2 shows the partial experiments results on the six different MTS datasets with different percentages of selected features. Due to the fact that MTS data normally have strong autocorrelation in the temporal dimension, maximum 15% of features are selected.

TABLE 1: Dataset information.

Dataset	Type	Var. no.	Win. size	Train/test	Size (million)
DC	R	7	20	1900/475	0.266
SRU	R	5	15	8053/2014	0.603
AEP	R	27	20	15772/3944	8.516
EEG	C	14	20	11968/2993	3.351
OD	C	5	20	6499/1625	0.650
WFRN	C	24	20	4349/1088	2.328

Table 2 shows that m-AFS and RF achieve the best performance on almost all the datasets and normally have big performance advantage over the other methods. RF leads with small percentage over m-AFS in the top 5% range and m-AFS ranks first in most top 10% features. However, their performances are quite close. It shows that both methods can identify the most influential factors for the prediction. Other methods, e.g., the LS\_121 have rather unstable performance in different datasets. LS\_121 ranks first in the OD dataset while it ranks last in the EEG dataset. Both datasets are the classification task. We also notice that more selected features normally yield little improvement towards the final results.



TABLE 2: Regression and classification accuracy with different percentages of selected features with the RF classifier.

	SRU ( $10^{-2}$ )	DC ( $10^{-2}$ )	AEP ( $10^{-2}$ )	EEG (%)	OD (%)	WFRN (%)
Top 5% of selected features						
m-AFS	1.65	6.55	<b>3.09</b>	<b>89.98</b>	95.38	92.74
Fisher's score	2.63	11.24	3.72	81.35	98.52	93.47
ReliefF	2.74	9.39	5.53	70.09	97.48	94.30
Trace	2.80	5.94	4.05	68.09	84.74	90.71
LS_l21	1.86	6.81	3.36	63.81	<b>99.01</b>	92.46
RF	<b>1.40</b>	<b>5.37</b>	3.29	82.15	98.58	<b>98.34</b>
Top 10% of selected features						
m-AFS	1.3	<b>3.18</b>	<b>3.22</b>	<b>95.16</b>	<b>99.26</b>	95.96
Fisher's score	2.55	7.31	3.38	85.63	98.65	96.04
ReliefF	2.52	7.73	5.29	76.54	98.46	95.31
Trace	2.58	5.62	3.44	75.67	86.77	92.09
LS_l21	1.57	5.24	3.23	67.65	98.52	93.29
RF	<b>1.10</b>	3.88	3.61	85.36	99.20	<b>98.07</b>
Top 15% of selected features						
m-AFS	0.99	<b>2.80</b>	3.27	<b>96.26</b>	99.26	95.96
Fisher's score	2.52	7.03	3.30	89.31	98.71	96.87
ReliefF	2.51	6.14	4.73	82.73	98.77	95.59
Trace	2.54	5.40	<b>3.15</b>	79.89	86.95	92.56
LS_l21	1.50	4.56	3.20	75.01	98.52	93.38
RF	<b>0.96</b>	2.81	3.60	86.97	<b>99.32</b>	<b>98.07</b>

And in the bigger range of top  $K$ , similar results are observed.

Here, the Random Forest algorithm is chosen also as the classifier for prediction and classification due to its performance and accuracy. We have to admit that this choice gives RF some advantages over the other methods. However, SVM is too slow to finish those tasks and KNN displays not so well accuracy in those tasks.

**3.4. Interpretability.** For many mission-critical domains, it is important that the generated feature weights have good interpretability and represent real system dynamics. Partial feature weights from the best two methods: m-AFS and RF for  $x_2$ ,  $x_3$ , and  $x_4$  of the SRU dataset are shown in Figure 5. It clearly shows that m-AFS generates more smooth feature weights and clearly identifies the system lags for variates  $x_2$  (around 5 7),  $x_3$  (around 14), and  $x_4$  (around 8 10). This result is quite close to the results deduced by domain expert supported with domain-specific data mining solutions [32]. Their conclusion is  $x_2$  (6),  $x_3$  (14), and  $x_4$  (10). However, results from RF hardly demonstrate this conclusion although it has the best performance in SRU.

These results also show the possibility that the global weight generation methods proposed have room for improvements. How to generate global consistent weights to facilitate the feature selection with two different dimension-specific weights still needs further investigations.

**3.5. Computational Complexity.** In Table 3, the computation overheads of different feature selection methods are illustrated. Note that AFS intentionally only uses the CPU rather than the

GPU as the calculation devices to make a fair comparison. Theoretically, it can execute 3~9 times faster on the GPU.

The overhead is measured with the execution time for the feature weight generation process. Results show that AFS has moderate computation complexity. For the training with 1000 steps, it takes about 10 s to 173 s for the feature weight generation. Its execution time increases almost linearly as the size of data increases. In contrast, Fisher's score and ReliefF suffer the high and unstable computation cost. Their calculation time does not increase exactly with the increase in data volume.

### 3.6. Discussions

**3.6.1. Possible Applications.** Obtaining the most relevant features of the target system and the time node with the greatest impact is essential for the modeling of any sequential system. As machine learning is more and more applied to the modeling of time series systems, the accuracy of the model is getting higher and higher, and the required parameters are becoming more and more complicated. The improvement of the accuracy of the model is of course very important, but the increase in the complexity of the model leads to a decrease in the intelligibility and robustness of the model. For many application scenarios that require high model availability and robustness, such as modeling of industrial systems, the existing deep learning models often cannot meet the modeling requirements of intelligibility and robustness. In our work, by identifying the most relevant features, the most relevant time delays, and the important system parameters and through the actual industrial data, the delay calculation of this SRU dataset is consistent with

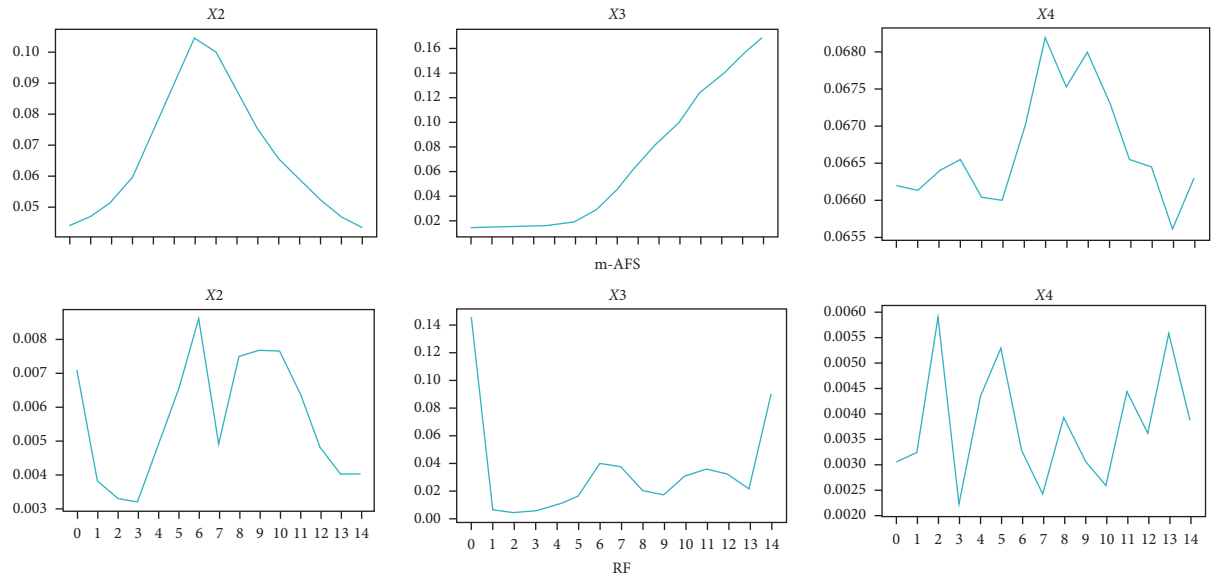


FIGURE 5: Feature weight distribution in the SRU dataset.

TABLE 3: Comparisons of the computation overhead (in seconds).

Meth. dataset	DC	SRU	EEG	AEP	OD	WFRN
m-AFS	10	52	101	173	44	60
Fisher's score	16	1511	68	128	21	5.6
ReliefF	633	45594	412	2707	72	45
Trace ratio	1.3	30	95	185	19.2	10
LS_L21	1	1.4	12	20	3.5	3.3
RF	3	9	33	124	1.5	7.8

the actual physical model, which effectively illustrates that this work plays an important role in the modeling of understandable industrial systems.

**3.6.2. Current Limitations.** The current major limitation is in the difference of feature weight evaluation. Traditional feature selection solutions calculate the feature weights and select the most influential features from the global perspective. In contrast, m-AFS calculates the feature weight from two different dimensions. Although our solution provides better interpretability, it introduces complexities in evaluating their contributions in the global aspect. And we need to balance the attention weight from multiple dimensions as proposed in Section 2.5. We are working on a more effective solution to condense weights from multiple dimensions.

## 4. Conclusion

In this paper, a novel multiattention-based feature selection architecture is introduced for the supervised feature selection for MTS data. In this architecture, two different attention mechanisms are designed to make the temporal and variable selection according to different feature selection patterns. Specifically, for the temporal dimension, the feature weight problem is formulated into a weighted average problem. For the variate dimension,

the variate selection problem is transformed into a binary classification problem for each variate. This architecture is designed to be easily stackable so it is possible to be extended to data with more than two dimensions. Experiment results show that m-AFS can achieve the best feature selection accuracy on most tested different datasets, compared with three off-the-shelf and widely used baselines.

In future work, we aim to develop more domain-optimized solutions for data with more than 3 dimensions. We are also working on the data-driven physical dynamics model reconstruction to enhance the model interpretability.

## Abbreviations

MTS:	Multivariate time series
m-AFS:	Multiattention-based supervised feature selection
AoT:	Attention over time
AoV:	Attention over variates
Conv:	Convolution
FC:	Fully connected.

## Data Availability

All data can be found at <https://archive.ics.uci.edu/ml/index.php>.

## Disclosure

The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

Ning Gui and Li Cao conceptualized the study; Li Cao was responsible for methodology; Li Cao and YanTing Chen were responsible for software; YanTing Chen and ZhiYang Zhang validated the data; YanTing Chen performed formal analysis. All authors have read and agreed to the published version of the manuscript.

## Acknowledgments

This work was funded by the National Natural Science Foundation of China (no. 61772473).

## References

- [1] M. Radovic, M. Ghalwash, N. Filipovic, and Z. Obradovic, "Minimum redundancy maximum relevance feature selection approach for temporal gene expression data," *BMC Bioinformatics*, vol. 18, no. 1, p. 9, 2017.
- [2] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, vol. 31, San Francisco, CA, USA, February 2017.
- [3] J. Li, J. Tang, Y. Wang, Y. Wan, Y. Chang, and H. Liu, "Understanding and predicting delay in reciprocal relations," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1643–1652, Lyon, France, April 2018.
- [4] A. Yang, W. Li, and X. Yang, "Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines," *Knowledge-Based Systems*, vol. 163, pp. 159–173, 2019.
- [5] Y. Liang, D. Niu, and W.-C. Hong, "Short term load forecasting based on feature extraction and improved general regression neural network model," *Energy*, vol. 166, pp. 653–663, 2019.
- [6] X. Na, M. Han, W. Ren, and K. Zhong, "Modified BBO-based multivariate time-series prediction system with feature subset selection and model parameter optimization," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–11, 2020.
- [7] J. Li, K. Cheng, S. Wang et al., "Feature Selection: A Data Perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2018.
- [8] J. Ircio, A. Lojo, U. Mori, and J. A. Lozano, "Mutual information based feature subset selection in multivariate time series classification," *Pattern Recognition*, vol. 108, Article ID 107525, 2020.
- [9] T. Niu, J. Wang, H. Lu, W. Yang, and P. Du, "Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting," *Expert Systems with Applications*, vol. 148, Article ID 113237, 2020.
- [10] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowledge-Based Systems*, vol. 86, pp. 33–45, 2015.
- [11] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet, "Spectral dimensionality reduction," in *Feature Extraction: Foundations and Applications*, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds., pp. 519–550, Studies in Fuzziness and Soft Computing, 2006.
- [12] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 1151–1157, Sydney, Australia, 2017.
- [13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley and Sons, New York, NY, USA, 2012.
- [14] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, no. 1/2, pp. 23–69, 2003.
- [15] H. Hanchuan Peng, F. Fuhui Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [16] J. C. Davis and R. J. Sampson, *Statistics and Data Analysis in Geology*, Vol. 646, Wiley, New York, 1986.
- [17] A. Liaw and M. Wiener, *Classification and Regression by randomForest*, vol. 2, p. 5, 2002.
- [18] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016*, pp. 785–794, arXiv: 1603.02754, San Francisco, CA, USA, August 2016.
- [19] G. Ke, Q. Meng, T. Finley et al., "LightGBM: a highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [20] Y. Li, C.-Y. Chen, and W. W. Wasserman, "Deep feature selection: theory and application to identify enhancers and promoters," in *Research in Computational Molecular Biology*, T. M. Przytycka, Ed., pp. 205–217, Lecture Notes in Computer Science, 2015.
- [21] D. Roy, K. S. R. Murty, and C. K. Mohan, "Feature selection using deep neural networks," in *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, Killarney, Ireland, July 2015.
- [22] N. Gui, D. Ge, and Z. Hu, "AFS: an attention-based mechanism for supervised feature selection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3705–3713, 2019, Number: 01.
- [23] C. Wong and M. Versace, "CARTMAP: a neural network method for automated feature selection in financial time series forecasting," *Neural Computing and Applications*, vol. 21, no. 5, pp. 969–977, 2012.
- [24] F. Jiménez, J. Palma, G. Sánchez, D. Marín, M. D. Francisco Palacios, and M. D. Lucía López, "Feature selection based multivariate time series forecasting: an application to antibiotic resistance outbreaks prediction," *Artificial Intelligence in Medicine*, vol. 104, Article ID 101818, 2020.
- [25] A. González-Vidal, F. Jiménez, and A. F. Gómez-Skarmeta, "A methodology for energy multivariate time series forecasting in smart buildings based on feature selection," *Energy and Buildings*, vol. 196, pp. 71–82, 2019.
- [26] S. Hido and T. Morimura, "Temporal feature selection for time-series prediction," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Tsukuba, Japan, November 2012.
- [27] Y. Sun, J. Li, J. Liu, C. Chow, B. Sun, and R. Wang, "Using causal discovery for feature selection in multivariate numerical time series," *Machine Learning*, vol. 101, no. 1-3, pp. 377–395, 2015.
- [28] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, Wiley, Econometric Society, vol. 37, no. 3, pp. 424–438, 1969.
- [29] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Advances in Neural Information Processing Systems*, vol. 18, pp. 507–514, 2005.

- [30] I. Kononenko, “Estimating attributes: analysis and extensions of RELIEF,” in *Machine Learning: ECML-94*, F. Bergadano and L. De Raedt, Eds., pp. 171–182, Lecture Notes in Computer Science, 1994.
- [31] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient  $l_2, l_1$ -norm minimization,” <http://arxiv.org/abs/1205.2631> 2012.
- [32] S. Han, T. Kim, D. Kim, Y.-L. Park, and S. Jo, “Use of deep learning for characterization of microfluidic soft sensors,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 873–880, 2018.

## Research Article

# A Single Target Grasp Detection Network Based on Convolutional Neural Network

Longzhi Zhang  and Dongmei Wu 

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Dongmei Wu; wdm@hit.edu.cn

Received 21 May 2021; Accepted 10 July 2021; Published 20 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Longzhi Zhang and Dongmei Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grasp detection based on convolutional neural network has gained some achievements. However, overfitting of multilayer convolutional neural network still exists and leads to poor detection precision. To acquire high detection accuracy, a single target grasp detection network that generalizes the fitting of angle and position, based on the convolution neural network, is put forward here. The proposed network regards the image as input and grasping parameters including angle and position as output, with the detection manner of end-to-end. Particularly, preprocessing dataset is to achieve the full coverage to input of model and transfer learning is to avoid overfitting of network. Importantly, a series of experimental results indicate that, for single object grasping, our network has good detection results and high accuracy, which proves that the proposed network has strong generalization in direction and category.

## 1. Introduction

Over recent years, deep learning has gained huge breakthroughs in computer vision [1, 2]. Unlike traditional hand-engineered features, deep learning can autonomically learn features from images, to acquire highly abstract and robust visual features via making use of image information to the most extent. Naturally, as one of the most representative deep learning models, convolutional neural network has become a research hotspot in computer vision, with easy training, high performance, few parameters, and strong generalization. Particularly, researchers have attempted to introduce it into research on robotic grasp detection, since its remarkable achievements in target detection [3–12].

Literature [13] innovatively used convolutional neural network for robotic grasps. More importantly, a deep neural network with four layers was proposed, which could effectively express multimodal features of grasping position, to achieve accurate detection of suitable grasping position on object [13]. Furthermore, a three-stage convolutional neural network was adopted to detect the grasping position of

objects in depth image [14], where the first-level convolutional neural network was used for performing preliminary location of grasping position, the second-level convolutional neural network was utilized for acquiring the preselected grasping boundary, and the third-level convolutional neural network was to reevaluate the preselected grasping boundary. To perform operations, a two-step robotic grasp detection system was proposed [15].

Distinct from above thoughts, although convolutional neural network was also adopted to identify the grasping region of the object, the entire image of the object was taken as the input of network, to directly generate the position of the possible grasping region on object [16]. Reference [17] evaluated the possible position to be grasped of the target via predicting the grasping function learned from the convolutional neural network. In addition, researchers converted the grasp detection into an 18-channel binary classification [18] and adopted a convolutional neural network to learn the clamping rule of the two-finger gripper to obtain the optimal grasping position on the target. Xia et al. proposed a planar grasping pose detection method of the robot based on the cascaded convolutional neural network

[19]; they established a cascaded two-stage convolution neural network model with position and attitude from coarse to fine to estimate the optimal grasping position and angle. In order to perform grasping new unknown model objects, visual feature points of an object in the process of being grasped were extracted via a convolutional neural network model, and a grasp strategy was constructed based on these visual feature points [20].

For actual robotic grasping, some grasp detection methods based on convolutional neural network models were put forward. Literature [21] proposed a hybrid deep architecture combining visual and tactile sensing for robotic grasp detection. An efficient framework of hierarchical cascaded forests to perform recognition and grasp detection of objects from RGB-D images of real scenes was proposed [22]. Ribeiro et al. [23] addressed the problems of grasp detection and visual servoing using deep learning and applied them as an approach to the problem of grasping dynamic objects. To acquire satisfactory grasp detection results, a self-supervised learning method was applied to learn grasping data directly collected by a robot [24]. To recognize and detect grasp rectangles on images of an object to be held by two-plates parallel grippers, a dictionary learning and sparse representation framework was proposed [25]. Also, unsupervised feature-learning methods were proposed for grasp detection [26–28]. In literature [26], a network model was proposed for predicting the 6 DOF pose of the target to confirm the position to be grasped. A beneficial attempt was conducted via using tactile sensors and an unsupervised feature-learning approach to predict whether a grasp is successful [27]. To clean water surface by aquatic robots, researchers came up with an unsupervised grasp detection method for water-surface object collection [28].

Additionally, for actual robotic grasping, another category of prediction approach is based on reinforcement learning. Zhang et al. proposed a reinforcement learning method for grasp detection to define a grasp as a point in a 2D image plane [29] via Q network [30] to perform target reaching after training in simulation. In literature [31], an asynchronous deep reinforcement learning approach was presented for learning robotic grasping policies, which can be trained on real physical robots. To perform complex sequences of pushing and grasping on a real robot, a method that combines deep reinforcement learning with affordance-based manipulation was put forward for detecting grasps [32]. Furthermore, to improve the flexibility of robotic detection for grasps, a curriculum-based reinforcement learning approach was conducted to learn reactive policies for the task of real picking [33]. Obviously, unlike above methods, grasp detection based on reinforcement learning mainly focuses on learning grabbing strategy for detecting grasps, rather than involving the network architecture itself.

However, with some success of grasp detection based on convolutional neural network in theories and applications, for grasp detection network inheritance itself, overfitting in multilayer convolutional neural network still exists and leads to poor detection precision. To achieve highly accurate detection for grasps, a single target grasp detection network with high detection accuracy is proposed, which generalizes the fitting of angle and position.

The remainder of this paper is organized as follows. Section 2 introduces our preliminary work to provide a theoretical basis for this research. Section 3 gives an exhaustive formulation of our thoughts. Experimental results are shown in Section 4 to demonstrate the superiority of the proposed network. Ultimately, Section 5 concludes the paper and looks forward to the future work.

## 2. Related Work

*2.1. Overview and Analysis of Components in Convolutional Neural Network.* Objective of exploring each component in convolutional neural network is to deepen the understanding of network structure, so as to carry out our research. As a matter of fact, convolutional neural network is a feed-forward neural network, but distinct from ordinary neural networks, it is generally composed of a convolution layer, activation layer, pooling layer, and fully connected layer. Following, each component is overviewed and analyzed.

*2.1.1. Convolutional Layer.* Convolutional layer is the core module in a convolutional neural network, which is usually composed of several convolution kernels with different sizes. After image input into the convolutional neural network, the convolution kernel performs convolution operations successively on the width and height of the image with a certain step length, to obtain a convolved feature vector.

Unlike connection ways of neurons in the ordinary neural network, convolution operation adopts sparse connection, which means that only neurons calculated with convolution kernels are connected to each other. Thus, this connection mode could increase the sparsity of the network to greatly reduce the number of network parameters and also could avoid overfitting of the network. In addition, convolutional neural network has the characteristic of weight sharing; that is, different positions of an image could be processed via the same convolution kernel, which could also reduce the number of network parameters.

Furthermore, the relation between input and output in a convolutional neural network is determined by convolution operation and selection of hyperparameters.

Assuming that the input image size is  $H \times W \times C$ , the convolution kernel size is  $F \times F \times C$ , the number is  $N$ , the convolution step is  $S$ , the unilateral filling size is  $P$ , and the

output eigenvector is  $H \times W \times C$ , then the output could be expressed as

$$\begin{cases} H' = \frac{H - F + 2P}{S} + 1, \\ W' = \frac{W - F + 2P}{S} + 1, \\ C' = N. \end{cases} \quad (1)$$

Apparently, the output height and width of the convolutional neural network are determined by input, convolution kernel size, filling size, and step, while the output channel number is determined by convolution kernel number.

**2.1.2. Activation Function.** Activation function plays an important role in the convolutional neural network. In fact, the inexistence of activation function will lead to the output that is a linear expression of input, which means that the network could only deal with linear problems, thereby greatly weakening the expression ability of the network model. As a result, to increase the nonlinear expression ability of network, activation function is usually added after convolutional layer.

Sigmoid function is one of the typical activation functions [34], and its expression is

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (2)$$

In Sigmoid activation function, definition domain is  $(-\infty, +\infty)$ , and value ranges at  $(0, 1)$ , as shown in Figure 1.

Sigmoid function was formerly widely used in the shallow neural network, but when the input is large, its gradient approaches 0, and with the increasing depth of the network, gradient dissipation is easy to occur in backpropagation, leading to failure of network training. Moreover, the output value of the Sigmoid function is not centered at 0.

Another typical activation function is the tanh function [35], which could be expressed as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (3)$$

Similar to the Sigmoid function, the definition domain of the tanh function is  $(-\infty, +\infty)$ , and the value also ranges at  $(-1, 1)$ . However, different from the Sigmoid function, the output value of the tanh function is centered at 0, as shown in Figure 2.

Although the output value of the tanh function is centered at 0, it still has not solved the problem that the network could not effectively backpropagate in case output or initial value is large. Hence, applications of above two activation functions tend to drop off.

Subsequently, a linear rectifier function called ReLU was proposed [36]; the expression is

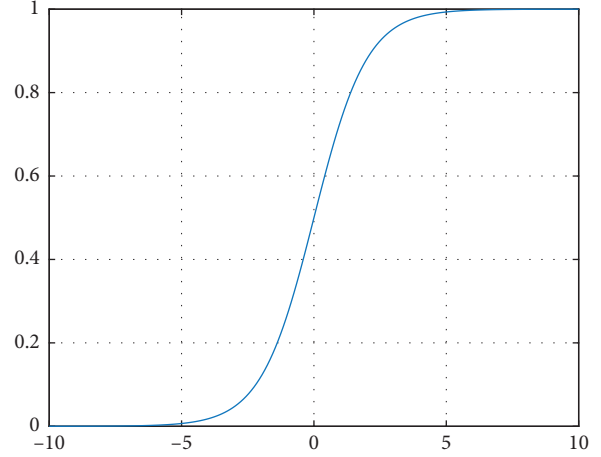


FIGURE 1: Sigmoid activation function.

$$f(x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0. \end{cases} \quad (4)$$

ReLU function is simple and easy to derive, which does not increase the difficulty in process of backpropagation and greatly accelerates the training speed. Even though the function could not be differentiated at 0, it has left derivatives and right derivatives around 0 and any of them could be selected since values exactly falling at 0 are minor and hardly affect the overall results. The image of this activation function is shown in Figure 3.

In ReLU activation function, the gradient saturation phenomenon is inexistence and the gradient is always 1, leading to fast convergence. Simultaneously, there is low computation due to nonexponential operations. Furthermore, neurons with output less than 0 do not work, which greatly increases the sparse expression ability of the network, to improve the network generalization performance. Thus, ReLU activation function is most widely used in current deep neural networks.

**2.1.3. Pooling Layer.** Pooling layer is also called downsampling layer and is commonly located behind the convolutional layer to reduce parameters number and computational complexity. Meanwhile, pooling layer could compress eigenvectors to exact main features and avoid overfitting. Generally, pooling layer could compress the sizes of eigenvectors but could not change their depth.

Typical pooling methods include average pooling and maximum pooling; their calculation principles are, respectively, shown in Figures 4 and 5.

In Figures 4 and 5, the size of convolution kernels is the same, since above convolution kernel size is the most universally used in the convolutional neural network. It can be clearly seen that the average pooling takes the average value of convolution kernel region size and blurs the eigenvectors; thus, it is not conducive to feature extraction. However, maximum pooling takes the maximum value of convolution kernel region size and retains remarkable features. Accordingly, maximum pooling is mostly used at present.

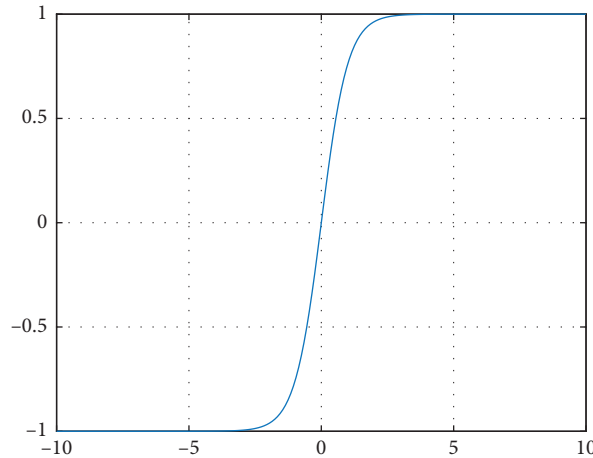


FIGURE 2: Tanh activation function.

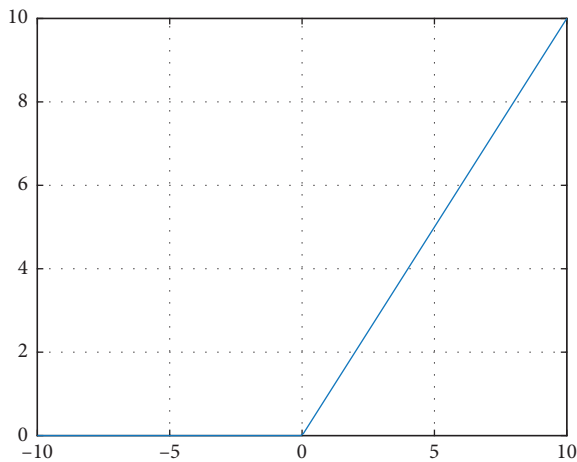


FIGURE 3: ReLU activation function.

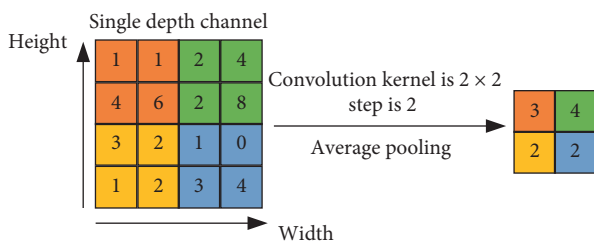


FIGURE 4: Calculation principle of average pooling.

**2.1.4. Fully Connected Layer.** Fully connected layer is similar to the ordinary neural network, without weight sharing and sparse connection of convolutional layer, and each neuron in it is interconnected. In a convolutional neural network, the input of the fully connected layer is eigenvectors extracted from the convolutional layer, and the output layer is selected based on completed task, such as Softmax output layer and logistic regression layer.

However, fully connection gives rise to a large number of parameters. If the number of data is too small, the network will easily fall into overfitting. Thus, in convolutional neural network, the emergence of the fully

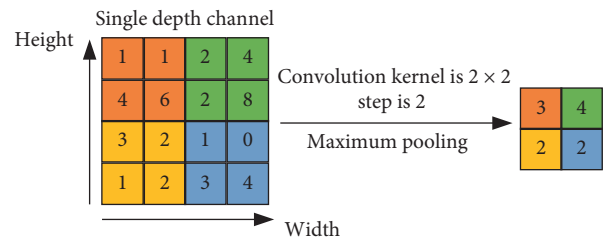


FIGURE 5: Calculation principle of maximum pooling.

connected layer is generally accompanied by the dropout layer. The dropout layer could stochastically discard some neurons to make them ineffective in fully connected layer. That is, the dropout layer is to imitate the sparse connection of the convolutional layer to prevent the overfitting of the network. In fact, the coefficient of dropout is confirmed by the specific application scenarios and network models, whose value is usually between 0.5 and 0.8 during training.

**2.2. Performance Comparison of End-to-End Target Detection Algorithms Based on Convolutional Neural Network.** Among target detection based on convolutional neural network, end-to-end networks directly detect the results from the image output, leading to a good performance in real time. Accordingly, we compare and analyze the performance of nowadays commonly used end-to-end networks to provide a theoretical foundation for our research.

In our implementation, VOC07 + 12 dataset is divided into a training set and test set, where the test set is 2007 test set, and the rest are training set. Detection results of different algorithms on test set are shown in Table 1.

It can be concluded from Table 1 that YOLOv2 is superior to YOLOv1 in accuracy and real time, and compared with YOLOv2-tiny, YOLOv2 gets a significant increase in accuracy at the expense of certain speed. Moreover, YOLOv2 is lower than SSD-300 in accuracy only at 1 percent, but more than four times faster in real time. Compared with



TABLE 1: Detection results of different end-to-end algorithms.

Algorithm	Training set	Test set	mAP	FPS
YOLOv1	VOC07 + 12 trainval	VOC07 test	48.1	71
YOLOv1-tiny	VOC07 + 12 trainval	VOC07 test	33.5	282
YOLOv2	VOC07 + 12 trainval	VOC07 test	75.4	100
YOLOv2-tiny	VOC07 + 12 trainval	VOC07 test	41.1	250
SSD-300	VOC07 + 12 trainval	VOC07 test	75.5	18
SSD-512	VOC07 + 12 trainval	VOC07 test	79.0	15

SSD-512, YOLOv2 is 6.7 times faster than it, while being lower than it in accuracy only at 3.6%.

Through comparative analysis of above results, it can be seen that YOLOv2 has superiority over others in high accuracy and better real time. Hence, this paper introduces it into research on grasp detection and makes use of its end-to-end detection thought to conceive a single grasp detection network, which takes an image as input and grasp parameters as output. Also, the proposed network has a great generalization ability to fit in angle and position and has high detection accuracy.

### 3. Constructing Single Target Grasp Detection Network

*3.1. Modeling Grasping Parameters.* Indeed, the essence of grasp detection based on a convolutional neural network is to find grasping parameters that could achieve stable grasping. Hence, establishment of an appropriate grasping parameter model to achieve stable grasping is the key to the research of grasp detection based on a convolutional neural network.

Saxena et al. adopted a 2D grasping point as a grasp parameter model [37], and Le et al. utilized a pair of grasping points as a grasp parameter model [38]. However, the limitation of above grasp parameter models lies in that they could not fully represent the seven dimension parameters in grasping operation of the robot, and the other parameters need to be estimated separately.

Due to this, Jiang et al. proposed a seven-dimensional representation method combining 2D grasping rectangle and 3D point cloud [39], which described the 3D position, attitude, and size of the end-gripper. However, 3D point cloud data need to be calculated, which means that the extracted point cloud data require high precision and large amounts of computation.

To deal with above problem, Redmon and Angelova [16] simplified the model of literature [39]. Their contribution simplified the grasping in three-dimensional into planar grasping and proposed a five-dimensional parameter representation method based on the 2D grasping rectangle, which brought inspiration to our research.

Obviously, simplifying 3D grasping into 2D planar grasping and using a grasping rectangle to express the grasp parameters could effectively reduce the computation, and the issue of grasp detection becomes relatively simple. Particularly, the grasping rectangle is used to describe the grasp parameters, which makes grasp detection quite similar to object detection, while the distinction between the two is

that the direction of the gripper needs to be considered in grasp detection.

Consequently, we utilize the strong learning ability of the convolutional neural network on image features to convert the grasp detection of the robot into target detection and adopt a 2D grasping rectangle to confirm the appropriate grasp parameters. More importantly, in order to enable 2D grasping to be fully mapped into 3D space and directly utilized by the robot to accomplish grasping operations, in this work, we assume that the gripper is always perpendicular to the  $z$ -axis to grasp vertically downward.

To sum up, we build up a model of grasp parameters with the manner of five-dimensional representation. More precisely, we use the position of the gripper  $(x, y)$ , the direction of gripper  $\theta$ , the opening size of the gripper before grasping objects  $w$ , and the size of gripper  $h$  to constitute a grasping rectangle, as exhibited in Figure 6.

The grasp parameters model could be expressed as

$$M = \{x, y, h, w, \theta\}, \quad (5)$$

where  $(x, y)$  is the center coordinates of grasping rectangle,  $\theta$  represents the rotation angle of grasping rectangle relative to the horizontal axis of the image (counterclockwise is positive),  $w$  means the width of grasping rectangle, and  $h$  refers to the height of grasping rectangle.

As displayed in Figure 6, a grasping rectangle of a remote device is composed of five grasp parameters defined by formula (5), where blue is on behalf of the gripper, red represents the distance between the two ends of the gripper before grasping,  $(x, y)$  is the center coordinates of grasping rectangle, and  $\theta$  represents the rotation angle of grasping rectangle relative to the horizontal axis.

*3.2. Modeling Grasp Detection Network.* As mentioned above, YOLOv2 has obvious advantages in detection accuracy and real time. Thus, we introduce it into the research of grasp detection and utilize its “end-to-end” detection manner to establish a grasp detection network model with the proposed 5 grasp parameters as output. Accordingly, it is necessary to comprehend and analyze the network structure of YOLOv2 before modeling the grasp detection network.

Darknet19 as the framework of YOLOv2 is composed of 19 convolutional layers and 5 maximum pooling layers. In darknet19, largely  $3 \times 3$  convolutional kernels are used for feature extraction, and after each maximum pooling layer, channels are doubled to prevent information loss. Simultaneously,  $1 \times 1$  convolutional kernels are added after  $3 \times 3$  convolutional kernels to compress eigenvectors. Lastly, global average pooling is adopted to reduce dimension, and the Softmax layer is utilized for prediction. Furthermore, batch normalization is used for improving the stabilization and accelerating the convergence of the model in process of training. The network model of darknet19 is shown in Figure 7.

In fact, darknet19 has good performance in target detection, and the established grasp detection network model in this paper only needs the output 5 grasp parameters. Thus, in order to simplify the process of training network,

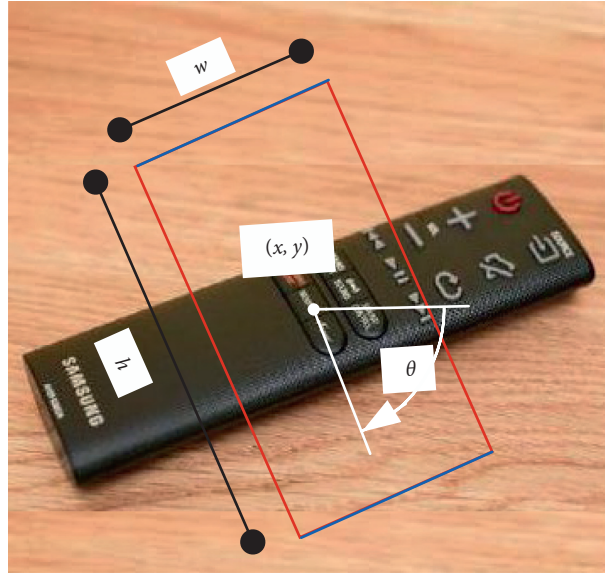


FIGURE 6: Schematic of grasp parameter model.

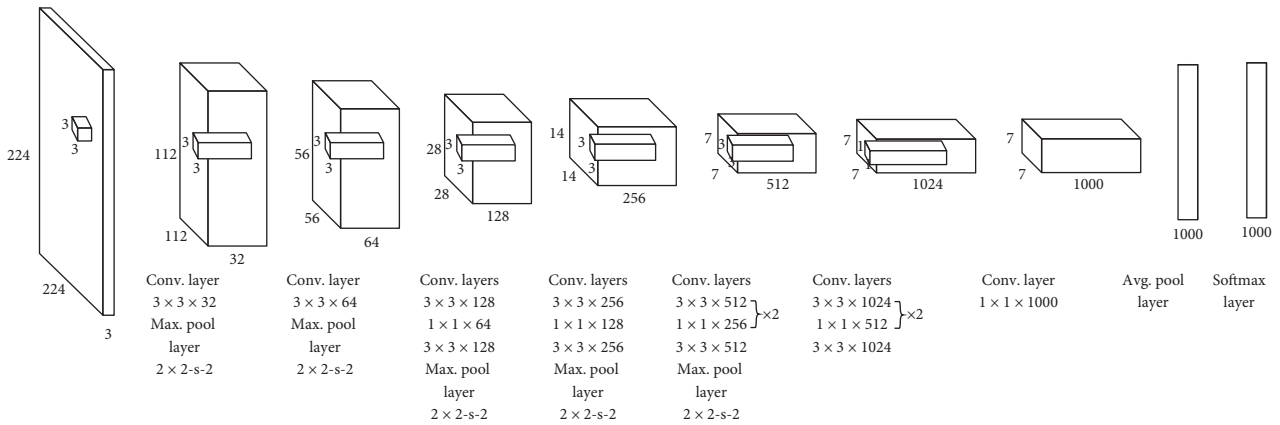


FIGURE 7: Network model of darknet19.

meanwhile shortening the process of forward reasoning and backpropagation, thereby to avoid the occurrence of overfitting, we construct a grasp detection network model based on the network structure of darknet19, which has a relatively simple structure and could adapt to the proposed grasp parameters.

On the other hand, both accuracy and real time in grasp detection are taken into consideration; the constructed grasp detection network model should be able to make full use of powerful learning ability and extraction ability of convolutional neural network on image features and could avoid multiple time-consuming classification calculations in a small part of the whole image. Hence, the established grasp detection network model should be able to carry out bounding box regression on the whole image to acquire the appropriate grasping rectangle.

In summary, based on the network architecture of darknet19, we put forward a grasp detection network model with the whole image as input and five grasp parameters as output, whose structure is displayed in Figure 8.

As shown in Figure 8, compared with darknet19, the grasp detection network model established in this paper prunes the  $1 \times 1$  convolutional kernel used for compressing eigenvectors, which was connected with  $3 \times 3$  convolutional kernel, and removes the  $3 \times 3$  convolutional kernel used for learning higher-level features, which was between  $1 \times 1$  convolutional kernel and maximum pooling layer. The eigenvectors of  $7 \times 7 \times 1024$  are obtained after six convolutional layers and pooling layers and without connection of pooling layers behind the last convolutional layer. In addition, the  $1 \times 1$  convolutional layer, fully connected layer, and Softmax output layer used for classification tasks are replaced by three fully connected layers with 1024, 512, and 5 neurons, respectively, where fully connected layers with 1024 and 512 neurons are used to deal with  $7 \times 7 \times 1024$  eigenvectors extracted by convolutional layer, and the last 5 neurons are used to output the grasp parameters.

When the original image is input into the network model, the convolutional layer is used to extract features from the image, and the fully connected layer of the last 5 neurons is used

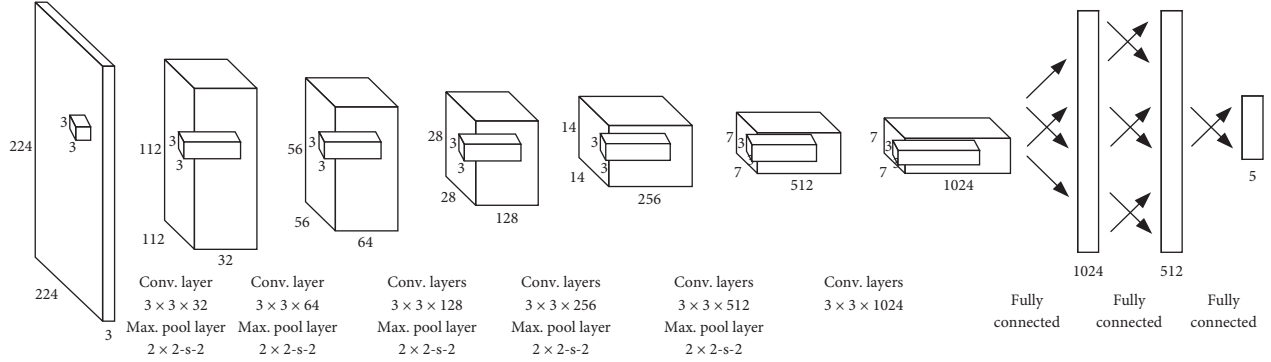


FIGURE 8: Structure of constructed grasp detection network model.

as the output layer corresponding to the coordinates of grasp parameters, where four neurons correspond to the position, width, and distance of the gripper. The grasping angle is symmetric; thus,  $\theta \in (-\pi/2, \pi/2)$ , but  $\tan \theta$  is monotone increasing in this interval. Accordingly, the last neuron corresponds to the  $\tan \theta$  value of the gripper relative to  $z$ -axis rotation angle. Although  $\theta$  between  $(-\pi/2, \pi/2)$  is reasonable,  $\tan \theta$  is closer to these two thresholds, and the value of  $|\tan \theta|$  is greater, which is quite disadvantageous to the calculation of regression and even leads to difficulty in continuing training the network model. To avoid the emergence of this situation, we further limit the range of  $\theta$ . Since  $\tan \pm 85^\circ \approx \pm 11$ , in this paper, the angle range is limited to  $\theta \in (-85^\circ, 85^\circ)$ , namely, only loss of  $5^\circ$ , and  $\tan \theta$  is limited to a small range, which is convenient for regression calculation of the model.

Indeed, the constructed network model is for single object grasp detection; hence, each object only needs to predict once grasp. That is, as long as the image is input into the model, our model could directly make a global regression prediction of the image.

During training the proposed grasp detection network model, the model randomly selects a real value as a label to carry our regression with the predicted value. Since label value is always changing each time, the network model is uneasy to overfit in grasp parameters of an object. In order to better training the proposed network model, we define the loss function, which could be expressed as

$$F_{\text{coord}} = \lambda_{\text{coord}} \left( (x - \hat{x})^2 + (y - \hat{y})^2 + (h - \hat{h})^2 + (w - \hat{w})^2 \right), \quad (6)$$

$$F_{\text{angle}} = \lambda_{\text{angle}} (\tan \theta - \tan \hat{\theta})^2, \quad (7)$$

$$F_{\text{total}} = F_{\text{coord}} + F_{\text{angle}}, \quad (8)$$

where  $\lambda_{\text{coord}}$  is the trade-off parameter of coordinate values losses,  $\lambda_{\text{angle}}$  is the trade-off parameter of angle values losses,  $F_{\text{coord}}$  is the coordinate values losses of the network,  $F_{\text{angle}}$  is the angle values losses of the network, and  $F_{\text{total}}$  is the total loss of network.

It can be seen from formula (6) and formula (7) that the paper adopts a sum of square errors to construct loss

function, but different weight factors are used for different parameters to ensure that the contribution of each parameter to the loss is approximately consistent. Through statistics, rectangular center coordinates  $x$  and  $y$  are mostly between 100 and 150 pixels, as well as  $h$  and  $w$  are mostly between 20 and 30 pixels. Obviously, it is unreasonable to add directly and proportionately to the loss. Indeed, grasp position is quite important, but the opening and closing size of the gripper is also equally important. Hence, the regulator of coordinate values  $\lambda_{\text{coord}}$  is added before error losses of  $x$  and  $y$ , whose value is 0.1. Similarly, since the value of  $\tan \theta$  is limited at the range of  $(-11, 11)$ , to adjust to the same level, the adjustment factor  $\lambda_{\text{angle}}$  is added before angular losses, whose value is 10. Through the above manners, losses of all parameters are basically guaranteed to account for the same proportion in total loss, which are conducive to the training network to obtain good results.

**3.3. Selection and Preprocessing of Dataset.** In order to verify the effectiveness of the proposed network model, it is necessary to select an appropriate dataset for the training model. At present, Cornell dataset is a widely used grasping dataset, which contains 240 common objects and 885 images obtained from different angles of these objects [1, 39]. In this dataset, numerous images contain the same kind of object, but the position and direction of the object in the image are different, which is extremely important for improving the robustness of the network model to the position and direction of the object during training. Consequently, this paper selects the Cornell grasping dataset to verify the validity of the proposed grasp detection network model.

However, in current data labels, cases that could not completely cover overall grasp positions and directions still exist. Thus, it is essential to preprocess the dataset to adapt the input of the model. In other words, we expand the dataset to achieve full coverage of input.

For the entire dataset, in order to prevent some objects in subsequent steps which are cut off, we primarily intercept pixel-sized areas of  $321 \times 321$  from the center in each image and utilize a filling algorithm to fill in the neighboring pixels to pixel-sized areas of  $501 \times 501$ . Then we randomly spin the image five times with a certain angle. Namely, the image is randomly, respectively, moved five times within 100 pixels in

$x$  and  $y$  directions. Lastly, pixel-sized area of  $320 \times 320$  from center in each image is cut out and scaled to the pixel-sized area of  $240 \times 240$  that the network model needs to input. At the same time, label values also need to be synchronized to match the changes of each image. The whole process of the data preprocessing algorithm is shown in Figure 9.

After preprocessing, the dataset is expanded 125 times, including 110625 images, which satisfies the requirements of the following network training.

In our implementation, we use a 50-fold cross-validation method to test our model. Meanwhile, we adopt two ways to segment the image. The first one is to randomly segment all the images in the dataset, which means that the most likely occurrence of the test set is objects seen during training, but the direction is random and unseen. This image segmentation method tests the sensitivity of the network model to angle. The other is to randomly segment each category of the object in data; that is, all images of the same object are in the same cross-validation set, which means that objects in the test set are unseen during training, but the direction is seen. This segmentation manner has higher requirements and greater difficulty for the model, which is to test the generalization ability of the network model. In fact, generalization ability is exactly what we expect the proposed model should have.

**3.4. PreTraining Grasp Detection Network.** As a matter of fact, the dataset used in this paper contains a limited amount of data; directly training the network model easily leads to network overfitting. Yet pretraining a large-scale convolutional neural network model could greatly shorten training time and avoid overfitting [40]. Hence, it is essential to pretraining the network model to avoid overfitting during training.

Due to data similarity between grasp detection and target detection is high, and the training set has 88500 images after expansion of the whole dataset via preprocessing, whose amount is large. Thus, we could use transfer learning to extract image features from networks trained by datasets in target detection for grasp detection.

Nevertheless, transfer learning has different processing manners for diverse application scenarios. Consider that the only distinction between grasp detection and target detection is the output of grasp detection which has an extra gripper angle. Therefore, after data classification in the network, we use parameters of six convolution layers to send the extracted eigenvectors to the following fully connected layer for processing and predicting results. The three fully connected layers are trained from scratch and only one initialization value is given.

**3.5. Training Grasp Detection Network.** After pretraining the network model, we adopt a small-batch gradient descent algorithm to training the network 100 times with the manner of end-to-end, where the value of each batch is 128. We set the learning rate  $\alpha$  to 0.0005, the weight attenuation coefficient  $\lambda$  to 0.01, and the dropout parameter among three

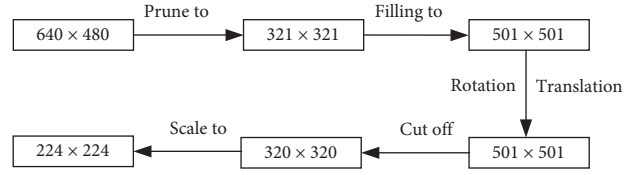


FIGURE 9: Flow of data preprocessing algorithm.

fully connected layers to 0.5. The loss of training processing is exhibited in Figure 10.

In Figure 10, the abscissa represents the number of training steps, and the ordinate refers to the corresponding loss value. Apparently, the total loss is decreasing with the increasing of iterative steps, but a short oscillation occurs when it decreases to a certain extent, and then it continues decreasing to a certain value, which indicates that the performance of the model for the training set tends to be stable at this time. Hence, in general, the model is reliable for the training set.

## 4. Experiments

**4.1. Select and Determine the Evaluation Index of Proposed Grasp Detection Network.** Point coordinates and rectangular coordinates are currently two general indexes to evaluate the performance of a grasp detection network [16, 41]. Indeed, point coordinates are to judge the quality of grasping via comparing the distance between the predicted coordinates of the center point in grasping rectangle and center points coordinates of all real grasping values, whereas this evaluation method does not consider the impact of grasping angle on accuracy, but angle value is particularly important in actual grasping. In addition, point coordinates also need to set another threshold to evaluate the results of point coordinates, which also affects the accuracy of calculation to a certain extent.

Rectangular coordinate is to judge the quality of grasping by comparing the difference between the predicted grasping angle and real grasping value. When the difference is less than  $30^\circ$  and the Jaccard similarity coefficient between the predicted grasping rectangle and real grasping value is greater than 25%, the grasping is considered to be effective [42]. In this paper, the Jaccard similarity coefficient is similar to Intersection-over-Union in target detection, which is defined as follows:

$$J(M_g, M_p) = \frac{|M_g \cap M_p|}{|M_g \cup M_p|}, \quad (9)$$

where  $M_g$  represents actual values of grasping rectangle and  $M_p$  refers to the predicted values of grasping rectangle.

Obviously, the value of the Jaccard similarity coefficient is larger, which indicates that the effect of grasp detection is better.

From above analysis, it can be concluded that the rectangle index considers both position and angle, which is more comprehensive than point coordinates and more convincing in judging the quality of grasping. Accordingly,

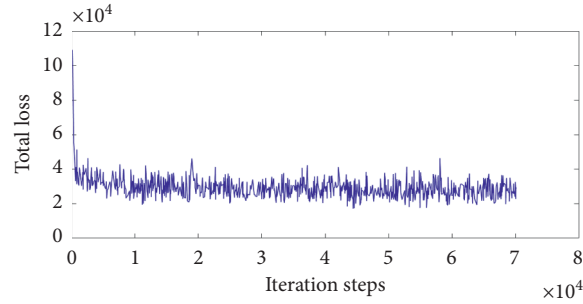


FIGURE 10: Changing curve of total loss.

in this paper, we adopt rectangle index to evaluate the performance of the proposed grasp detection network.

*4.2. Experimental Results and Analysis.* In order to validate the effectiveness of our network model, we conduct experimental verification on Cornell dataset. The image is input into the proposed network model, and the output result is the prediction grasping rectangle box of each input image. Some of the visual detection results are exhibited in Figure 11.

Obviously, detection results in Figure 11 illustrate that our model could detect the grasping region. Thus, to further illustrate the effectiveness of prediction, we calculate the Jaccard similarity coefficient of each prediction rectangle in Figure 11, and calculation results are shown in Table 2.

It can be clearly seen from Table 2 that all Jaccard similarity coefficients are greater than 0.25, which indicates that our grasp detection is effective, and grasp detection results for single object grasping could be regarded as good.

Through analysis of established network model, it can be known that acquired good detection results lie in two reasons. The first one is that our model adopts directly calculation of the loss and carry out global boundary regression on image to acquire the appropriate grasping rectangle. The other is that our model randomly selects a label value for each image during model training, which means that, after multiple training of dataset, the model predicts an average value for each object. Thus, for single object grasping, the predicted average value still has a good detection effect.

Additionally, to further verify the performance of the proposed network model, we make a comparison with other models based on convolutional neural networks, and the results are exhibited in Table 3.

It can be seen from above table that, in terms of detection accuracy, the prediction accuracy of our network model for image segmentation is 88.7%, and prediction accuracy for object segmentation is 87.2%; both of them stay at the third, belonging to an upper level. On the other hand, our research is inspired by literature [16, 39], and the comparison results

TABLE 2: Jaccard similarity coefficients of grasp detection in Figure 11.

Image no.	Jaccard similar coefficient
a	0.83
b	0.85
c	0.82
d	0.51
e	0.81
f	0.68
g	0.85
h	0.75
i	0.86
j	0.67

TABLE 3: Grasping prediction accuracy of different algorithms on Cornell dataset.

Algorithms	Image segmentation accuracy (%)	Object segmentation accuracy (%)
Jiang et al. [39]	60.5	58.3
Lenz et al. [13]	73.9	75.6
Redmon and Angelova [16]	88.0	87.1
Wang et al. [15]	81.8	N/A
Guo et al. [21]	93.2	89.1
Asif et al. [22]	88.2	87.5
Ribeiro et al. [23]	94.8	86.9
Trottier et al. [25]	87.7	86.6
<b>Ours</b>	<b>88.7</b>	<b>87.2</b>

in Table 3 show that our model is superior to the above two in detection accuracy, indicating that our research is meaningful even though it is not the best in above comparisons.

In summary, above experimental results demonstrate that the constructed network model has good detection results and high accuracy in single object grasping. Also, these results validate that our model is effective with strong generalization in direction and category.

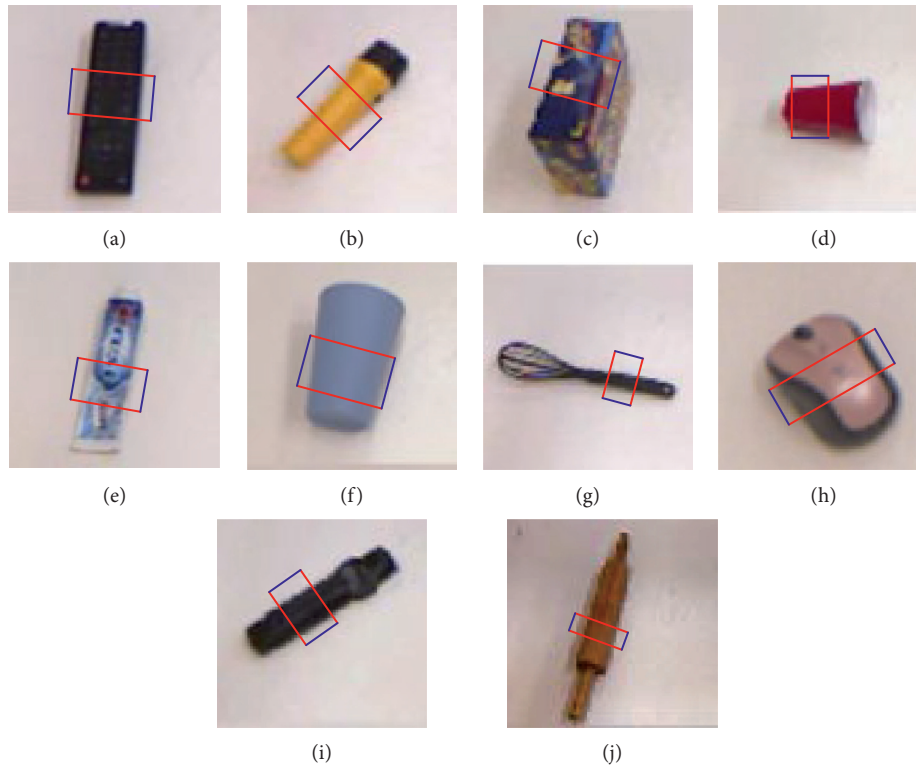


FIGURE 11: Some grasp detection results.

## 5. Conclusions

In this work, a single target grasp detection network based on a convolutional neural network is put forward, which generalizes the fitting of angle and position with high detection accuracy. Specifically, we simplified 3D space grasping into 2D planar grasping and modeled grasping parameters with the manner of five-dimensional representation. Afterward, we adopted end-to-end detection ways to construct a grasp detection network model with the image as input and five grasping parameters as output. In order to verify the effectiveness of the proposed grasp detection network model, the Cornell grasp dataset is selected and expanded to match the input of the model. Furthermore, a 50-fold cross-validation method was adopted to test our network model, and the image was split into two ways. Moreover, for the sake of avoiding overfitting of the network in training, the constructed network model was pretrained via transfer learning. Ultimately, experimental results indicate that, for single object grasping, the proposed grasp detection network has good detection results and high prediction accuracy, which demonstrates that our detection model has strong generalization in direction and category.

Particularly, in the future, using other datasets to further optimize and validate our model is a beneficial work to be finished. Also, applying the proposed network to actual grasping operation is worth being deeply researched.

## Data Availability

In this paper, the dataset is the Cornell dataset.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

Valuable suggestions given by Research Associate Baoshi Cao of the Harbin Institute of Technology are also acknowledged. This work was supported by the Self-Planned Task of the State Key Laboratory of Robotics and Systems under Grant No. SKLRS201910B.

## References

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.
- [4] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

- [6] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems NIPS'16*, Barcelona, Spain, December 2016.
- [7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR'2017*, pp. 2117–2125, Honolulu, HI, USA, July 2017.
- [8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, Venice, Italy, October 2017.
- [9] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon, "Attentionnet: aggregating weak directions for accurate object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2659–2667, Santiago, Chile, December 2015.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [11] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21–37, Amsterdam, The Netherlands, October 2016.
- [12] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, July 2017.
- [13] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.
- [14] Q. Yu, W. Shang, and C. Zhang, "Object grasp detecting based on three-level convolution neural network," *Jiqiren/Robot*, vol. 40, no. 5, pp. 762–768, 2018.
- [15] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering*, vol. 8, no. 9, pp. 1–12, 2016.
- [16] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1316–1322, Seattle, WA, USA, May 2015.
- [17] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4461–4468, Daejeon, South Korea, October 2016.
- [18] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3406–3413, Stockholm, Sweden, May 2016.
- [19] J. Xia, K. Qian, X. Ma, and H. Liu, "Fast planar grasp pose detection for robot based on cascaded deep convolutional neural networks," *Robot*, vol. 40, no. 6, pp. 794–802, 2018.
- [20] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519, Stockholm, Sweden, May 2016.
- [21] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1609–1614, Singapore, May 2017.
- [22] U. Asif, M. Bennamoun, and F. A. Soheli, "RGB-D object recognition and grasp detection using hierarchical cascaded forests," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 547–564, 2017.
- [23] E. G. Ribeiro, R. De Queiroz Mendes, and V. Grassi, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, vol. 139, no. 2, Article ID 103757, 2021.
- [24] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.
- [25] L. Trottier, P. Giguère, and B. Chaib-draa, "Dictionary learning for robotic grasp recognition and detection," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)*, Daejeon, South Korea, October 2016.
- [26] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3 dmatch: learning the matching of local 3d geometry in range scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, vol. 1, no. 2, p. 4, Honolulu, HI, USA, July 2017.
- [27] D. Cockburn, J.-P. Roberge, T.-H.-L. Le et al., "Grasp stability assessment through unsupervised feature learning of tactile images," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2238–2244, Singapore, May 2017.
- [28] S. Kong, X. Chen, Z. Wu et al., "An unsupervised grasp detection for water-surface object collection," in *Proceedings of the 38th Chinese Control Conference*, pp. 4421–4426, Guangzhou, China, July 2019.
- [29] F. Zhang, J. Leitner, M. Milford et al., "Towards vision-based deep reinforcement learning for robotic motion control," in *Proceedings of the Australasian Conference on Robotics and Automation*, Canberra, Australia, December 2015.
- [30] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [31] S. Gu, E. Holly, T. Lillicrap et al., "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396, Singapore, May 2017.
- [32] A. Zeng, S. Song, S. Welker et al., "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4238–4245, Madrid, Spain, October 2018.
- [33] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1549–1556, 2019.
- [34] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proceedings of the 1995 International Workshop on Artificial Neural Networks*, vol. 930, pp. 195–201, Torremolinos, Spain, 1995.

- [35] W. Malfliet, "The tanh method: a tool for solving certain classes of nonlinear evolution and wave equations," *Journal of Computational and Applied Mathematics*, vol. 164-165, pp. 529-541, 2004.
- [36] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947-951, 2000.
- [37] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pp. 1209-1216, Vancouver, Canada, December 2006.
- [38] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5062-5069, Anchorage, AK, USA, May 2010.
- [39] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: learning using a new rectangle representation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2011*, pp. 3304-3311, Shanghai, China, May 2011.
- [40] D. Guo, T. Kong, F. Sun, and H. Liu, "Object discovery and grasp detection with a shared convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA) 2016*, pp. 2038-2043, Stockholm, Sweden, May 2016.
- [41] J. Donahue, Y. Jia, O. Vinyals et al., "A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, Beijing China, June 2014.
- [42] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 57, 2018.



## Research Article

# 3D M-Net: Object-Specific 3D Segmentation Network Based on a Single Projection

Xuan Li , Sukai Wang , Xiaodong Niu , Liming Wang , and Ping Chen 

State Key Lab for Electronic Testing Technology, North University of China, Taiyuan 030051, China

Correspondence should be addressed to Liming Wang; [wlm@nuc.edu.cn](mailto:wlm@nuc.edu.cn) and Ping Chen; [chenping@nuc.edu.cn](mailto:chenping@nuc.edu.cn)

Received 2 June 2021; Accepted 28 June 2021; Published 13 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Xuan Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The internal assembly correctness of industrial products directly affects their performance and service life. Industrial products are usually protected by opaque housing, so most internal detection methods are based on X-rays. Since the dense structural features of industrial products, it is challenging to detect the occluded parts only from projections. Limited by the data acquisition and reconstruction speeds, CT-based detection methods do not achieve real-time detection. To solve the above problems, we design an end-to-end single-projection 3D segmentation network. For a specific product, the network adopts a single projection as input to segment product components and output 3D segmentation results. In this study, the feasibility of the network was verified against data containing several typical assembly errors. The qualitative and quantitative results reveal that the segmentation results can meet industrial assembly real-time detection requirements and exhibit high robustness to noise and component occlusion.

## 1. Introduction

In the industrial production process, real-time assembly detection is an essential link [1]. Especially for critical disposable products (such as fuses, solid rocket motors, and airbags), conventional functional testing destroys the product structure. Due to the particularity of this kind of product, abnormal assembly inevitably causes notable safety hazards and property losses, so these products must be detected one at a time before being put into use. Therefore, a real-time automatic assembly detection method that can match the production rhythm is highly important to improve production efficiency and product reliability.

Since X-rays can obtain internal information, this technology is widely applied in internal abnormality detection. To ensure the detection speed, a series of internal abnormality detection methods based on a single projection has been widely implemented in different fields, such as the security field [2–5] and the aerospace field [6–8]. These methods achieve rapid detection via the direct extraction of features from projections. However, in regard to the assembly detection of industrial products, these kinds of single-projection methods are susceptible to component

occlusion, thereby reducing the accuracy. The main reason is that industrial products possess complex structures, and the distribution of internal components is compact, so component occlusion is inevitable. Furthermore, projections contain integral information of all the components passed by the ray path. It is difficult to separate the information contribution of the different components. An effective way to avoid occlusion is to apply computed tomography (CT) algorithms. The 3D model of the product can provide richer structural information for detection while avoiding the influence of occlusion. However, the CT reconstruction algorithm requires complete projection data and consumes much time. Limited by the projection data acquisition speed and reconstruction speed, the CT reconstruction approach does not meet the needs of real-time detection.

Researchers have introduced convolutional neural networks (CNNs) [9] based on deep learning [10] in the field of X-ray 3D reconstruction and proposed a series of single-projection 3D reconstruction algorithms for specific targets. Henzler et al. [11] used the encoder-decoder network [12] to predict a low-resolution 3D model and fused the result with the projection to improve the resolution, thus achieving single-projection reconstruction of the mammalian skull.

Shen et al. [13] designed an automatic encoder network with an embedded conversion module and used the feature representation across dimensions to realize reconstruction of specific patients based on ultrasparse projection data. On this basis, Lei et al. [14] introduced generative adversarial networks (GANs) [15], using adversarial supervision to improve the realism of generated 3D images relative to ground truth images. Wang et al. [16] employed multiorgan template selection and smooth free-form deformation (FFD) strategies to generate high-quality manifold meshing models of organs. Based on the U-Net [17], Vlontzos et al. [18] proposed the 2D to 3D U-Net, which realizes 3D volume generation of the target organ based on a single projection. Compared to the traditional CT reconstruction algorithms, the above algorithms do not reconstruct 3D volumes by solving the mathematical inversion but rely on structural features extracted from the projection for reconstruction. By combining the structural priors implied in the dataset of a specific target, the 3D structure of the reconstruction result is constrained, thereby achieving a single-projection reconstruction of the specific target. These single-projection reconstruction algorithms highly reduce the data acquisition time, thus facilitating real-time detection based on 3D data.

The purpose of assembly detection is to determine the position and posture of different product components. Through segmentation of the internal components of a given product, the results of the segmentation algorithm can be applied to accurately determine the position and posture of the components. Since Long et al. [19] first applied fully convolutional networks (FCNs) to image segmentation, semantic image segmentation based on CNNs has become a research area of heightened interest, and many breakthroughs have been achieved. Researchers have successively proposed DeconvNet [20], SegNet [21], U-Net, LinkNet [22], DeepLab [23], PSPNet [24], and other image segmentation networks based on CNNs. These semantic image segmentation networks can be summarized as encoder-decoder networks, where the encoder is adopted for image feature extraction, and the decoder is employed to map the learned semantic features onto the pixel space to obtain the probabilistic classification of the different pixels. These algorithms are widely adopted in the medical field and have achieved many results [25–27]. However, these works segment the target from 2D slices, only consider 2D features in the cross section and ignore 3D features. Regarding assembly detection, industrial products contain many components with similar cross-sectional features but different 3D structures. It is difficult to accomplish an accurate distinction only via 2D segmentation of the cross section. Aiming at the semantic segmentation of 3D images, Milletari et al. [28] proposed a fully convolutional 3D segmentation network (V-Net) to directly segment the 3D volume and designed the Dice loss function to train the network. Yang et al. [29] introduced a pyramid pooling module into a 3D convolutional network and adopted a combination of global and local features for more accurate voxel prediction. In contrast to the above single-target segmentation algorithms, Gibson et al. [30] designed a dense FCN (Dense V-Net) for multiclass 3D segmentation.

In terms of assembly detection, whether the assembly is correct or not, the product exhibits a similar structure, with only partial differences. Based on this characteristic, by combining the single-projection reconstruction algorithm and the 3D segmentation algorithm, we proposed an end-to-end X-ray single-projection 3D segmentation network for specific products. The network adopts a single projection of any view as input and performs segmentation of different components under the same perspective. The proposed approach first generates asymmetric mappings with a deep encoder-decoder network under the constraints of a specific dataset, thereby adaptively extracting features from 2D projections and mapping them onto the 3D space domain. In the mapping process, by postponing cross-dimensional feature transformation and applying 2D convolution instead of 3D convolution for upsampling, the feature processing flow is optimized to reduce the calculations. Furthermore, a mixed loss function comprising Dice and cross-entropy terms is applied to solve the data imbalance issue. Compared to CT-based detection methods, the application of this network in assembly detection can reduce the data acquisition time and achieve real-time detection. Furthermore, this network can help to simplify imaging hardware and improve radiation utilization, thus reducing detection costs. To our knowledge, this is the first article to propose a single-projection 3D segmentation network.

## 2. Methods

*2.1. Principle.* The essence of semantic image segmentation algorithms is the pixelwise classification algorithm, which can be broadly regarded as involving the two stages of feature extraction and feature mapping. At the feature extraction stage, cascaded convolutional layers are used for feature extraction, usually accompanied by downsampling to reduce the dimensionality of features and finally form the semantic features of the image. At the feature mapping stage, upsampling is performed to map the learned discriminative features onto a high-resolution pixel space. Different networks add various feature transfer mechanisms (skip connection [17], pyramid pooling [24], etc.) to increase the information and accuracy of mapping. Finally, a probability vector is constructed for each pixel, and pixelwise classification is achieved via the prediction of pixels belonging to the different targets. Most image segmentation networks (such as FCNs [19], SegNet [21], and U-Net [17]) follow this process and have achieved great segmentation results. The projections and the reconstruction results should share semantic features, as they represent the same object [13]. Based on this consideration, previous works on single-projection reconstruction [11, 13, 14] have verified that, under the strict constraint condition that the structure of specific targets is similar, the 2D features containing local differences extracted from projections can be mapped onto 3D features and correctly expressed in the constructed 3D output. This study combines this idea with the semantic image segmentation algorithm to achieve 3D segmentation of specific targets based on a single projection. The following three problems need to be solved:

- (1) Computational cost of 3D feature processing: It is necessary to improve the efficiency of 3D feature processing to realize real-time segmentation under existing hardware resources.
- (2) Cross-dimensional manifold mapping: It is necessary to map the 2D features of the projection image onto the 3D structural features of the object in order to construct the probability vector output of the 3D voxels.
- (3) Data imbalance: It is necessary to solve the problem of inconsistent training efficiency for different segmentation targets due to volume differences.

Taking these three problems as clues, the following content of this section introduces the network architecture and loss function.

*2.2. Network Architecture.* The proposed network can be regarded as an extension of the encoder-decoder network model [12] and follows the process of feature extraction and feature mapping. As shown in Figure 1, the encoder network comprises four residual convolution blocks and five downsampling blocks. The residual convolution blocks extract 2D features from the input projections and gradually increase feature channels to 512. The downsampling blocks gradually reduce the spatial size of the input feature map to  $8 \times 8$  and keep the number of feature channels unchanged so that convert high-dimensional features into low-dimensional embedded semantic representations. The decoder network consists of five upsampling blocks, a feature transformation model, and three 3D convolution blocks. The upsampling blocks restore the low-dimensional features and gradually increase the spatial size of the feature maps to the target size ( $256 \times 256$ ). The feature transformation model transforms the high-dimensional feature representation across dimensions for the subsequent generation of the probability vector. Then, the number of channels of the 3D features is gradually increased through the 3D convolution blocks to ensure that the output is of the same size as that of the target probability vector ( $256 \times 256 \times 256$ ). Finally, the probability vector of each voxel is obtained through the softmax layer. Refer to section 2.5 for detailed network parameter settings.

*2.3. Improve the Efficiency of Feature Processing.* The 3D convolution process can maintain the spatial association of features and control the size of the output feature, so it is an essential operation in 3D segmentation. However, 3D convolution is associated with a large number of parameters and computations, occupying a large amount of memory. Under the existing hardware resources, this limits the resolution and speed of the segmentation algorithm. This problem is common in 3D segmentation networks and is usually solved by improving hardware utilization and optimizing the algorithm's computing efficiency. For example, literature [30] achieved high-resolution 3D segmentation through memory-efficient dropout and feature reuse.

To improve the feature processing efficiency to realize real-time 3D segmentation of industrial products, we

postponed feature cross-dimensional mapping and 3D convolution in the decoder network and adopted the same technique as reported in the literature [11], applying 2D convolution instead of 3D convolution for upsampling (as shown by the green arrow in Figure 1). 3D convolution is only employed in probability vector construction from 3D features (as shown by the red arrow in Figure 1). Specifically, in the 3D segmentation network, feature mapping in the decoder network is usually implemented via 3D convolution. The computation is mainly concentrated on upsampling. To improve the computational efficiency, we encode depth information into the channel dimension and apply 2D convolution instead of 3D convolution for upsampling, which highly reduces the number of parameters and computation. Since downsampling and upsampling comprise convolution processes with the same dimensions, skip connections similar to those in the U-Net [17] can be used in the network (shown by the dotted arrow in Figure 1). This can provide more detailed information for the feature mapping process, which is helpful for the segmentation of tiny structures. In the process of downsampling and upsampling, the feature channel is fixed to twice the spatial resolution, i.e.,  $2 \times 256 = 512$ . The structure of the downsampling and upsampling blocks and skip connections is shown in Figures 2(b) and 2(c), respectively. In addition, because of the notable depth of the network, in all 2D convolution operations (residual convolution blocks, downsampling blocks, and upsampling blocks), we adopt the residual learning scheme [31] to improve the training efficiency and avoid gradient disappearance, as shown in Figure 2(a).

*2.4. Cross-Dimensional Feature Mapping.* In the process of downsampling and upsampling, depth information is encoded in the channel dimension of the feature. This process can be regarded as a process involving the extraction and fusion of depth and structural information. To bridge the upsampling blocks and subsequent 3D convolution blocks, we designed a feature transformation model to decode depth information and realize cross-dimensional mapping. As shown in Figure 2(d), through the convolution operation with a kernel size of  $1 \times 1$  and rectified linear unit (ReLU) activation, the 2D convolutional layer learns the transformation of all 2D features and reorganizes the depth information implicit in the channel dimension. Then, the feature map is reshaped from  $256 \times 256 \times 512$  to  $256 \times 256 \times 256 \times 2$ . In this manner, the 2D features are transformed across dimensions for the subsequent generation of the probability vector. Next, we apply the 3D convolution operation with a kernel size of  $1 \times 1 \times 1$  and a stride of  $1 \times 1 \times 1$  to learn the transformations among all 3D features and maintain the feature size unchanged. The feature transformation model connects the 2D and 3D feature domains and maps the 2D features with hidden depth information into 3D features.

*2.5. Details of the Network Structure and Parameters.* The parameter settings of the entire network are summarized in Tables 1 and 2. The encoder network and the upsampling

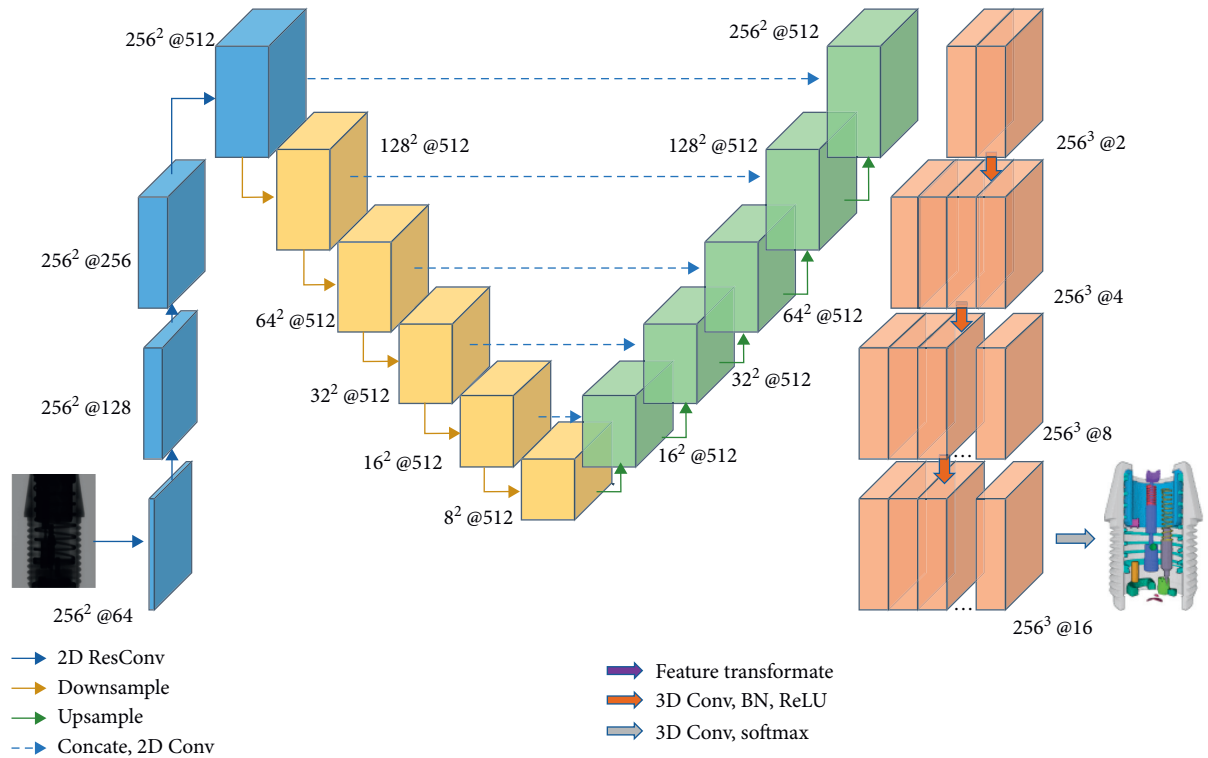


FIGURE 1: Schematic of the network architecture. The encoder network consists of residual convolution (blue arrow) and downsampling (yellow arrow) processes. The decoder network comprises upsampling (green arrow), a feature transformation model (purple arrow), and 3D convolution (red arrow). The upsampling and downsampling blocks share features through skip connections (dashed arrows). Finally, the probability vector is output through the softmax layer. The number next to the feature map indicates the spatial resolution and number of channels of the feature maps.

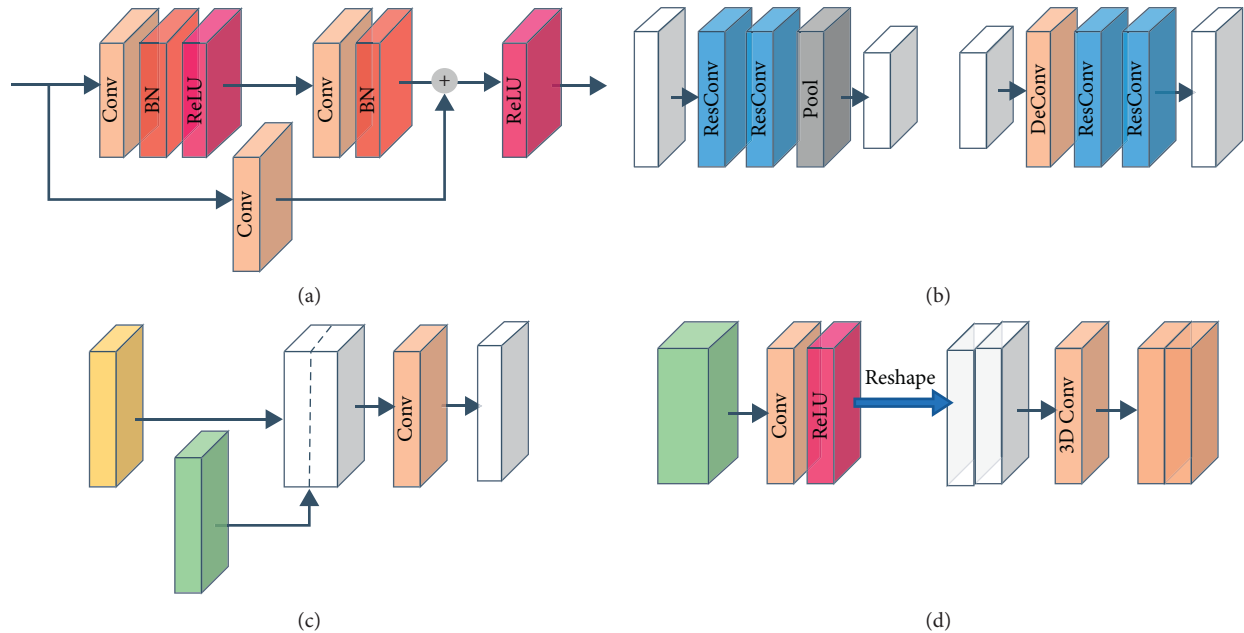


FIGURE 2: Schematic of the modules in the network. (a) Convolution residual block. (b) Downsampling block and upsampling block. (c) Skip connection. (d) Feature transformation model.

TABLE 1: Parametric structure of the essential components.

Layer	Parameters	Output size
ResConv block ( $k$ )	$3 \times 3 \times k$ Conv + BN + ReLU	$256^2 \times k$
	$3 \times 3 \times k$ Conv + BN	
	$1 \times 1 \times k$ Conv	
	ReLU	
DownSample block ( $n$ )	ResConv block (512)	$n^2 \times 512$
	ResConv block (512)	
	$2 \times 2$ max-pooling	
UpSample block ( $n$ )	$3 \times 3$ Deconv with $2 \times 2$ stride	$n^2 \times 512$
	ResConv block (512)	
	ResConv block (512)	
Skip connect ( $n$ )	Concatenate + $1 \times 1 \times 512$ Conv	$n^2 \times 512$
Transformation module	$1 \times 1 \times 512$ Conv + ReLU	$256^3 \times 2$
	Reshape	
	$1 \times 1 \times 1 \times 2$ Conv	
3D Conv block ( $k$ )	$3 \times 3 \times 3 \times k$ Conv + BN + ReLU	$256^3 \times k$

$k$  denotes the number of filters in the convolution layers, and  $n$  denotes the output resolution of the downsampling or upsampling block.

TABLE 2: Parametric structure of the entire network.

	Layer	Output size
Encoder network	ResConv block (64)	$256^2 \times 64$
	ResConv block (128)	$256^2 \times 128$
	ResConv block (256)	$256^2 \times 256$
	ResConv block (512)	$256^2 \times 512$
	DownSample block (128)	$128^2 \times 512$
	DownSample block (64)	$64^2 \times 512$
	DownSample block (32)	$32^2 \times 512$
	DownSample block (16)	$16^2 \times 512$
	DownSample block (8)	$8^2 \times 512$
	Decoder network	UpSample block (16)
UpSample block (32)		$32^2 \times 512$
UpSample block (64)		$64^2 \times 512$
UpSample block (128)		$128^2 \times 512$
UpSample block (256)		$256^2 \times 512$
Transformation module		$256^3 \times 2$
3D Conv block (4)		$256^3 \times 4$
3D Conv block (8)		$256^3 \times 8$
3D Conv block (16)		$256^3 \times 16$
$1 \times 1 \times 1$ Conv + softmax		$256^3 \times 15$

process in the decoder network comprise residual blocks. Each residual block comprises two sets of  $3 \times 3$  2D convolutional layers, batch norm layers, and ReLU activation functions. A residual path is added between the input and the second ReLU through a  $1 \times 1$  convolution layer. As input, the projection first performs 2D feature extraction through four residual blocks, thereby maintaining the spatial size fixed and gradually expanding the channels to 512. The downsampling block comprises two residual blocks and a  $2 \times 2$  max-pooling layer. Five downsampling blocks constitute the compression path of the feature stream. Through downsampling, a low-resolution feature with a large receptive field is gradually established, with a size of

$8 \times 8 \times 512$ . The upsampling block comprises a 2D deconvolution layer (with a kernel size of  $3 \times 3$  and a stride of  $2 \times 2$ ) and two residual blocks. Five upsampling blocks constitute the extension path of the feature stream. Through upsampling, the spatial size of the feature maps is gradually restored to  $256 \times 256 \times 512$ , which expands the spatial support of the lower-resolution feature maps. Via upsampling and downsampling, the depth information encoded in the channel dimension is integrated and reorganized. Between the upsampling and downsampling blocks of the same level, a path of feature flow transfer is added through a skip connection. In the skip connection, the feature maps from the downsampling block and previous upsampling block are first concatenated and then merged through a  $1 \times 1$  2D convolution operation to ensure that the number of channels remains fixed at 512. After passing through the feature transformation module, the 2D features with hidden depth information are transformed into 3D features. Next, three 3D convolution blocks are employed to reorganize the structural features and expand the channels. Each 3D convolution block comprises a  $3 \times 3 \times 3$  3D convolution layer, a batch norm layer, and a ReLU activation function. Finally, the network output is adjusted to a suitable size via  $1 \times 1 \times 1$  3D convolution and transformed into a probability vector by the softmax layer.

**2.6. Loss Function.** Due to differences in the sample number among the various segmentation targets, the network often ignores categories containing fewer samples, which in turn affects the segmentation effect of these categories [32]. In terms of the 3D segmentation of components in industrial products, the data imbalance issue is mainly reflected in the number of voxels. The voxel number of the components of different sizes often differs by several orders of magnitude. This kind of difference cannot be balanced through data enhancement, so in this study, we address this problem via loss function optimization.

The output of the proposed network is processed by the softmax layer for multiclassification, and the probability of each voxel belonging to the background or a certain component is calculated. To optimize the segmentation performance of the network, the accuracy of the predicted probability over the ground truth must be evaluated via calculating loss function. As a common loss function applied in segmentation, the Dice loss function [28] measures the accuracy of prediction by calculating the ratio between the intersection and union of the segmentation and ground truth regions. The Dice loss between the predicted probability  $P$  and ground truth  $R$  can be expressed as follows:

$$\mathcal{L}_{\text{Dice}}(P, R) = 1 - \frac{1}{M} \sum_{i=1}^M \frac{2 \sum_{j=1}^N p_{i,j} r_{i,j} + \epsilon}{\sum_{j=1}^N p_{i,j}^2 + \sum_{j=1}^N r_{i,j}^2 + \epsilon} \quad (1)$$

where  $M$  is the number of categories in the probability vector, and each category represents a kind of component or background (the background is set to category 0). Moreover,  $N$  is the number of voxels,  $p_{i,j}$  and  $r_{i,j}$  denote the probability that the  $j^{\text{th}}$  voxel belongs to the  $i^{\text{th}}$  category in the predicted

probability and the ground truth, respectively. And  $\varepsilon$  is applied to prevent the denominator from equalling 0, which is set to  $10^{-10}$  in this study. The Dice loss balances the voxel number of the different categories through the square term in the denominator. However, due to the complex gradient form of the Dice loss, gradient saturation occurs in the training process, which often leads to training instability. To solve this problem, we added a weighted cross-entropy (WCE) term to the Dice loss. The WCE loss is defined as follows:

$$\mathcal{L}_{WCE} = \sum_{i=1}^M \left( \sum_{j=1}^N \omega_i r_{i,j} \log(p_{i,j}) \right), \quad (2)$$

$$\omega_i = \frac{1}{\sum_{j=1}^N r_{i,j} + \varepsilon},$$

where  $\omega_i$  is the weight of the  $i^{\text{th}}$  category, which is used to penalize the gradient contribution of the large-size component in training. Therefore, the mixed loss is defined as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{Dice}} + (1 - \alpha) \mathcal{L}_{\text{WCE}}. \quad (3)$$

where  $\alpha$  balances the Dice and the WCE terms, which is set to 0.5.

**2.7. Implementation Details.** The network is implemented using the Tensorflow framework and optimized with the Adam optimizer at an initial learning rate of  $10^{-4}$  and a minibatch size of 5. In the training process, we evaluate the model on the validation set and gradually reduce the learning rate from  $10^{-4}$  to  $10^{-6}$ . The training and testing of the network are carried out on a workstation with an E5-2620 CPU, 32 GB of RAM, and a TITAN RTX GPU.

### 3. Material

Taking a fuse as the detection target, we perform data acquisition. Under the best imaging conditions, we acquire 1080 projections of the fuse at equal angular intervals on the YXLON FF20 CT system with tube voltage 160kV and current  $40 \mu\text{A}$  and then adopt the FDK algorithm for reconstruction. Next, regarding the 14 critical fuse components, the reconstructed 3D image was manually segmented. Specifically, each reconstructed slice was segmented with the watershed algorithm involving artificial participation, and all the segmented slices were then combined into a 3D segmented image as the ground truth data for training the network. Since the perspective of the reconstruction result depends on the order of the projections, we reordered the projections before reconstruction so that the components attained the same spatial distribution in the reconstruction results. In addition, as the input of the network, the projections were resized into  $256 \times 256$  and normalized to  $[0, 1]$ . For the convenience of description, we numbered the 14 critical components, as shown in Figure 3.

Regarding the most error-prone striker and spring, according to typical assembly errors (posture error, position

error, and omission), we set a total of six different assembly situations, as shown in Figure 4. For each situation, 12 sets of data were generated through the abovementioned data acquisition process. Before acquiring each set of data, the fuse has been reassembled. Ten sets of data were used for training. Moreover, to control the size of the training dataset, we randomly selected half of them as the training dataset, containing 32400 samples. The rest two sets were reserved for validating and testing, each containing 6480 samples.

## 4. Experiment Results and Discussion

**4.1. Segmentation Results of the Proposed Network.** We evaluate the segmentation performance of our network on the test dataset and randomly select a sample from each assembly situation for display. Figure 4 shows the 3D rendering of the segmentation results. To avoid occlusion, the results are shown as anatomical diagrams. In addition, we randomly select four slices from the segmentation results to compare the segmented foreground regions, as shown in Figure 5. The yellow, red, and green areas represent the ground truth, predicted segmentation, and overlap area, respectively. To increase the prominence of the difference, we display magnified views of partial areas. Furthermore, we adopt four metrics for quantitative analysis of the network segmentation results, namely, the Dice similarity coefficient (DSC), Jaccard similarity coefficient (JSC), positive prediction value (PPV), and sensitivity (SEN). These metrics are defined as follows:

$$\begin{aligned} \text{DSC} &= \frac{2 \|V_{gt} \cap V_{pd}\|}{\|V_{gt}\| + \|V_{pd}\|}, \\ \text{JSC} &= \frac{\|V_{gt} \cap V_{pd}\|}{\|V_{gt} \cup V_{pd}\|}, \\ \text{PPV} &= \frac{\|V_{gt} \cap V_{pd}\|}{\|V_{pd}\|}, \\ \text{SEN} &= \frac{\|V_{gt} \cap V_{pd}\|}{\|V_{gt}\|}, \end{aligned} \quad (4)$$

where  $V_{gt}$  and  $V_{pd}$  denote the ground truth and predicted segmentation voxels, respectively. The quantitative results of the different component segmentations are summarized in Table 3.

The qualitative and quantitative analysis results indicate that the difference between the segmentation result of the network and the manual segmentation result is very small. The differences are mainly concentrated along the edge of the components and include mispredicted scattered points. The segmentation results fully reflect the assembly situation of the fuse. The advantage of the network is that the use of projections from any angle as the input can reduce the dependence on mechanical equipment, which helps simplify the imaging system and reduce the cost of detection. In

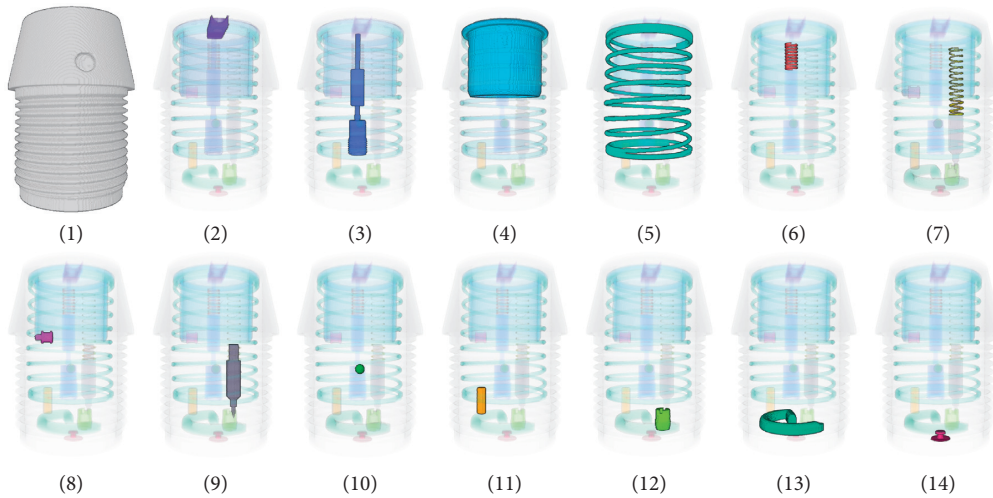


FIGURE 3: Segmentation of critical components. The spring and striker are numbered as 7 and 9, respectively.

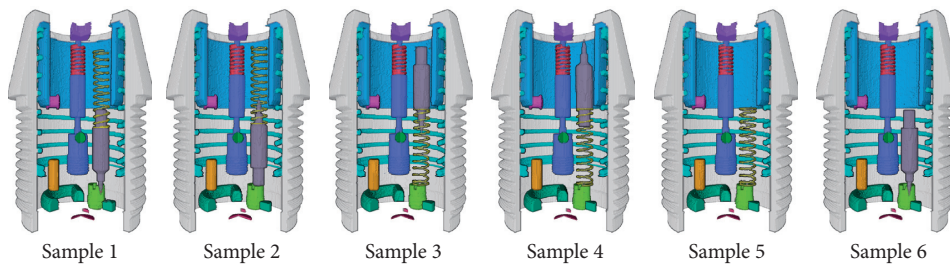


FIGURE 4: Three-dimensional rendering of the segmentation results. Sample 1: correct assembly. Sample 2: the striker is assembled to point upward. Sample 3: the spring is assembled below the striker. Sample 4: the spring is assembled below the striker with the striker points upward. Sample 5: the striker is missing. Sample 6: the spring is missing. To avoid occlusion, anatomical diagrams are shown here.

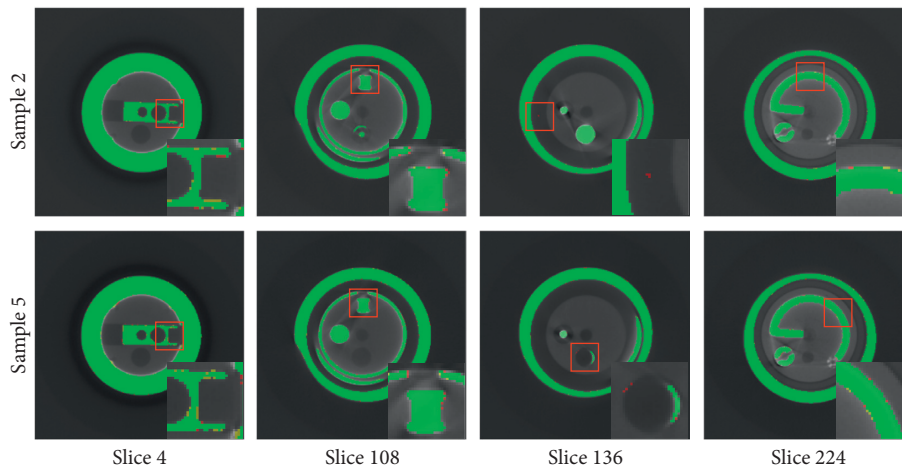


FIGURE 5: Slices of the segmentation results. The yellow, red, and green areas indicate the ground truth, predicted segmentation, and overlap area, respectively. To make the difference prominent, we display magnified views of partial areas, which are marked with red boxes.

addition, the segmentation results output by the network are generated in the same perspective, which allows the position and posture information obtained from the segmentation results to be directly used to infer the assembly situation without any coordinate transformation.

*4.2. Comparison to General 3D Segmentation Networks.* To our knowledge, there is no 3D segmentation algorithm based on a single projection. Therefore, we compare our network to general segmentation algorithms based on 3D images. U-Net [17] and V-Net [28] are the baseline

TABLE 3: Quantitative results obtained by the different component segmentations.

Components no.	1	2	3	4	5	6	7
DSC (%)	98.6	97.5	97.6	97.8	97.5	96.7	92.7
JSC (%)	97.3	96.1	97.3	96.6	96.1	94.9	91.6
PPV (%)	97.6	97.0	97.5	97.5	97.5	95.2	91.6
SEN (%)	98.6	98.1	98.7	98.0	97.5	98.5	93.7
Components no.	8	9	10	11	12	13	14
DSC (%)	96.7	91.6	97.2	98.1	97.0	98.6	96.8
JSC (%)	95.6	90.2	96.4	97.4	95.2	97.3	94.8
PPV (%)	95.6	90.8	96.4	97.4	95.2	97.4	94.8
SEN (%)	98.9	92.4	98.8	98.9	98.8	98.8	98.7

architectures for 2D and 3D image segmentation, respectively, which have been widely applied and adapted. Therefore, V-Net and 3D U-Net [33] (a 3D variant of U-Net) are selected as candidates for comparison. Since the original V-Net and 3D U-Net are designed for binary segmentation, we extend their loss functions to support multiclass data. Applying the CT reconstruction result and artificial segmentation result as the input and ground truth data, we train the V-Net and 3D U-Net on the training dataset and then test these networks on the test dataset. The qualitative and quantitative results of the different algorithms are shown in Figures 6 and 7 and Table 4.

The comparison reveals that the difference between the segmentation results obtained with the proposed network and the general 3D segmentation networks is extremely small. The performance of the proposed network almost reaches the level of the general 3D segmentation algorithms. It should be emphasized that the proposed network uses a single projection as the input for 3D segmentation, and a single segmentation requires approximately 0.2 seconds. Applying this network to industrial product assembly detection can greatly reduce the time required for data acquisition and 3D reconstruction and achieve real-time detection, which is of great significance for industrial products with a high production speed and huge production.

**4.3. Segmentation Results with Noise.** Quantum fluctuation noise in radiography obeys the Poisson distribution. Therefore, Poisson noise is added to the projections for analysis to illustrate the robustness of our network to noise. Noise addition is according to the following formula:

$$P_i \sim \text{Poisson}\{b_0 e^{-l_i}\}, \quad (5)$$

where  $P_i$  is the detector measurement along the  $i^{\text{th}}$  ray,  $b_0$  is the blank scan factor, and  $l_i$  is the line integral of the attenuation coefficients along the  $i^{\text{th}}$  ray. The Poisson noise level can be adjusted by setting the blank scan factor  $b_0$ . In this study,  $b_0$  is varied from  $1 \times 10^6$  to  $1 \times 10^3$ . During the decrease of  $b_0$ , several segmentation results with notable changes are shown in Figure 8. The performance metrics of the segmentation results are summarized in Table 5.

Before  $b_0$  decreases to  $1 \times 10^5$ , the segmentation performance of the network remains relatively stable. When the noise level is worse than  $1 \times 10^4$ , the components in the

segmentation results start to exhibit adhesion and the number of scattered points increases. When the noise level further deteriorates to  $4 \times 10^3$ , part of the information in the projection is masked by the noise. In the segmentation results, certain components are structurally missing, and the number of scattered points further increases. The results demonstrate that when  $b_0$  is greater than  $1 \times 10^5$ , the network effectively suppresses noise, and the segmentation results completely and accurately reflect the position, structure, and posture information of each component. The proposed network remains robust to a relatively broad range of noise levels.

**4.4. Segmentation Results with Occlusion.** We selected samples in different occlusion cases for comparison. Figure 9 shows the segmentation results in the three occlusion cases and the grayscale level profiles extracted along the dashed red line.

In the projections and the grayscale level profiles, it is difficult to determine whether the striker exists in cases 2 and 3 with the naked eye. Comparing the former two samples demonstrates that the network can use projections from different angles for segmentation and can completely segment the occluded component. Comparing the latter two samples reveals that the network can perform correct segmentation in the different assembly situations with similar projections. Therefore, the proposed network achieves high robustness to occlusion. In assembly detection, the network can effectively avoid the influence caused by component occlusion.

**4.5. Segmentation Results with Untrained Assembly Errors.** In order to further verify the effectiveness of the proposed network, we set up two additional assembly errors for the spring and the striker (the striker missed with the spring stuck upside, and the spring missed with the striker stuck upside) and acquire the data under these two wrong assembly conditions for testing. The segmentation results are shown in Figure 10 and Table 6.

The results indicate that for untrained assembly errors, the network can also correctly extract the features of each component and perform correct segmentation. Compared with the trained data, there is no noticeable difference in the performance metrics of the segmentation results. Therefore,



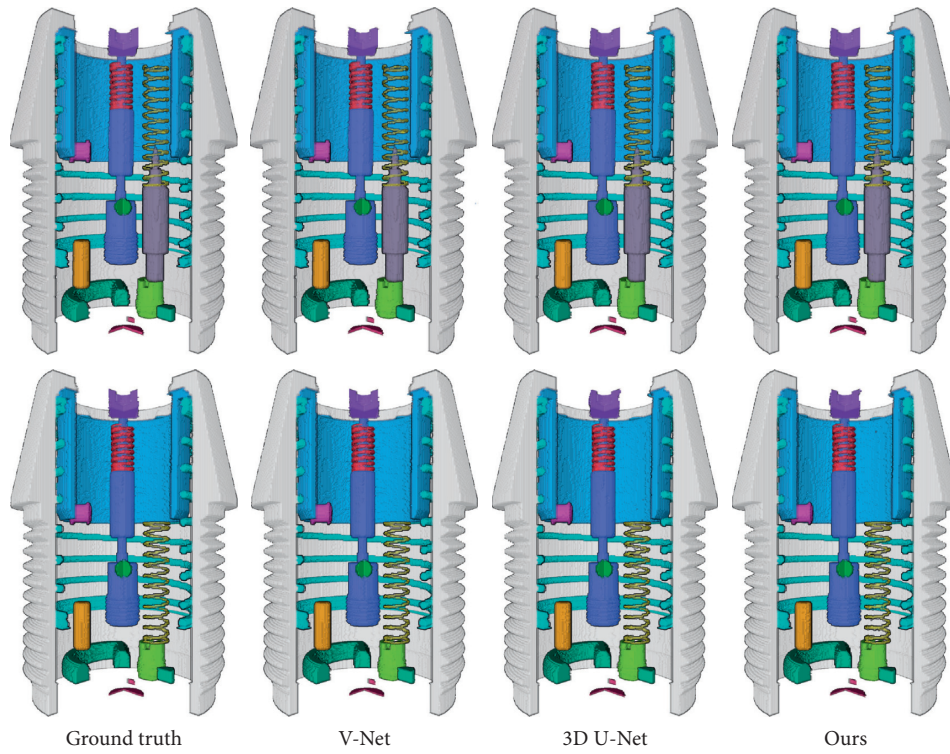


FIGURE 6: Segmentation results of the different algorithms.

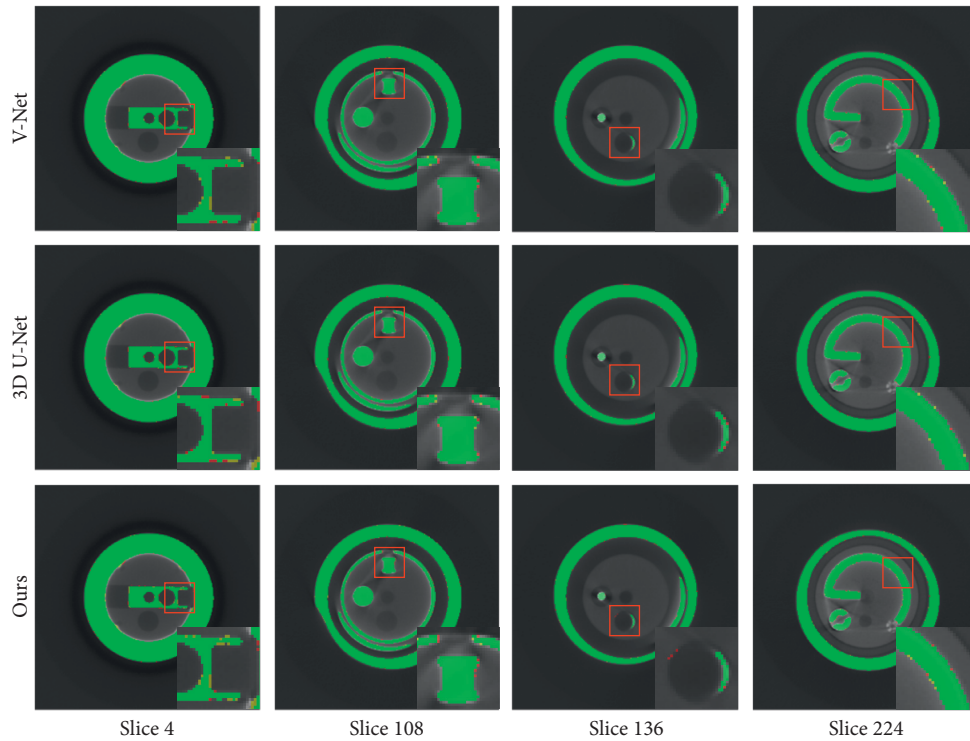


FIGURE 7: Slices of the segmentation results of the different algorithms. The yellow, red, and green areas indicate the ground truth, predicted segmentation, and overlap area, respectively. To make the difference prominent, we display magnified views of partial areas, which are marked with red boxes.

TABLE 4: Quantitative results obtained by the different segmentation algorithms.

	V-Net	3D U-Net	Ours
DSC (%)	97.2	96.8	96.7
JSC (%)	96.1	95.7	95.4
PPV (%)	96.2	96.4	95.8
SEN (%)	97.6	97.5	97.7

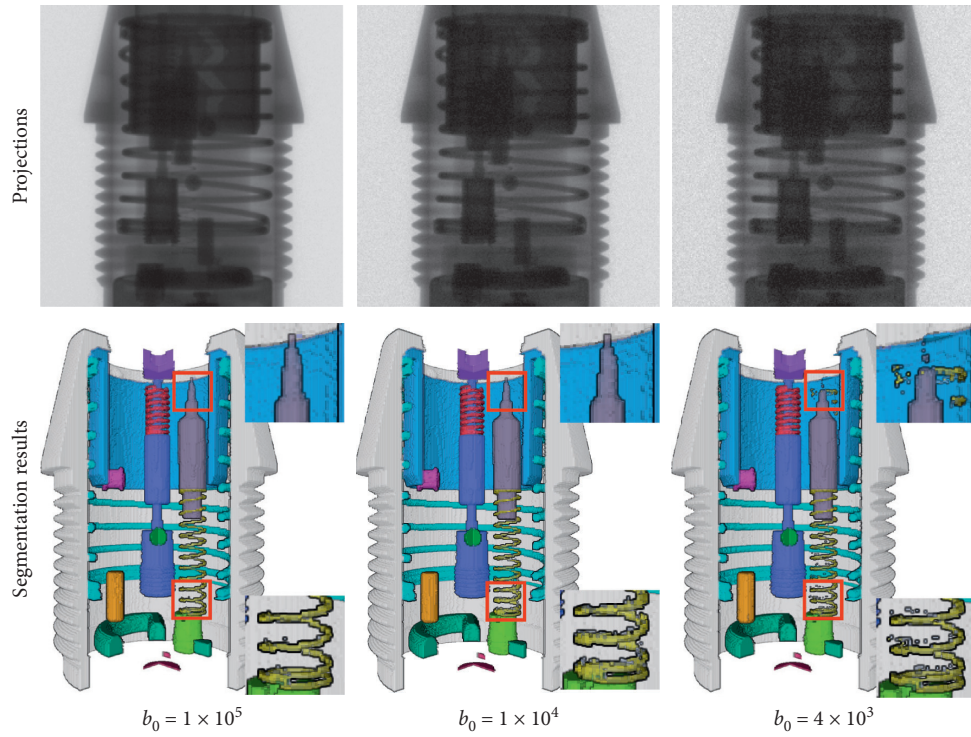


FIGURE 8: Segmentation results under the different noise levels. The first row shows the projections under the different levels of noise. The second row shows the predicted segmentation results. The zoomed regions of interest are shown on the right side.

TABLE 5: Quantitative results obtained under the different noise levels.

	$b_0 = 1 \times 10^5$	$b_0 = 1 \times 10^4$	$b_0 = 4 \times 10^3$
DSC (%)	96.5	96.2	94.8
JSC (%)	95.1	94.6	92.0
PPV (%)	95.6	95.1	93.0
SEN (%)	97.4	97.4	96.7

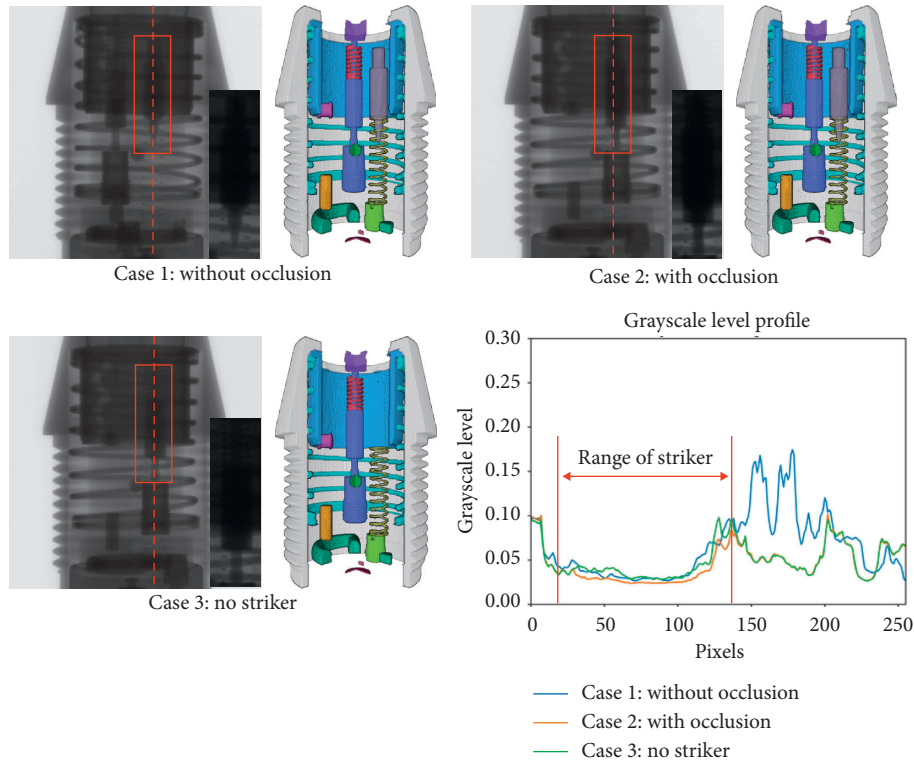


FIGURE 9: Segmentation results in different occlusion cases. Case 1: without occlusion. Case 2: with occlusion. Case 3: no striker. The striker is marked with a red box, and the zoomed views are shown on the right side. The grayscale level profiles are extracted along with the red dashed line.

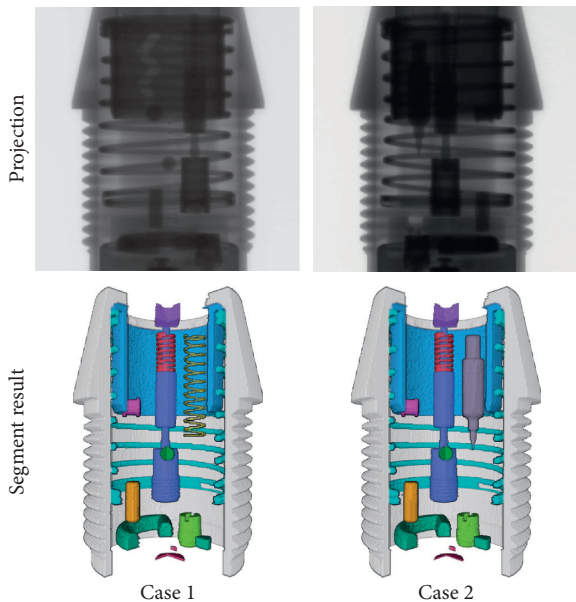


FIGURE 10: Segmentation results of untrained data. Case 1: the striker missed with the spring stuck upside. Case 2: the spring missed with the striker stuck upside.

TABLE 6: Quantitative results obtained by untrained assembly errors.

	Case 1	Case 2
DSC (%)	96.6	96.5
JSC (%)	95.2	95.3
PPV (%)	95.6	95.6
SEN (%)	97.5	97.4

for the assembly errors of the striker and the spring, the segmentation results can be applied to detect effectively.

### 5. Conclusion

In this study, we proposed a multiclass 3D segmentation network based on a single X-ray projection by combining the single-projection reconstruction algorithm and the semantic image segmentation algorithm. Adopting a single projection as the input, the network can segment different targets within a specific object and can output 3D segmentation results. The experimental results indicate that the segmentation results of the network completely reflect the position, structure, and posture information of the different internal

targets, and the segmentation performance for the specific objects is close to that of the 3D semantic image segmentation network. In addition, the network achieves high robustness to noise and component occlusion. The advantage of implementing the network in assembly detection is that it takes a single projection to perform 3D segmentation, which can improve the ray utilization rate and detection efficiency, thereby realizing real-time detection. Furthermore, the network is suitable for projections from different angles, which can simplify the imaging system and help reduce detection costs.

In the application process, the network can be directly deployed in digital radiography detection systems without any additional machinery or imaging equipment. However, the network has certain drawbacks and limitations. First, in contrast to the general semantic image segmentation algorithm, the network performs segmentation of specific objects, which suggests that changing the detection products requires network retraining. Second, the network relies on complete training data, which means that it needs to acquire data of different assembly situations for training.

To solve the problem whereby training data are difficult to obtain, in future work, we plan to conduct research on simulation data synthesis to reduce the difficulty and time cost of training data acquisition.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was funded by the National Natural Science Foundation of China under Grant no. 61871351.

## References

- [1] J.-I.-R. Cojocaru, D. Popescu, and L. Ichim, "Real-time assembly fault detection using image analysis for industrial assembly line," in *Proceedings of the 2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 484–487, Milan, Italy, July 2020.
- [2] D. Mery, E. Svec, M. Arias, V. Riffo, J. M. Saavedra, and S. Banerjee, "Modern computer vision techniques for X-Ray testing in baggage inspection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 682–692, 2017.
- [3] S. Lyu, X. Tu, and Y. Lu, "X-ray image classification for parcel inspection in high-speed sorting line," in *Proceedings of the 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, Beijing, China, October 2018.
- [4] J. Liu, X. Leng, and Y. Liu, "Deep convolutional neural network based object detector for X-ray baggage security imagery," in *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1757–1761, Portland, OR, USA, November 2019.
- [5] Y. Wei, Z. Zhu, H. Yu, and W. Zhang, "An automated detection model of threat objects for X-ray baggage inspection based on depthwise separable convolution," *Journal of Real-Time Image Processing*, vol. 18, no. 3, pp. 923–935, 2021.
- [6] Y. Gong, H. Shao, J. Luo, and Z. Li, "A deep transfer learning model for inclusion defect detection of aeronautics composite materials," *Composite Structures*, vol. 252, Article ID 112681, 2020.
- [7] Z.-H. Chen and J.-C. Juang, "AE-RTISNet: aeronautics engine radiographic testing inspection system net with an improved fast region-based convolutional neural network framework," *Applied Sciences*, vol. 10, no. 23, p. 8718, 2020.
- [8] D. Gamdha, S. Unnikrishnakurup, K. J. J. Rose et al., "Automated defect recognition on X-ray radiographs of solid propellant using deep learning based on convolutional neural networks," *Journal of Nondestructive Evaluation*, vol. 40, no. 1, p. 18, 2021.
- [9] B. Aubert, C. Vazquez, T. Cresson, S. Parent, and J. A. de Guise, "Toward automated 3D spine reconstruction from biplanar radiographs using CNN for statistical spine model fitting," *IEEE Transactions on Medical Imaging*, vol. 38, no. 12, pp. 2796–2806, 2019.
- [10] Z. Jiang, Y. Chen, Y. Zhang, Y. Ge, F.-F. Yin, and L. Ren, "Augmentation of CBCT reconstructed from under-sampled projections using deep learning," *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, pp. 2705–2715, 2019.
- [11] P. Henzler, V. Rasche, T. Ropinski, and T. Ritschel, "Single-image tomography: 3D volumes from 2D cranial X-Rays," *Computer Graphics Forum*, vol. 37, no. 2, pp. 377–388, 2018.
- [12] H. Shan, Y. Zhang, Q. Yang et al., "3-D convolutional encoder-decoder network for low-dose CT via transfer learning from a 2-D trained network," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1522–1534, 2018.
- [13] L. Shen, W. Zhao, and L. Xing, "Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning," *Nature Biomedical Engineering*, vol. 3, no. 11, pp. 880–888, 2019.
- [14] Y. Lei, Z. Tian, T. Wang et al., "Deep learning-based real-time volumetric imaging for lung stereotactic body radiation therapy: a proof of concept study," *Physics in Medicine & Biology*, vol. 65, no. 23, Article ID 235003, 2020.
- [15] I. Goodfellow, "Generative adversarial networks," 2014, <http://arxiv.org/abs/1406.2661v1>.
- [16] Y. Wang, Z. Zhong, and J. Hua, "DeepOrganNet: on-the-fly reconstruction and visualization of 3D/4D lung models from single-view projections by deep deformation network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, 9701 pages, 2019.
- [17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," pp. 234–241, 2015, <http://arxiv.org/abs/1505.04597v1>.
- [18] A. Vlontzos, S. Budd, B. Hou, D. Rueckert, and B. Kainz, "3D probabilistic segmentation and volumetry from 2D projection images," *Thoracic Image Analysis*, vol. 12, pp. 48–57, 2020.
- [19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.
- [20] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the*

- 2015 *IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1528, Santiago, Chile, December 2015.
- [21] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: a deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [22] A. Chaurasia and E. Culurciello, “LinkNet: exploiting encoder representations for efficient semantic segmentation,” in *Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, St. Petersburg, FL, USA, December 2017.
- [23] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.
- [24] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, Honolulu, HI, USA, July 2017.
- [25] R. Hasegawa, Y. Iwamoto, L. Lin, H. Hu, and Y.-W. Chen, “Automatic segmentation of liver tumor in multiphase CT images by mask R-CNN,” in *Proceedings of the 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*, pp. 231–234, Kyoto, Japan, March 2020.
- [26] K. He, C. Lian, B. Zhang et al., “HF-UNet: learning hierarchically inter-task relevance in multi-task U-net for accurate prostate segmentation in CT images,” *IEEE Transactions on Medical Imaging*, vol. 99, 2021.
- [27] A. Saood and I. Hatem, “COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet,” *BMC Medical Imaging*, vol. 21, no. 1, p. 19, 2021.
- [28] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-net: fully convolutional neural networks for volumetric medical image segmentation,” in *Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, Stanford, CA, USA, October 2016.
- [29] G. Yang, G. Li, T. Pan et al., “Automatic segmentation of kidney and renal tumor in CT images based on 3D fully convolutional neural network with pyramid pooling module,” in *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3790–3795, Beijing, China, August 2018.
- [30] E. Gibson, F. Giganti, Y. Hu et al., “Automatic multi-organ segmentation on abdominal CT with dense V-networks,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 8, pp. 1822–1834, 2018.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908, pp. 630–645, 2016.
- [32] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Proceedings of the International Workshop on Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 240–248, Québec City, QC, Canada, September 2017.
- [33] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-net: learning dense volumetric segmentation from sparse annotation,” in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention-MICCAI 2016*, pp. 424–432, Athens, Greece, October 2016.

## Research Article

# A New Hybrid Forecasting Model Based on SW-LSTM and Wavelet Packet Decomposition: A Case Study of Oil Futures Prices

Jie Wang <sup>1</sup> and Jun Wang<sup>2</sup>

<sup>1</sup>Department of Statistics, College of Science, North China University of Technology, Beijing 100144, China

<sup>2</sup>School of Science, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Jie Wang; [jiewang@ncut.edu.cn](mailto:jiewang@ncut.edu.cn)

Received 27 May 2021; Revised 26 June 2021; Accepted 1 July 2021; Published 13 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Jie Wang and Jun Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The crude oil futures prices forecasting is a significant research topic for the management of the energy futures market. In order to optimize the accuracy of energy futures prices prediction, a new hybrid model is established in this paper which combines wavelet packet decomposition (WPD) based on long short-term memory network (LSTM) with stochastic time effective weight (SW) function method (WPD-SW-LSTM). In the proposed framework, WPD is a signal processing method employed to decompose the original series into subseries with different frequencies and the SW-LSTM model is constructed based on random theory and the principle of LSTM network. To investigate the prediction performance of the new forecasting approach, SVM, BPNN, LSTM, WPD-BPNN, WPD-LSTM, CEEMDAN-LSTM, VMD-LSTM, and ST-GRU are considered as comparison models. Moreover, a new error measurement method (multiorder multiscale complexity invariant distance, MMCID) is improved to evaluate the forecasting results from different models, and the numerical results demonstrate that the high-accuracy forecast of oil futures prices is realized.

## 1. Introduction

Crude oil is a natural and nonrenewable resource that has an irreplaceable effect on the development of the global economy and international financial markets. Since oil is the main source of energy production, it is often considered the single important commodity in the world. The price fluctuations of crude oil may affect the economic situation, social stability, and even national security in the world [1]. Meanwhile, international crude oil price series are regarded as nonlinear and nonstationary time series. Hence, accurate forecasting of the crude oil price is a challenging task of energy market and has increasingly become an active research field.

In recent years, numerous methods for time series predictions have been proposed [2–13]. These methods can be classified into the following three categories: traditional econometric models, machine learning approaches and deep learning models. The autoregressive integrated moving

average model (ARIMA) is a popular statistical model applied to time series prediction. Liu et al. [3] proposed two novel forecasting models based on ARIMA, which was employed to forecast two sections of actual wind speed series. Abdollahi and Ebrahimi [4] established a new composite model to predict Brent crude oil prices by integrating the adaptive neuro fuzzy inference system (ANFIS), autoregressive fractionally integrated moving average (ARFIMA), and Markov-switching models. However, the traditional econometric models have evident shortcomings. For instance, the time series data must be stable when these models are used for forecasting. It is difficult to capture the characters if the datasets are nonstationary. Therefore, the model is less effective when applied for time series forecasting during periods of sharp fluctuations [14]. With the development of artificial intelligence, machine learning models, such as support vector machine (SVM) and artificial neural networks (ANNs), have attracted a lot of attention because of the learning capabilities for nonlinear

kernel mapping between input and output vectors. For instance, Huang et al. [7] explored the forecasting ability of SVM for financial movement direction and proposed a combining model based on SVM and classification methods. Ghiassi et al. [15] presented a dynamic neural network model for time series events prediction, and compared with the ARIMA model, the prediction results of the proposed model have higher accuracy. Liao and Wang [6] established an improved neural network, the stochastic time-effective neural network model, and analyzed the volatility statistics characteristics of the Chinese stock price indices. Wang and Wang [8] established a hybrid model by combining the principle component analysis (PCA) algorithm and random time-effective neural networks (STNN) and explored the predictive performance by considering financial time series. Although machine learning techniques have considerable prediction processing capacity, their precision on the correlations exploring between data is still not efficient. Meanwhile, these methods are extremely time-consuming for big data and predictions are not quite expected [16]. With the establishment of the hidden layer units, the transmission of historical information can be realized by recurrent neural networks (RNNs). Wang and Wang [9] proposed a new forecasting model to elevate the prediction accuracy of crude oil price fluctuations, which is based on multilayer perceptrons (MLP) and Elman recurrent neural networks (ERNN) with stochastic time effective function. Berradi and Lazaara [17] combined principal component analysis and RNNs to predict the stock price from Casablanca Stock Exchange, and the results enhanced the accuracy of the original method and performed a desirable prediction for the stock price. Deep learning methods are the broader series of machine learning methods, which try to learn advanced features from the given data. Compared with traditional neural network models, deep learning methods contain multiple hidden layers of multilayer perceptrons, and they have better performances in managing strong nonlinear characteristics. Long short-term memory network (LSTM) is a type of deep learning method devised to deal with the long-term dependence problems for a special purpose [18]. The network structure of LSTM is much more complex than that of RNNs, which utilizes memory cell states to maintain essential historical information and get rid of the unimportant. Due to the superior algorithm mechanism, LSTM is widely applied to natural language processing (NLP) and sentimental analysis [19, 20], time series forecasting [10, 21, 22], and synthesizing a piece of music [23]. However, the individual forecasting models cannot precisely reveal the complicated connections existing in the nonlinear and nonstationary datasets.

To obtain more accurate and reliable time series prediction, different kinds of hybrid forecasting models have been proposed which could take the advantage of different single models [24–26]. Among them, the hybrid models based on decomposition and prediction have been widely recognized, and such models are usually composed of nonlinear decomposition method and forecasting model. Liu et al. [27] presented an improved hybrid forecasting model for wind speed, which includes the empirical wavelet

transform method and three types of deep learning networks. By comparing all the data results of different methods, the proposed reinforcement learning based hybrid model is effective in combining three types of deep learning networks and performs better than conventional optimization-based hybrid models. Wang and Wang [28] combined empirical mode decomposition (EMD) method with random time strength neural network to predict global stock indices, and the empirical results showed that the proposed approach veritably has a great effect in predicting stock market fluctuations. Wang et al. [29] established a two-layer decomposition model and then developed an ensemble approach by integrating the fast ensemble empirical mode decomposition method (FEEMD), variational mode decomposition (VMD), and optimized backpropagation neural network by firefly algorithm (FA-BPNN). The empirical results indicated that the developed new model has exceptional forecasting implementation in electricity price series. The first key point of hybrid models is to break down the original data series into several independent subseries and makes it likely for models to adaptively learn the nonlinear characteristics of fluctuations in each subseries. Then, by using the inverse transformation algorithm, the forecasting series of each subseries are integrated to acquire the final forecasting results. These hybrid models could raise the efficiency and precision of modelling by conquering the handicap of nonlinear and nonstationary of original series [30–32]. The empirical results show that wavelet transform (WT) is a time-frequency localization analysis method in which the window area is fixed but its shape can be changed. Because it only recombines low-frequency signals during the decomposition process, and no longer breaks down high-frequency signals, its frequency resolution decreases as the frequency increases. The EMD, FEEMD, and VMD methods also have some certain limitations, for example, inadequate mathematical explanations, the boundary effects, noise oversensitivity, and pattern overlap. These may cause excessive decomposition of the original data and adversely affect the prediction results [33, 34]. On the other hand, the well-known deep learning model causes overfitting problems and is always based on historical information without thinking over the statistical regularity of behavior in the financial market, which leads to deficient precision [10, 32].

To improve the disadvantages of the above widely recognized decomposition methods and the traditional deep learning methods, this paper proposes a novel ensemble energy forecasting framework, WPD-SW-LSTM, which combines wavelet packet decomposition (WPD), the stochastic time strength weights (SW) method, and LSTM. The WPD is proposed on the basis of the issue that the inferior frequency resolution of wavelet decomposition in the high-frequency range and poor time resolution in the low-frequency range. It is a more sophisticated method of signal analysis to improve the temporal resolution signal. Moreover, the WPD working speed is faster than the traditional WT, and by selecting the appropriate wavelet basis function and mother function, the mixing-frequency problem can be improved. Therefore, WPD is adopted in this research to explore the complexity of nonlinear

characteristics for original energy future time series. In fact, there are complicated factors that affect energy futures prices in the process of market transactions fluctuations. SW is based on stochastic process which conforms with both the real trading market and the gating mechanism in the forecasting model [6, 8, 10]. The mechanism of SW is to measure historical information in conformity with the time of occurrence. The newer the historical data occurs, the more valuable its data information is to present future information, so that historical price figures can be employed to advanced pick up the fluctuations statistics in the energy futures series. In addition, this research employs the WPD method to extract the original crude oil series for the first time and firstly improves the conventional LSTM model with stochastic time strength weights for the crude oil prices forecasting. With the method of WPD, the original energy futures price series can be decomposed into several subseries ( $SS_i$ ), which are in different frequency bands. Then, different SW-LSTM models are modeled for the corresponding  $SS_i$ , respectively. Finally, the ensemble forecasting result of the original energy futures series is produced by integrating all the predicted  $SS_i$  components. To estimate the predictive power of the proposed model WPD-SW-LSTM, the conventional and latest hybrid models (SVM, BPNN, LSTM, WPD-BPNN, WPD-LSTM, CEEMAD-LSTM, VMD-LSTM, and ST-GRU) are introduced for comparative analysis. In order to reveal the predictive capabilities of different forecasting models, quantitative analysis is performed through different error methods. At the same time, this research proposes a new error measurement method called multiorder multiscale complexity invariant distance (MMCID) [9,35]. The main contributions of this paper are summarized as follows:

- (a) A novel hybrid forecasting model SW-LSTM is established for energy futures series, which based on the LSTM network and the theory of stochastic process.
- (b) Combined with WPD method, several subseries ( $SS_i$ ) with different fluctuation frequency are derived from the original data series. Each  $SS_i$  is trained by the new SW-LSTM model, respectively.
- (c) The empirical results of corresponding forecasting models are estimated and contrasted with different error criteria and the new measurement MMCID.

The structure of this article is as follows. Section 2 explains the price datasets from the energy futures markets. Section 3 introduces the WPD and SW-LSTM methodologies and provides the main framework of this paper. Section 4 demonstrates the experimental forecasting results in detail. Section 5 compares the proposed hybrid method with other models, which are SVM, BPNN, LSTM, WPD-BPNN, WPD-LSTM, CEEMAD-LSTM, VMD-LSTM, and ST-GRU. Moreover, error measurement methods are applied to estimate the prediction performance of each model in this section. Finally, Section 6 summarizes the main conclusion of this study.

## 2. Datasets

Crude oil is an international bulk financial commodity, which can be traded in markets around the world either through spot oil or through financial derivative contracts. This research mainly focuses on the oil futures market, and four representative oil futures indices are selected for the case study: west Texas intermediate (WTI) futures prices series, Brent crude oil futures prices series, RBOB gasoline, and heating oil. These four datasets are from the New York Mercantile Exchange (NYMEX) energy futures market, which can be downloaded from <https://www.wind.com.cn/>. WTI crude oil price is widely applied in the pricing of US domestic crudes. Brent is the theoretical international oil benchmark, and prices of most oil use Brent crude as the criterion, which connected with two-thirds of all the world's oil contracts. Brent crude and WTI dominate the oil market, and both determine pricing in their corresponding markets. They are known as light sweet oil because they contain low sulfur, making it "sweet," and have low density, making it "light." Gasoline and heating oil are refined from crude oil which are usually merchandised as futures contracts in financial markets. Figure 1 reveals the similar dynamic changes in more than a 10-year period from January 2, 2009, to October 23, 2019, of the four corresponding oil futures series. In the past decades, the price fluctuation trends of these four futures series are almost the same, which manifest that there is a certain correlation between them.

## 3. Methodology

*3.1. Wavelet Packet Decomposition.* Wavelet transform is a mathematical method produced to solve the problem of decomposition of nonstationary signals. Compared with wavelet analysis, wavelet packet decomposition (WPD) can be used to analyze the signal more meticulous. Wavelet packet analysis can divide the time-frequency plane in more detail, and the resolution of the high-frequency part of the signal is better than wavelet analysis [36]. It can also adaptively select the best wavelet basis function according to the characteristics of the signal in order to better analyze the signal. The theory of the WPD analysis is as follows [37–39]. The wavelet packet function is a time-frequency function; it can be defined as

$$W_{j,k}^n(t) = 2^{(j/2)} W^n(2^j t - k), \quad (1)$$

where the integers  $j$  and  $k$  are the index scale and translation operations. The index  $n$  is an operation modulation parameter or oscillation parameter. The first two wavelet packet functions are the scaling and mother wavelet functions:

$$\begin{aligned} W_{0,0}^0(t) &= \phi(t), \\ W_{0,0}^1(t) &= \psi(t). \end{aligned} \quad (2)$$

When  $n = 2, 3, \dots$ , the function has the following recursive relationship:



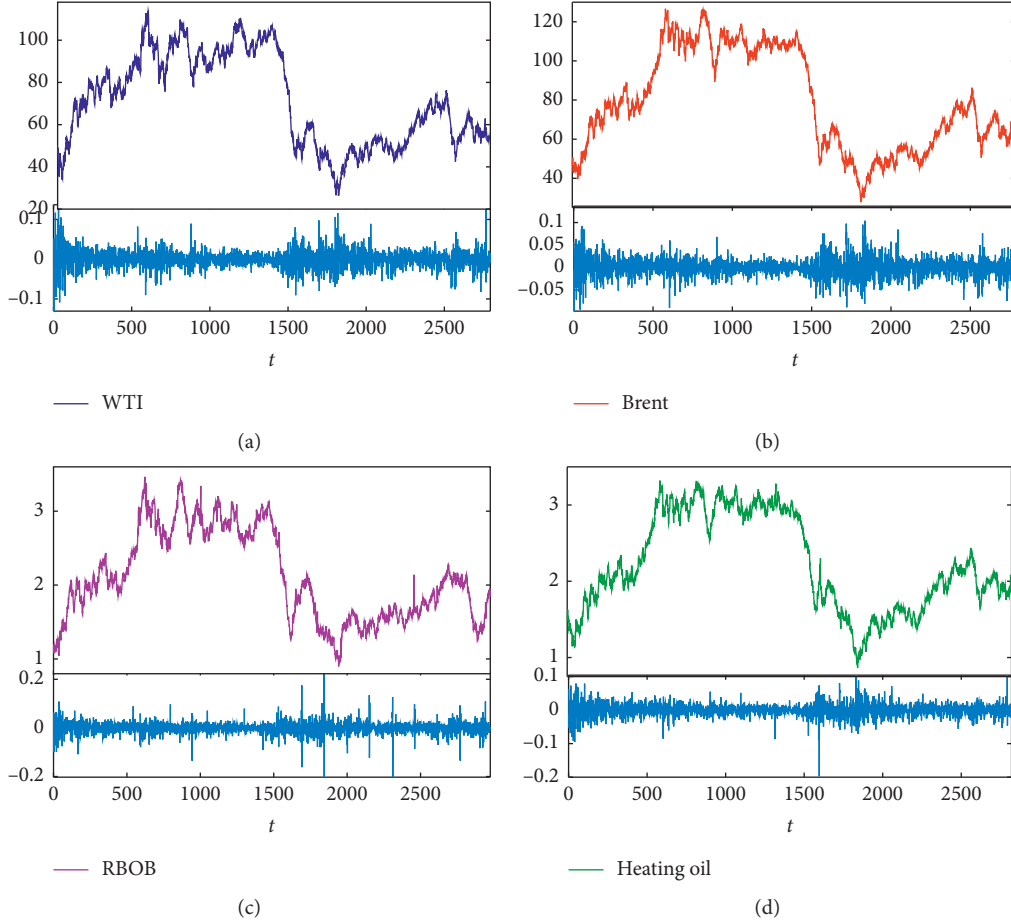


FIGURE 1: Dynamic changes of energy futures series between January 2, 2009, and October 23, 2019.

$$\begin{aligned}
 W_{0,0}^{2n}(t) &= \sqrt{2} \sum_k h(k) W_{1,k}^n(2t-k), \\
 W_{0,0}^{2n+1}(t) &= \sqrt{2} \sum_k g(k) W_{1,k}^n(2t-k),
 \end{aligned} \tag{3}$$

where  $h(k)$  and  $g(k)$  are the quadrature filter function related to the previously defined scaling function and mother wavelet function. The wavelet packet coefficients  $w_{j,k}^n$  are calculated by the inner product  $\langle f(t)W_{j,k}^n \rangle$ , which is defined as

$$w_{j,k}^n = \langle f(t)W_{j,k}^n \rangle = \int f(t)W_{j,k}^n dt. \tag{4}$$

According to the literature [40], the number of the decomposition level is often in the range from 2 to 4 in forecasting model. In the present work, the 3-level framework of WPD algorithm is applied, which is schematically shown in Figure 1(a). Additionally, the Daubechies wavelets of order 4 are employed as the mother wavelet in this research [41], and the corresponding decomposition result of the WTI crude oil is demonstrated in Figure 2(b). Each subseries with different frequency band represents a sort of oscillatory factor embedded in the futures price indices. In Figure 2(b), the decomposed subseries “DDD3,” “DDA3,”

“DAD3,” “DAA3,” “ADD3,” “ADA3,” “AAD3,” “AAA3” are recorded as  $SS_i$  ( $i = 1, 2, \dots, 8$ ) series subsequently.

**3.2. Long Short-Term Memory Network.** Long short-term memory networks are a particular form of RNNs that can handle with long-term and short-term dependencies. They were introduced in 1997 by Hochreiter and Schmidhuber [18] and were improved and promoted in subsequent work. Although the structure of traditional RNNs are entirely component of handling long-term memory dependencies in theory, the effect is confined in the actual application [42]. Therefore, the memory storage capacity of RNNs is more suitable for short-term sequences. On the basis of conventional RNNs, cell states and gate mechanism are added to the hidden layer, so that the gradient vanishing problem can be largely mitigated through its control gates. In addition, each time the historical message is dispatched to the neurons of the hidden layer, several control gates with different functions are employed to regulate the information of the past and latest. The principle of the control gate is described as follows. It is mainly composed of a sigmoid neural net layer and a pointwise multiplication operation. The output values of sigmoid function stage are between 0 and 1, which indicate how much information can be delivered to the next step. A value of zero means

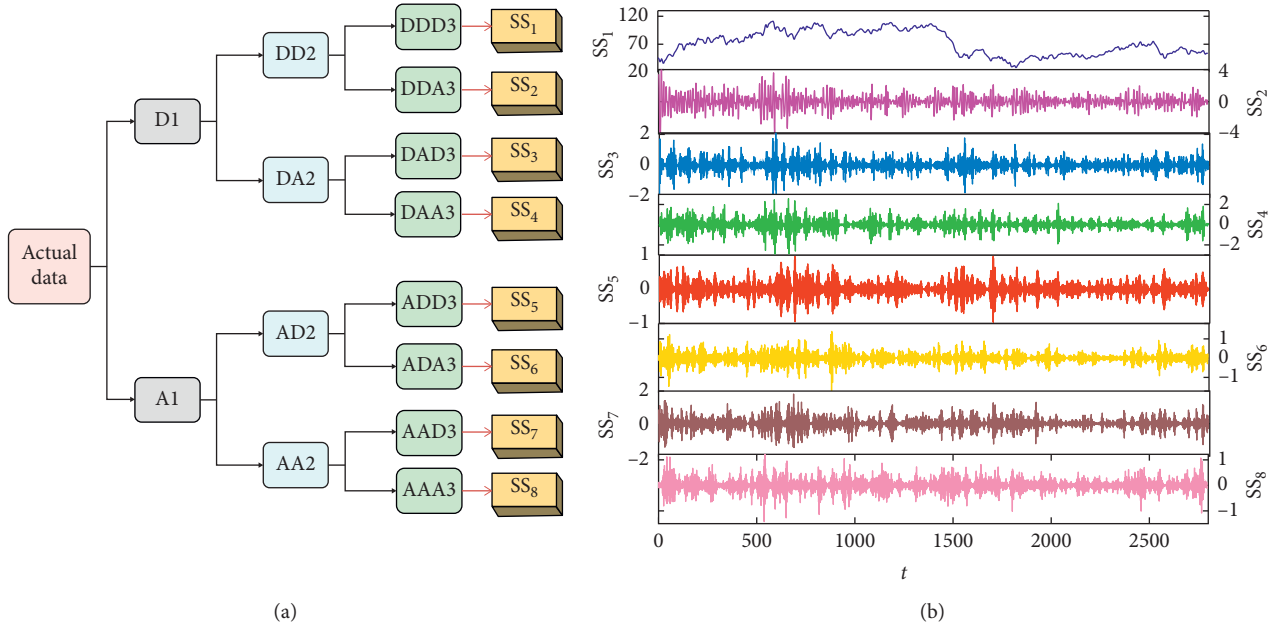


FIGURE 2: (a) The process of WPD algorithm. (b) The corresponding subseries  $SS_i$  of WTI index derived from WPD.

letting nothing through, while a value of one means letting everything through. Specially, when the value is 0, it means nothing can be transmitted, and when the value is 1, it implies everything can be transmitted. The LSTM control gates involve three gates: the forget gate  $f_t$ , the input gate  $i_t$ , and the output gate  $o_t$ . The forget gate determines how much historical information stored in the current moment from the last moment. The input gate judges the information saved in the cell state, and the output gate decides the output data based on the cell state. The architecture of LSTM network is shown in Figure 3. The description of LSTM networks follows Fischer and Krauss [43], Sainath et al. [44], and He et al. [45]. The specific algorithm steps of LSTM are as follows:

- (i) The memory cell reads in the input  $x_t$  and the previous hidden state  $h_{t-1}$ , which can reveal long-term dynamic trends and abandon the redundant useless information. The forget gate is determined by the following equation:

$$f_t = \sigma\{W_f \cdot (x_t, h_{t-1}) + b_f\}. \quad (5)$$

- (ii) The first part of input gate in the model determines how much current information should be retained in the cell state:

$$i_t = \sigma\{W_i \cdot (x_t, h_{t-1}) + b_i\}. \quad (6)$$

- (iii) The second part is to generate a new candidate vector  $\tilde{C}_t$  to update the state, which is according to the following equation:

$$\tilde{C}_t = \tanh\{W_C \cdot (x_t, h_{t-1}) + b_C\}. \quad (7)$$

- (iv) After that, the new cell state  $C_t$  is constructed on the basis of the outcomes of the last steps with  $\otimes$  denoting the Hadamard (element-wise) product:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t. \quad (8)$$

- (v) Finally, the output gate  $o_t$  is updated and the final output  $h_t$  is decided based on the updated state and the output gate state:

$$\begin{aligned} o_t &= \sigma\{W_o \cdot (x_t, h_{t-1}) + b_o\}, \\ h_t &= o_t \otimes \tanh(C_{t-1}). \end{aligned} \quad (9)$$

In the previous equations, the following notation is used:

- (i)  $x_t$  is the input vector at current time step  $t$ .
- (ii)  $W_f, W_i, W_C,$  and  $W_o$  are the weight matrices which associate with corresponding vectors. They can be split into

$$\begin{cases} W_f = W_{fx} + W_{fh}, \\ W_i = W_{ix} + W_{ih}, \\ W_C = W_{Cx} + W_{Ch}, \\ W_o = W_{ox} + W_{oh}. \end{cases} \quad (10)$$

- (iii)  $b_f, b_i, b_C,$  and  $b_o$  are bias indicators.
- (iv)  $f_t, i_t,$  and  $o_t$  are forget gate, input gate, and output gate vectors.
- (v)  $C_t$  and  $\tilde{C}_t$  are vectors for the cell states and candidate values.
- (vi)  $h_t$  is a vector for the output of the LSTM layer.

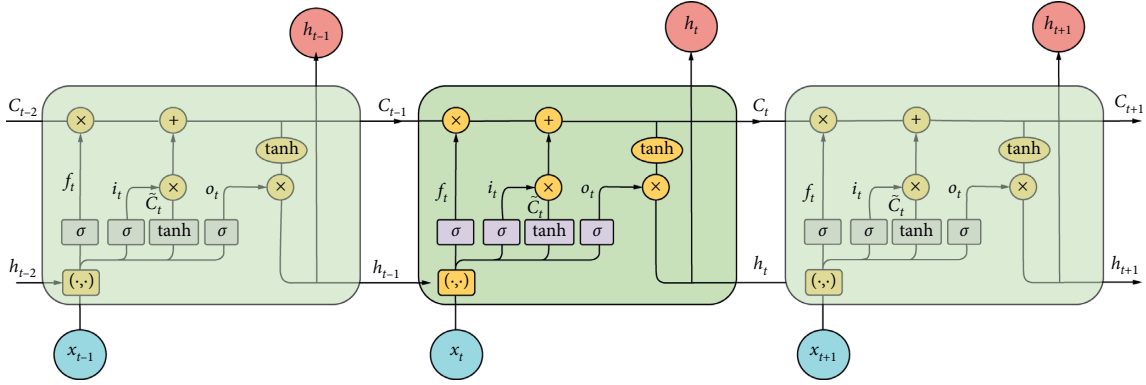


FIGURE 3: The architecture of LSTM network.

- (vii)  $\sigma(\cdot)$  and  $\tanh(\cdot)$  are the sigmoid function and hyperbolic tangent function, respectively.

**3.3. LSTM with Stochastic Time Effective Weight Function (SW-LSTM).** Dufresne and Gatheral et al. [46, 47] demonstrate that the prediction of financial market price series should integrate great amount of historical data, because the information represented in different periods has different impacts on future results. In other words, the closer the data is to the current time, the stronger the impact of information is at that moment, and, on the contrary, the further the data is, the weaker the influence is [48]. Therefore, to improve the accuracy of forecasting in actual application, this paper considers combining the SW function with LSTM theory in the predictive modelling process. During the stage of model training, SW function is integrated into the LSTM model to construct a novel forecasting model, which is referred to as long short-term memory with stochastic time strength weight function model (SW-LSTM). The expression of SW function derives from a stochastic process [6]. It can assign different weights to different data in the light of the variant time of occurrence. The mathematical expression is as follows:

$$\varphi(t_n) = \frac{1}{\beta} \exp \left\{ \int_{t_0}^{t_n} \mu(t) dt + \int_{t_0}^{t_n} \omega(t) dB(t) \right\}, \quad (11)$$

where  $\beta (> 0)$  is the depth of market parameter,  $t_0$  is the moment of the latest time point in the data set, and  $t_n$  is an

arbitrary time point in the dataset.  $B(t)$  is the standard Brownian motion which is commonly considered as random movement of a particle in liquid [49].  $\mu(t)$  is the drift function which mainly direct trend changes.  $\omega(t)$  is the wave function which is applied to model the uncertain events during the forecasting process. The mathematical expression of  $\mu(t)$  and  $\omega(t)$  is as follows:

$$\begin{aligned} \mu(t) &= \exp(-\alpha t), \\ \omega(t) &= \omega(T) = \frac{1}{T-1} \left( \sum_{i=1}^T (x_i - \bar{x})^2 \right)^{(1/2)}. \end{aligned} \quad (12)$$

In the training process of conventional LSTM network, the parameter matrices  $W_f$ ,  $W_i$ ,  $W_c$ , and  $W_o$  are modified following the backpropagation in each iteration through time procedure of typical RNNs [17]. The model training error of the sample point  $n$  is defined as

$$E(t_n) = \frac{1}{2} \varepsilon_{t_n}^2 = \frac{1}{2} (d_{t_n} - y_{t_n})^2. \quad (13)$$

For the SW-LSTM model, a new description of model training error  $E_{tn}$  can be obtained:

$$E(t_n) = \frac{1}{2} \varphi(t) \varepsilon_{t_n}^2 = \frac{1}{2} \varphi(t) (d_{t_n} - y_{t_n})^2. \quad (14)$$

Then, the corresponding global error of model training is defined as

$$E = \frac{1}{N} \sum_{i=1}^N E_i = \frac{1}{2N} \sum_{i=1}^N \frac{1}{\beta} \exp \left\{ \int_{t_0}^{t_n} \mu(t) dt + \int_{t_0}^{t_n} \omega(t) dB(t) \right\} (d_{t_n} - y_{t_n})^2. \quad (15)$$

In the modelling process, based on the newly defined global error  $E$ , the model parameters are updated through the gradient descent method [10, 50, 51]. First, the partial

derivative of each model parameter needs to be calculated from the global error function. Then, the principle of parameter update is as follows:

$$\begin{aligned}
\frac{\partial E}{\partial W_{fx}} &= \frac{\partial E}{\partial \text{net}_{f,t}} \frac{\partial \text{net}_{f,t}}{\partial W_{fx}} = \delta_{f,t} \varphi(t) x_t, \\
\frac{\partial E}{\partial W_{fh}} &= \sum_{j=t_0}^{t_n} \delta_{f,j} \varphi(t) h_{j-1}, \\
\frac{\partial E}{\partial W_{ix}} &= \frac{\partial E}{\partial \text{net}_{i,t}} \frac{\partial \text{net}_{i,t}}{\partial W_{ix}} = \delta_{i,t} \varphi(t) x_t, \\
\frac{\partial E}{\partial W_{ih}} &= \sum_{j=t_0}^{t_n} \delta_{i,j} \varphi(t) h_{j-1}, \\
\frac{\partial E}{\partial W_{Cx}} &= \frac{\partial E}{\partial \text{net}_{C,t}} \frac{\partial \text{net}_{C,t}}{\partial W_{Cx}} = \delta_{C,t} \varphi(t) x_t, \\
\frac{\partial E}{\partial W_{Ch}} &= \sum_{j=t_0}^{t_n} \delta_{C,j} \varphi(t) h_{j-1}, \\
\frac{\partial E}{\partial W_{ox}} &= \frac{\partial E}{\partial \text{net}_{o,t}} \frac{\partial \text{net}_{o,t}}{\partial W_{ox}} = \delta_{o,t} \varphi(t) x_t, \\
\frac{\partial E}{\partial W_{oh}} &= \sum_{j=t_0}^{t_n} \delta_{o,j} \varphi(t) h_{j-1}, \\
\frac{\partial E}{\partial b_f} &= \sum_{j=t_0}^{t_n} \delta_{f,j} \varphi(t), \\
\frac{\partial E}{\partial b_i} &= \sum_{j=t_0}^{t_n} \delta_{i,j} \varphi(t) \frac{\partial E}{\partial b_C} = \sum_{j=t_0}^{t_n} \delta_{C,j} \varphi(t), \\
\frac{\partial E}{\partial b_o} &= \sum_{j=t_0}^{t_n} \delta_{o,j} \varphi(t),
\end{aligned} \tag{16}$$

where  $\text{net}_{f,t}, \text{net}_{i,t}, \text{net}_{C,t}, \text{net}_{o,t}$  denotes the input of the corresponding function,  $\delta_{f,t} = (\partial E / \partial \text{net}_{f,t})$ ,  $\delta_{i,t} = (\partial E / \partial \text{net}_{i,t})$ ,  $\delta_{C,t} = (\partial E / \partial \text{net}_{C,t})$ , and  $\delta_{o,t} = (\partial E / \partial \text{net}_{o,t})$ .

The above is the algorithm of SW-LSTM model, which corrects the model parameters accords with the gradient descent method. Figure 4 illustrates the training algorithm procedures of the proposed model, which involve six steps. For the different subseries of different crude oil series, different hyperparameters, which include the training steps, the number of hidden layers units, the learning rate, number of iterations, and the batch size, should be trained by the proposed model. The specific modelling and empirical prediction are given in Section 4.

### 3.4. Forecasting Process of the Hybrid WPD-SW-LSTM Model.

In this study, the fluctuation of energy futures prices is applied to the proposed hybrid forecasting model, WPD-SW-LSTM. The procedure of the WPD-SW-LSTM approach is described in brief subsequently, and the flowchart of this

research is shown in Figure 5. Firstly, the main process of the proposed model is displayed on the upper left of Figure 5, which includes three steps. The first step is data decomposition, where the original preprocessed data are decomposed by WPD method. Then, applying the improved SW-LSTM method for subseries forecasting step, the third step is the ensemble forecasting step. Then, the final forecasting results can be obtained by aggregating the subseries forecasting results with inverse wavelet packet transform. The specific description of each step is as follows:

Step 1: the WPD technique is employed to analyze the original energy futures series  $X(t) (t = 1, 2, \dots, N)$ . And, 8 subseries  $SS_i, i = 1, 2, \dots, 8$  are derived from the three-layer WPD method, which indicate that the local oscillations in different frequency bands. The details of the WPD algorithm are given in Section 3.1.

Step 2: each subsequence  $SS_i$  derived from WPD method is separated into training and testing datasets. The SW-LSTM network is utilized to train and establish

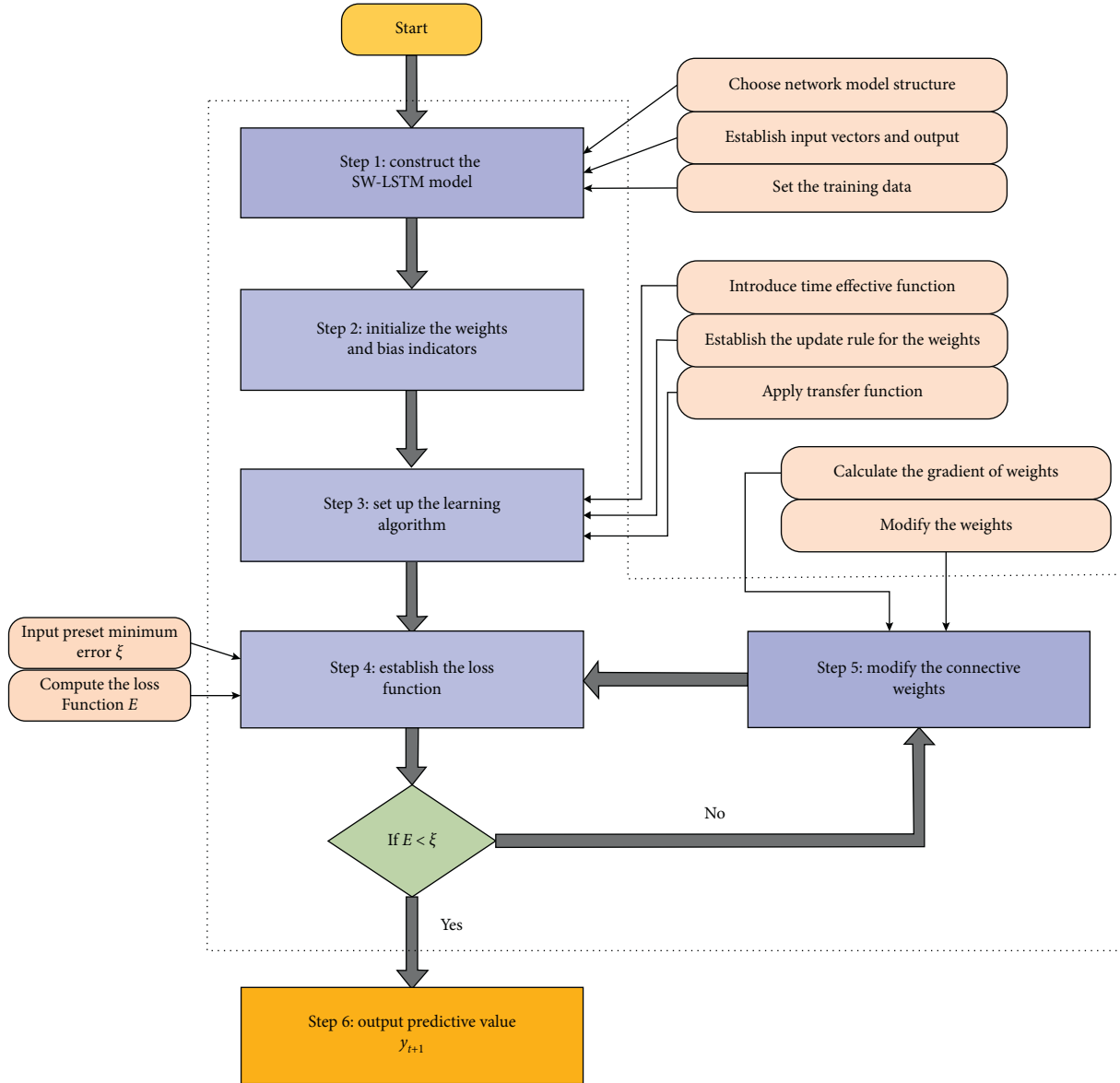


FIGURE 4: The training algorithm procedures of SW-LSTM model.

the forecasting model on the basis of the training dataset. Model parameters need to be set in advance, which includes the learning rate, the number of hidden layer units, the number of iterations, and the batch size. They are essential for predicting precision of the model. The training algorithm procedures of SW-LSTM model are proposed in Sections 3.2 and 3.3.

Step 3: it composites the prediction of each  $SS_i$  to obtain the final forecasting results by employing the theory of inverse wavelet packet transform. Moreover, linear regression and relative error are applied to investigate the correlation between predictive points and actual values.

Step 4: multiple evaluation indicators are adopted to estimate the prediction ability of WPD-SW-LSTM, which involves MAE, RMSE, MAPE, SMAPE, and TIC and a novel method multiple multiorder complexity-

invariant distance (MMCID) based on information theory. In addition, other models like SVM, BPNN, LSTM, WPD-BPNN, and WPD-LSTM are taken into account for prediction comparison.

## 4. Forecasting and Statistical Analysis

**4.1. Data Preprocessing.** To estimate the performance of the proposed WPD-SW-LSTM forecasting model, the futures prices of WTI crude oil, Brent crude oil, RBOB gasoline, and heating oil are selected. Table 1 displays the selected data sets of all indices that are from 02/01/2009 to 23/10/2019. Usually, the non-trading days are regarded as frozen such that this research only adopts the data during trading time. To conduct the experiments, nearly eighty percent of the samples from 2009 to 2017 are used to train the model, and the remaining twenty percent of data are used for

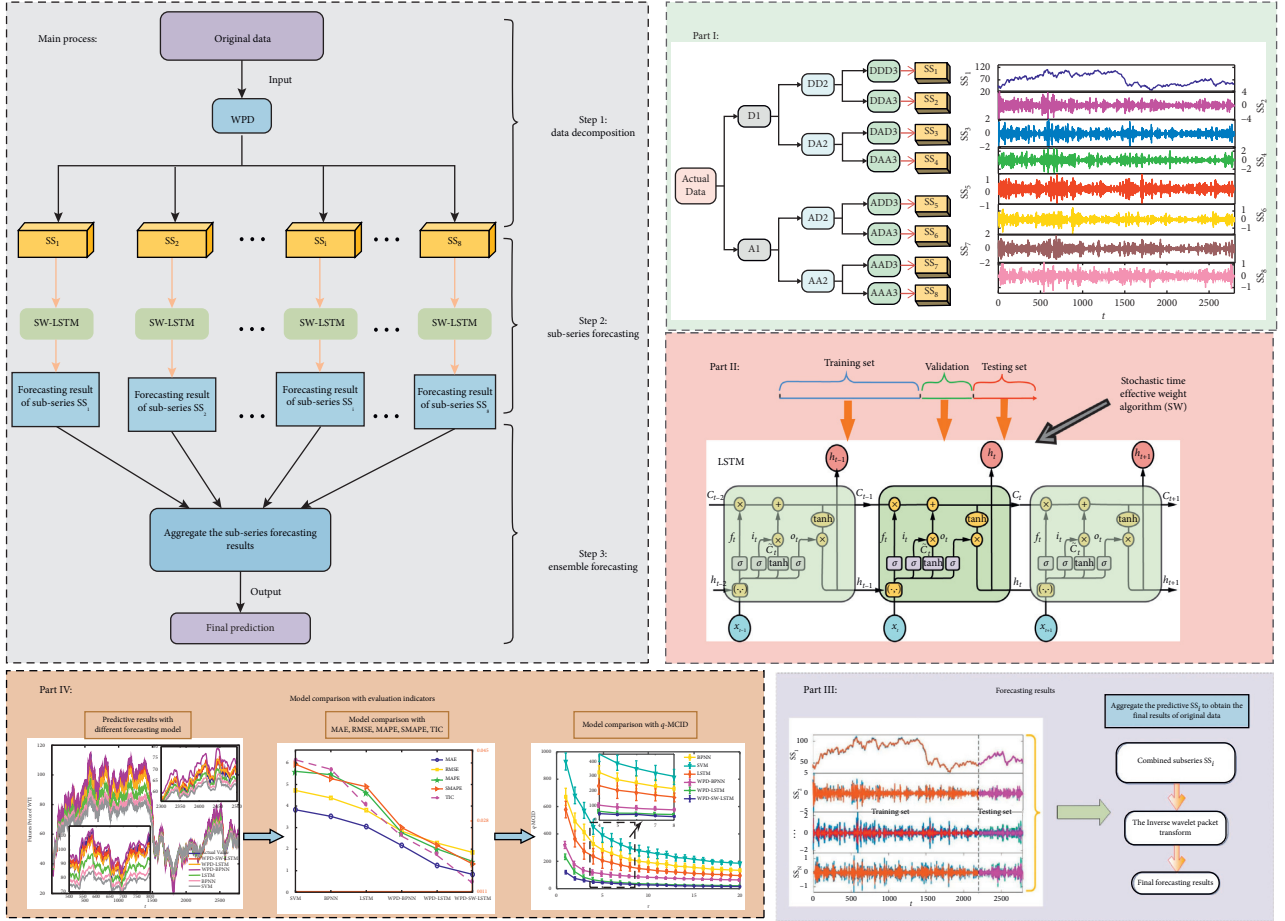


FIGURE 5: Flowchart of the hybrid WPD-SW-LSTM model.

TABLE 1: Data selection.

Index	Data sets	Total number	Training sets	Training number	Testing number
WTI	02/01/2009 ~ 23/10/2019	2794	02/01/2009 ~ 31/08/2017	2230	564
Brent	02/01/2009 ~ 23/10/2019	2791	02/01/2009 ~ 21/08/2017	2230	561
RBOB	02/01/2009 ~ 23/10/2019	2976	02/01/2009 ~ 20/11/2017	2380	570
Heating oil	02/01/2009 ~ 23/10/2019	2821	02/01/2009 ~ 22/08/2017	2250	571

Note: training number means the number in training set; testing number represents the number in testing set.

testing to examine the effectiveness of the proposed model. Table 1 provides the selection and division of the four selected oil futures indices. Generally, to minimize the influence of noise and finally enhance the accuracy of forecasting, each subseries  $SS_i$  derived from WPD is normalized to the range of  $[0, 1]$  by the following standardized method [52, 53]:

$$S(t)' = \frac{S(t) - \min S(t)}{\max S(t) - \min S(t)}. \quad (17)$$

After that, to acquire the true predictive value and then intuitively compare the numerical results with the actual value, the normalized output variables  $S(t)$  should be reverted to  $S(t)$  as follows:

$$S(t) = S(t)' (\max S(t) - \min S(t)) + \min S(t). \quad (18)$$

**4.2. Training and Forecasting by the Hybrid WPD-SW-LSTM Model.** In this section, four different energy futures price series are carried out to support the proposed hybrid WPD-SW-LSTM model. The decomposition merit of WPD makes it exceptional in the extraction of feature sequences. The model parameters are trained by calculating the root mean square error between the predicted value and actual value. The global error between the predicted value and the actual target is reduced through weights modification. The training enters the next step when the global error is less than the preset value. For all prediction models involved in this article, the input units are set to 4, and the output units are set to 1. In WPD-SW-LSTM model, the batch size is set to 32, the hidden size is 30, and the epochs number is 400.

Afterwards, the normalized subseries  $SS_i$  obtained from WPD are trained and predicted by the SW-LSTM model. The

number of input samples is set to 4, and the number of outputs is set to 1; that is, the 4th order historical data are used to predict the data of the next period. Figure 6 shows the forecasting results of each subseries from the futures series of WTI crude oil. It is shown visually that the predicted value of each subseries  $SS_i$  is almost consistent with the actual values. With the purpose of illustrating the prediction from the SW-LSTM forecasting model, Figure 7 demonstrates the empirical results of each subseries from RBOB gasoline. Figures 6 and 7 present decomposed forecasting results of WTI crude oil and RBOB gasoline as examples, which is a critical component that measures the fluctuations of the prediction, especially in forecasting the direction of fluctuations accurately. The subseries  $SS_i$  has been recognized as the whole trend of the futures price series, whose results from the proposed forecasting model are well predicted. The curves of the actual data and the predicted data intuitively are very approximating. Then, the final predictive results of the four sample datasets can be calculated by employing the theory of inverse wavelet packet transform.

Figure 8 shows the final predictive results for four indices, WTI, Brent, heating oil, and RBOB, with the proposed WPD-SW-LSTM model. From this figure, the fluctuation trends of the predictive data are extremely near that of the actual data. In addition, the absolute correlation error results of the empirical analysis are also revealed in Figure 7, which can be calculated by  $RE(t) = |\hat{y}_t - y_t|/y_t$ . It can be concluded that the predicted results nearly have consistent trends with the fluctuations of the actual data. The results of RE are also centralized in  $(0, 0.01)$ , and only a few sectional data points surpass 0.01 and are smaller than 0.015. It means that with repeated experiments, the energy futures series have been trained excellently, and the forecasting performance of the WPD-SW-LSTM model is improving.

It is generally known that the predicted results and the actual value can be fitted by linear regression method, where the predicted points are regarded as the dependent variable  $Y$ , and the actual data are considered as the independent variable  $X$ . Through linear regression analysis between the predicted value of the WPD-SW-LSTM model and the actual data, the prediction accuracy can be judged by the goodness of fit. The closer the goodness of fit value is to 1, the closer the predicted value is to the true value. An effective numerical indicator between the two variables is the correlation coefficient  $R$ . The curves of linear regression for series WTI, Brent, heating oil, and RBOB are revealed, respectively, in Figure 9, and the numerical results are revealed in Table 2. In detail, the values of  $R$  for these four series are all above 0.98, and the regression coefficients  $a$  of the linear equations are near to 1, which indicates that the predicted values are almost close to the actual values. The regression equation parameters of the proposed model for WTI are  $a = 0.9934, b = 0.6931$ , which is approaching to the ideal situation  $y = x$ , followed by the Brent indices,  $a = 0.9217, b = 4.864$ . The heating oil is  $a = 0.9441, b = 0.0823$  and RBOB gasoline is  $a = 0.9930, b = 0.0007$ .

## 5. Models Comparison and Prediction Accuracy Evaluation

**5.1. Performance Evaluation Criteria.** While the established model WPD-SW-LSTM is utilized to the forecasting experiments, it is also indispensable to validate the forecasting effects of different models. Then, five models (SVM, BPNN, LSTM, WPD-BPNN, and WPD-LSTM) are employed to the forecasting evaluations in this part. Support vector machine (SVM) technique is displayed in this part, which is regarded as the state-of-the-art machine learning theory for binary classification [54–56]. Additionally, to fully prove the effectiveness of the proposed model, BPNN, LSTM, and WPD-BPNN are selected to make a comparison because the proposed model is constructed based on LSTM network, and backpropagation neural network (BPNN) is the most typical neural network. For the purpose of estimating the forecasting error of the new hybrid model and comparing it with other five models, the error measurement between actual data points and predicted value for different models are investigated. Among them, mean absolute error (MAE), root mean square error (RMSE), mean absolute percent error (MAPE), symmetric mean absolute percent error (SMAPE), and Theil inequality coefficient (TIC) are selected as the error evaluation criteria, which can indicate the forecasting performance of each model. Generally, the smaller the error (MAE, RMSE, MAPE, SMAPE, and TIC) values are, the more accurate the predictive ability of the forecasting model is [52]. The evaluation definitions are expressed as follows:

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{t=1}^N |y_t - \hat{y}_t|, \\ \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2}, \\ \text{MAPE} &= 100 \times \frac{1}{N} \sum_{t=1}^N \left| \frac{y_t - \hat{y}_t}{y_t} \right|, \\ \text{SMAPE} &= 100 \times \frac{2}{N} \sum_{t=1}^N \frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|}, \\ \text{TIC} &= \frac{\sqrt{1/N \sum_{t=1}^N (y_t - \hat{y}_t)^2}}{\sqrt{1/N \sum_{t=1}^N y_t^2} + \sqrt{1/N \sum_{t=1}^N \hat{y}_t^2}}, \end{aligned} \quad (19)$$

where  $y_t$  and  $\hat{y}_t$  are the actual value and the predicted value at time  $t$ , respectively, and  $N$  is the total number of the data.

Figure 10 illustrates the forecasting results of WTI, Brent, RBOB, and heating oil for the six forecasting models in comparison. Additionally, the forecasting results from the insert plots of Figure 10 show the local prediction of training sets and testing sets from the proposed WPD-SW-LSTM

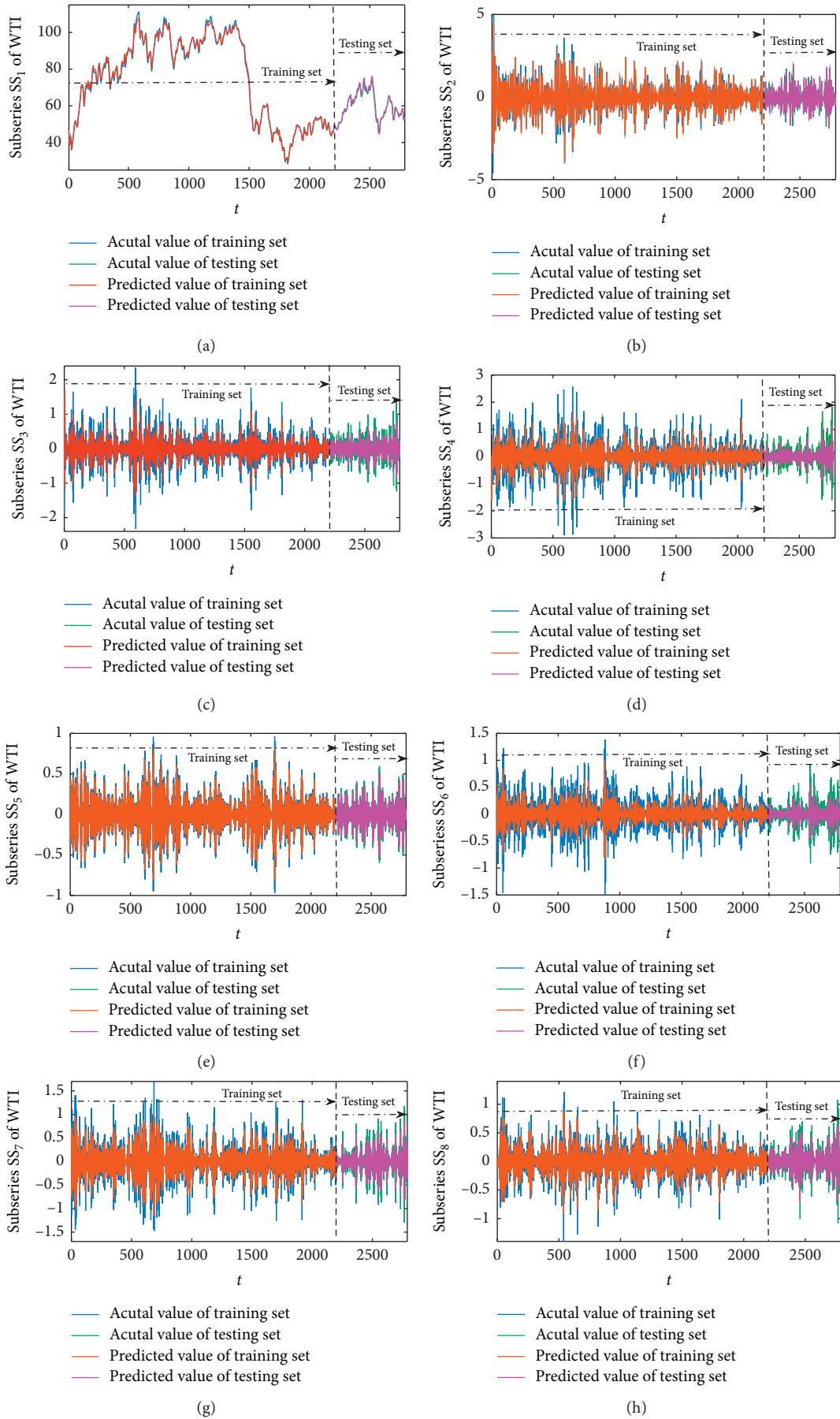


FIGURE 6: The predicted data and the actual data of each subseries from WTI crude oil. (a)  $SS_1$ . (b)  $SS_2$ . (c)  $SS_3$ . (d)  $SS_4$ . (e)  $SS_5$ . (f)  $SS_6$ . (g)  $SS_7$ . (h)  $SS_8$ .



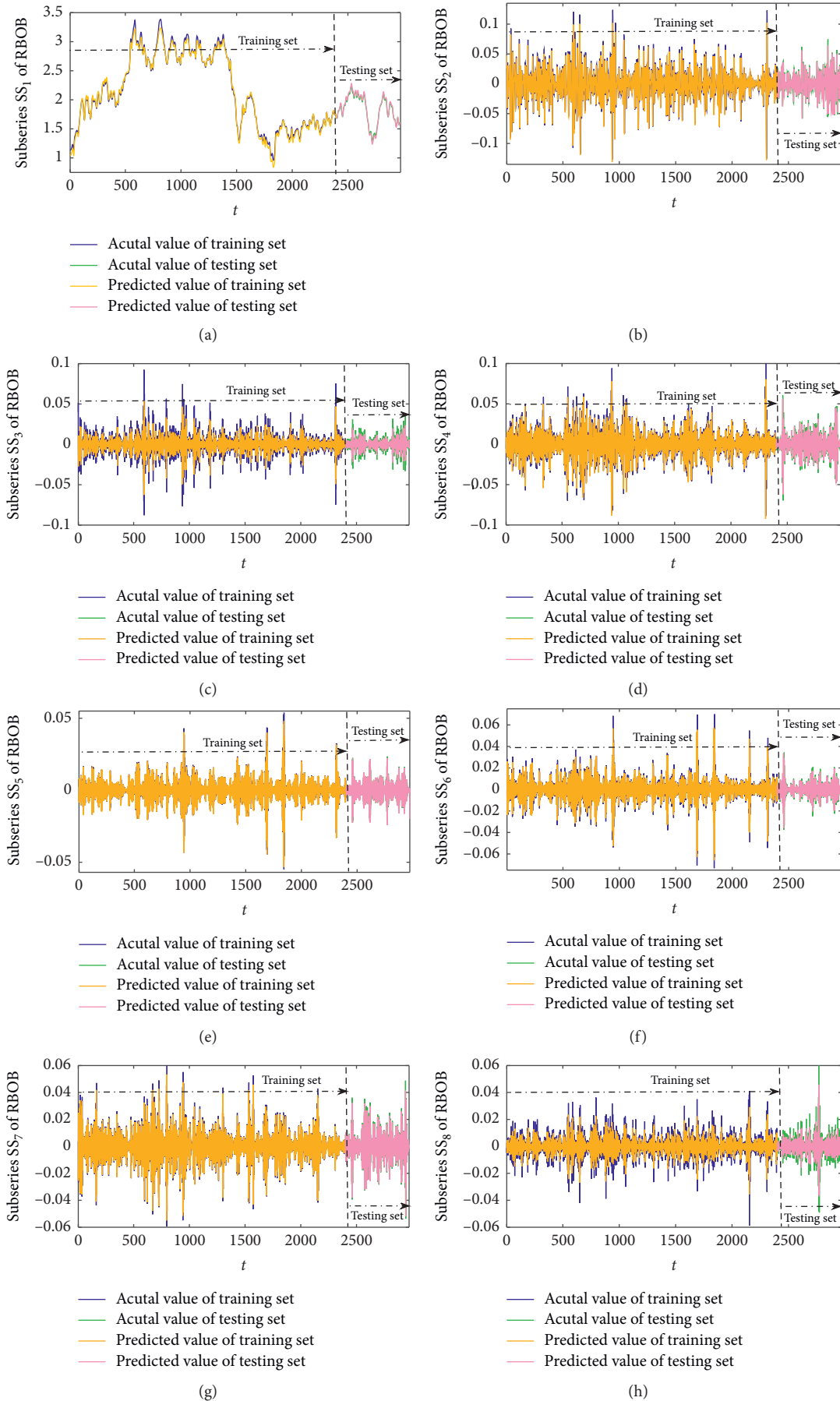


FIGURE 7: The predicted data and the actual data of each subseries from RBOB gasoline. (a)  $SS_1$ . (b)  $SS_2$ . (c)  $SS_3$ . (d)  $SS_4$ . (e)  $SS_5$ . (f)  $SS_6$ . (g)  $SS_7$ . (h)  $SS_8$ .

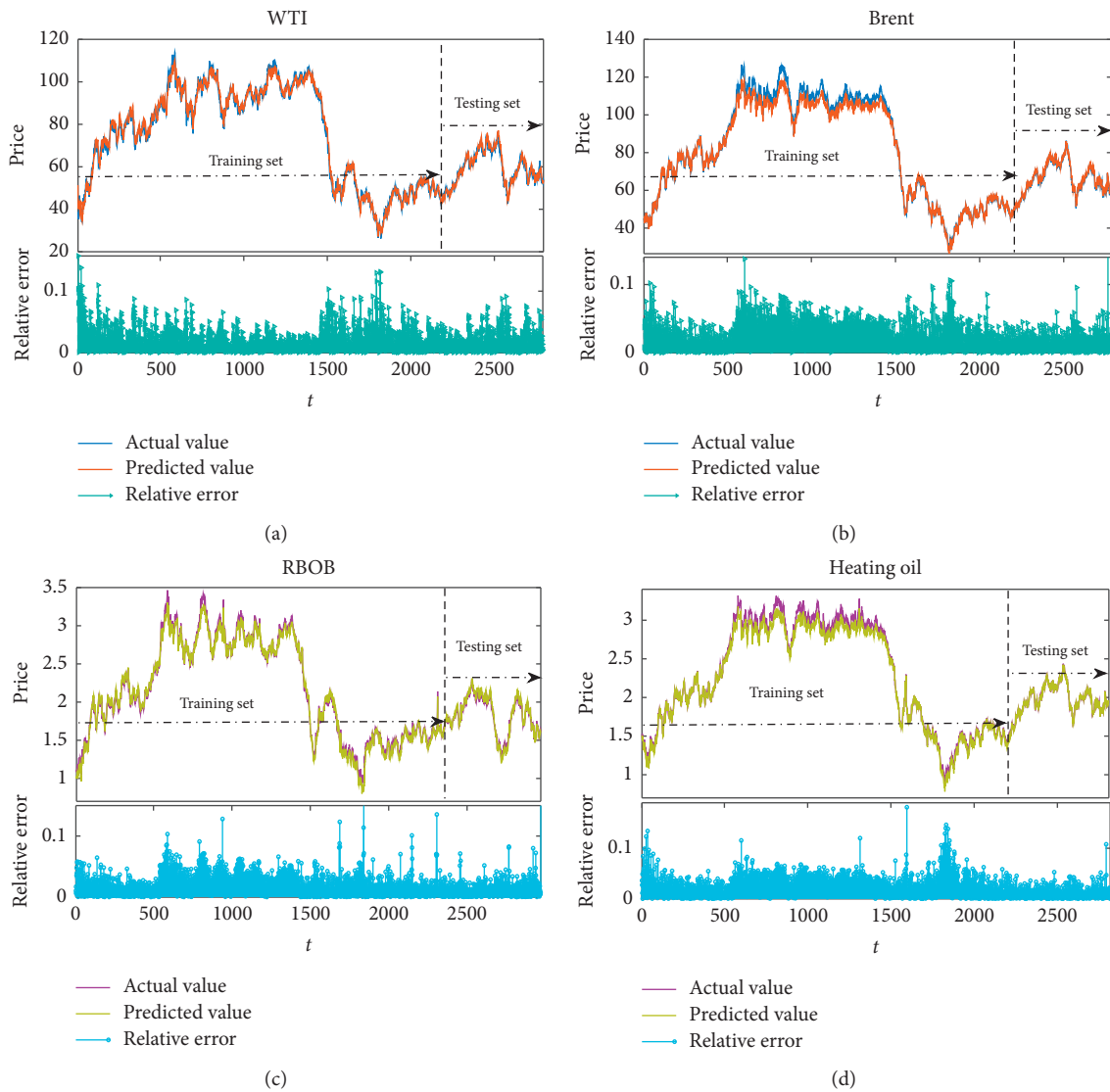


FIGURE 8: The predicted results of the proposed model for the original series. (a) WTI. (b) Brent. (c) RBOB. (d) Heating oil.

model, respectively. It displays the distinct advantages contrast with the other five models, SVM, BPNN, LSTM, WPD-BPNN, and WPD-LSTM, especially at big fluctuation stages. Affected by the changes of social economy and various external environment, the energy market shows different fluctuations. Besides, the predicted results during the small fluctuation period seem comparatively accurate for all predictive models.

Tables 3–6 demonstrate a detailed comparison of the evaluation criteria quantitatively, by applying MAE, RMSE, MAPE, SMAPE, and TIC among aforementioned six models. The numerical results demonstrate that the evaluation indicators from the WPD-SW-LSTM model are all the smallest ones among these models, and the evaluation indicators by the hybrid models are almost less than those by the individual models. For example, the MAPE values for WTI futures indices from the first three hybrid models are 1.4329, 2.0092, and 2.7653, and the individual models MAPE values are 4.6351, 5.4562, and 5.6108, respectively. Overall,

the empirical results demonstrate that the WPD-SW-LSTM predictor has higher forecasting accuracy. From the error evaluations, the hybrid models WPD-SW-LSTM, WPD-LSTM, and WPD-BPNN are superior to the LSTM, BPNN, and SVM models. Moreover, compared with the WPD-LSTM and WPD-BPNN model, the superior predictive accuracy of the proposed model WPD-SW-LSTM reflects that the stochastic time effective weights (SW) method can play an important role during forecasting process. In particular, after WPD-LSTM is combined with SW, the hyperparameters are extremely improved, and error indicators MAE, RMSE, MAPE, SMAPE, and TIC are raised by 33.32%, 19.14%, 28.69%, 39.59%, and 48.06%, respectively. In order to show the forecast results more intuitively, Figure 11 displays the evaluation values of MAE, RMSE, MAPE, SMAPE, and TIC for different models, respectively. Due to the different data structures and character of these four indices, the left  $y$ -axis of Figure 11 in the case of WTI and Brent stands for the value of MAE, RMSE, MAPE, and

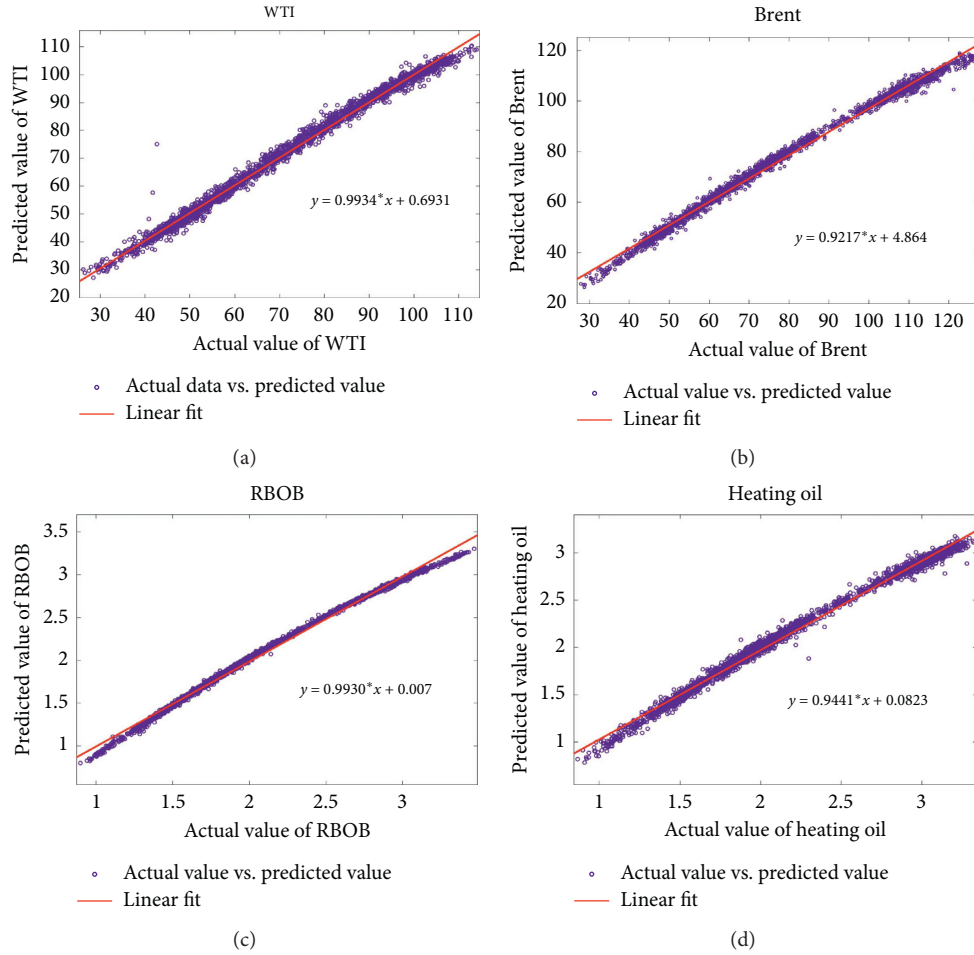


FIGURE 9: Comparisons of the predicted data and the actual data for the forecasting models. (a) WTI. (b) Brent. (c) RBOB. (d) Heating oil.

TABLE 2: Linear regression parameters from WPD-SW-LSTM model.

Parameter	WTI	Brent	RBOB	Heating oil
$a$	0.9934	0.9217	0.9930	0.9441
$b$	0.6931	4.864	0.0007	0.0823
$R$	0.9901	0.9845	0.9856	0.9822

SMAPE, and the right  $y$ -axis is the TIC value. But for the case of RBOB and heating oil, the left  $y$ -axis represents the value of MAE, RMSE, and TIC, and the right  $y$ -axis is the value of MAPE and SMAPE. From Figure 11, the MAPE and SMAPE have similar numerical results for all the case study. The MAPE, SMAPE, and TIC values of RBOB and heating oil indicate that there is no obvious difference between WPD-LSTM model and the WPD-BPNN model, but in accordance with the results of MAE and RMSE, the former is slightly better than the latter model.

In order to verify whether the proposed model is significantly different from other forecasting models (WPD-LSTM, WPD-BPNN, LSTM, BPNN, and SVM), the non-parametric Wilcoxon signed rank test is applied on two absolute errors by two compared models [57–59]. The corresponding statistical test results of the four indexes are

presented in Table 7. The results illustrate that the proposed model has statistical significance among the other models. Besides, in Tables 3–6, the error evaluations of MAE, RMSE, MAPE, SMAPE, and TIC by WPD-SW-LSTM are all smaller than those by other five models for indexes WTI, Brent, RBOB, and heating oil. It can be inferred that the WPD-SW-LSTM model is significant superior to other models for the four indexes.

*5.2. Evaluation of Multiorder Multiscale CID Analysis (MMCID).* In this section, novel error evaluation methods are proposed to detect the predicted performance. The new analysis method is based on complexity-invariant distance (CID) which generally brings about major improvements in time series classification and clustering accuracy [35]. Complexity invariance makes use of knowledge about complexity discrepancy between two different datasets as a modification factor for the existing distance measurement methods [35, 60]. By improving the CID method, multiorder multiscale complexity invariant distance (MMCID) is derived to evaluate the predictions of the energy futures prices with different forecasting models. In practical application, the complexity is not limited to a single scale. The MMCID measurement considers multiple time scales when validating

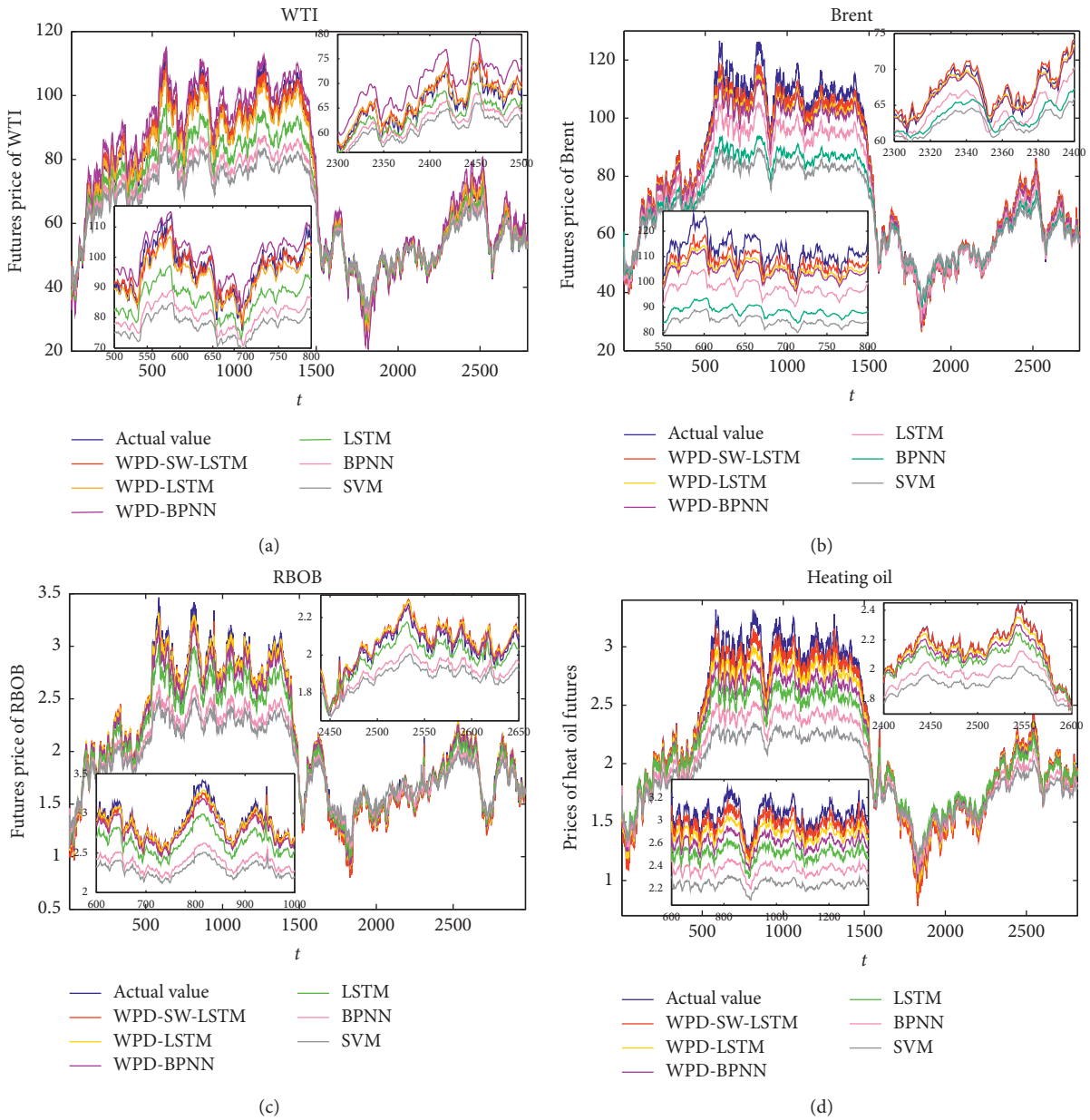


FIGURE 10: Forecasting comparison of different models for WTI, Brent, heating oil, and RBOB. (a) WTI. (b) Brent. (c) RBOB. (d) Heating oil.

TABLE 3: Prediction performance evaluation of distinct prediction models for WTI.

Model	MAE	RMSE	MAPE	SMAPE	TIC
WPD-SW-LSTM	0.8283	1.8493	1.4329	1.3143	0.0130
WPD-LSTM	1.2422	2.2842	2.0092	2.1755	0.0195
WPD-BPNN	2.1742	2.8328	2.7653	2.9991	0.0239
LSTM	3.0488	3.8097	4.6351	4.8980	0.0310
BPNN	3.5219	4.3772	5.4562	5.2742	0.0395
SVM	3.8286	4.7274	5.6108	5.9419	0.0417

TABLE 4: Prediction performance evaluation of distinct prediction models for Brent.

Model	MAE	RMSE	MAPE	SMAPE	TIC
WPD-SW-LSTM	0.6756	1.8180	0.9579	1.2339	0.0191
WPD-LSTM	1.5148	2.4533	1.3950	1.6971	0.0235
WPD-BPNN	2.0444	3.5218	3.0684	3.6750	0.0261
LSTM	3.6183	4.5880	5.2474	5.5041	0.0327
BPNN	3.9204	4.9236	5.4507	5.8135	0.0355
SVM	4.1716	5.3706	6.0980	6.6347	0.0412

TABLE 5: Prediction performance evaluation of distinct prediction models for RBOB.

Model	MAE	RMSE	MAPE	SMAPE	TIC
WPD-SW-LSTM	0.0122	0.0302	1.0350	1.0012	0.0085
WPD-LSTM	0.0351	0.0498	1.8576	1.7286	0.0166
WPD-BPNN	0.0464	0.0570	1.9764	1.8878	0.0164
LSTM	0.0514	0.0692	2.2633	2.3379	0.0189
BPNN	0.0671	0.0889	3.6352	4.3773	0.0246
SVM	0.0817	0.1070	4.4089	5.4389	0.0297

TABLE 6: Prediction performance evaluation of distinct prediction models for heating oil.

Model	MAE	RMSE	MAPE	SMAPE	TIC
WPD-SW-LSTM	0.0212	0.0642	0.4775	0.6298	0.0143
WPD-LSTM	0.0463	0.0945	1.8461	1.9538	0.0241
WPD-BPNN	0.0505	0.1271	2.0593	2.1686	0.0252
LSTM	0.0714	0.1529	3.1326	3.2484	0.0284
BPNN	0.0844	0.1980	5.2090	5.9175	0.0340
SVM	0.1056	0.2156	7.2594	8.4040	0.0465

and quantifying the connection between different futures series. The MMCID measurement can consist of the following two procedures: (i) considering one-dimensional discrete time series:  $x_1, x_2, \dots, x_j, \dots, x_N$ , consecutive coarse-grained vector  $y^{(\tau)}$  is calculated with the scale parameter  $\tau$ . The specific mathematical expressions are as follows, which refers to [61]

$$y_j^{(\tau)} = \frac{1}{\tau} \sum_{i=(j-1)\tau+1}^{j\tau} x_i, \quad 1 \leq j \leq \frac{N}{\tau}. \quad (20)$$

Particularly, when  $\tau = 1$ , the coarse-grained time series is  $y^{(1)}$ , which is merely the primitive sequence. The length of

each coarse-grained time series is equal to the length of primitive series divided by the scale parameter  $\tau$ . (ii) According to the principle of CID, we compute the multiorder value of CID for each coarse-grained time series and then acquire the MMCID method as a function with scale parameter  $\tau$ . Assuming that there are two time series,  $R$  and  $S$ , with length  $n$ ,

$$\begin{aligned} R &= r_1, r_2, \dots, r_j, \dots, r_n, \\ S &= s_1, s_2, \dots, s_j, \dots, s_n. \end{aligned} \quad (21)$$

The multiorder distance expression is given as

$$\begin{aligned} ED^q(T) &= \left( \sum_{i=1}^n (r_i - s_i)^q \right)^{(1/q)}, \\ CF^q(R, S) &= \frac{\max\{CE^q(R), CE^q(S)\}}{\min\{CE^q(R), CE^q(S)\}}, \\ CE^q(T) &= \left( \sum_{i=1}^{n-1} (t_{i+1} - t_i)^q \right)^{(1/q)}, \end{aligned} \quad (22)$$

$$M - CID(R, S) = ED^q(R, S) \times CF^q(R, S),$$

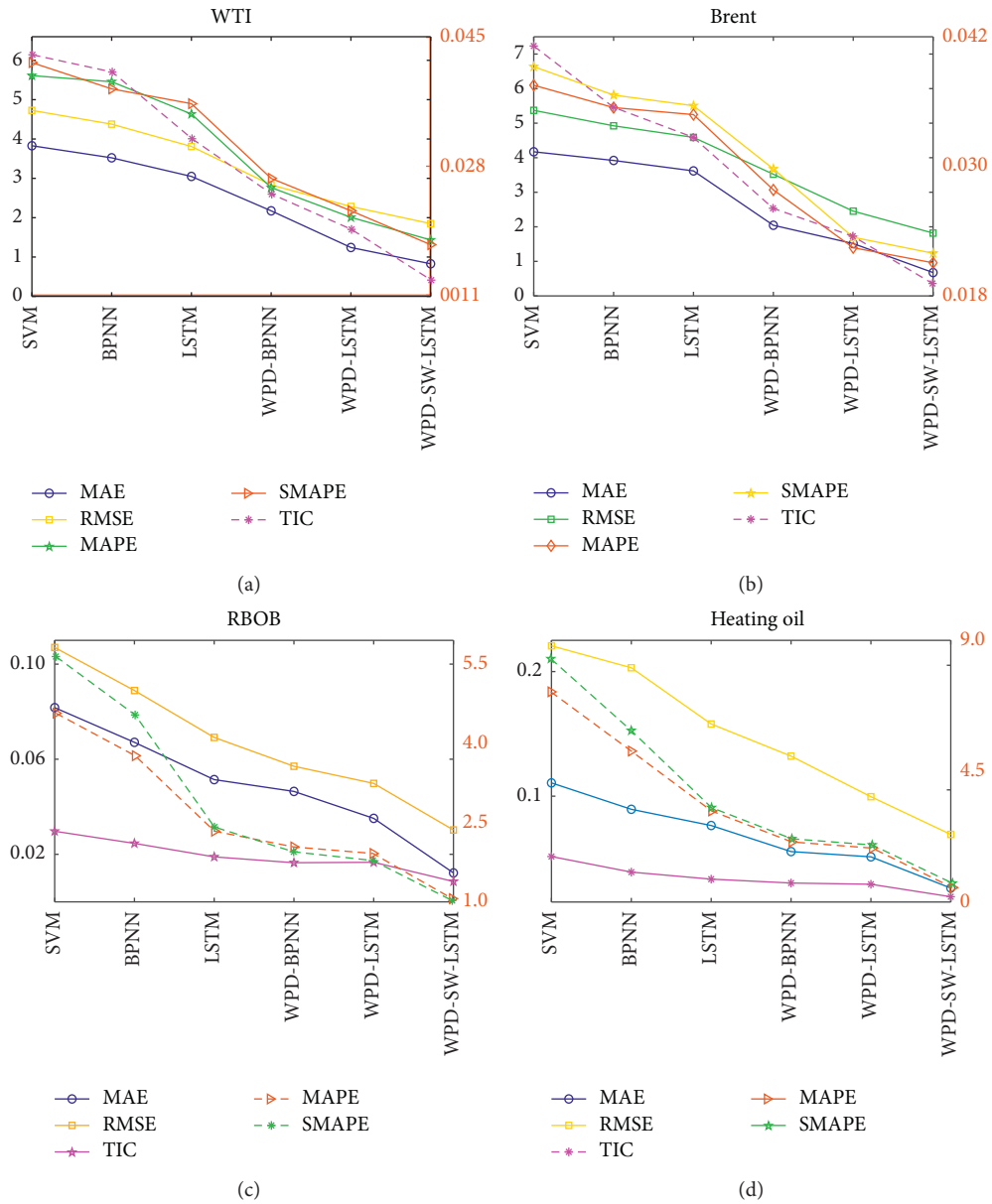


FIGURE 11: Forecasting comparison of the evaluation errors from the six involved models. (a) WTI. (b) Brent. (c) RBOB. (d) Heating oil.

TABLE 7: Wilcoxon signed rank test for proposed model with different prediction models.

		WTI	Brent	RBOB	Heating oil
WPD-LSTM	$H$	1	1	1	1
	$z$ value	-39.1953	-42.0197	-7.7244	-20.9041
	Prob. $p$	$3.5050e^{-10}$	$3.1244e^{-12}$	$1.1234e^{-14}$	$4.9117e^{-6}$
WPD-BPNN	$H$	1	1	1	1
	$z$ value	-22.2057	-21.8502	-30.8334	-20.6775
	Prob. $p$	$3.0267e^{-6}$	$4.3975e^{-9}$	$9.3655e^{-29}$	$5.5209e^{-9}$
LSTM	$H$	1	1	1	1
	$z$ value	-45.7889	-36.8169	-26.5620	-18.0050
	Prob. $p$	$6.0053e^{-19}$	$9.8998e^{-27}$	$1.8654e^{-15}$	$1.7815e^{-7}$
BPNN	$H$	1	1	1	1
	$z$ value	-23.8007	-30.5701	-31.1161	-21.1325
	Prob. $p$	$3.7303e^{-11}$	$8.5523e^{-29}$	$1.4575e^{-22}$	$3.9941e^{-6}$
SVM	$H$	1	1	1	1
	$z$ value	-8.9625	-37.2304	-33.3299	-21.0766
	Prob. $p$	$1.6290e^{-27}$	$2.1975e^{-33}$	$1.4235e^{-23}$	$1.3043e^{-8}$

TABLE 8: MMCID value between the actual data and the corresponding predictions.

Index	WTI	Brent	RBOB	Heating oil
WPD-SW-LSTM	120.1289	237.8508	4.5909	8.9039
WPD-LSTM	239.3461	384.7985	6.8226	11.9508
WPD-BPNN	305.2824	416.4688	11.4907	13.7667
LSTM	577.9065	516.0784	18.0672	18.0709
BPNN	730.7260	595.6528	26.0103	24.4857
SVM	929.2038	779.0730	41.2264	31.5295

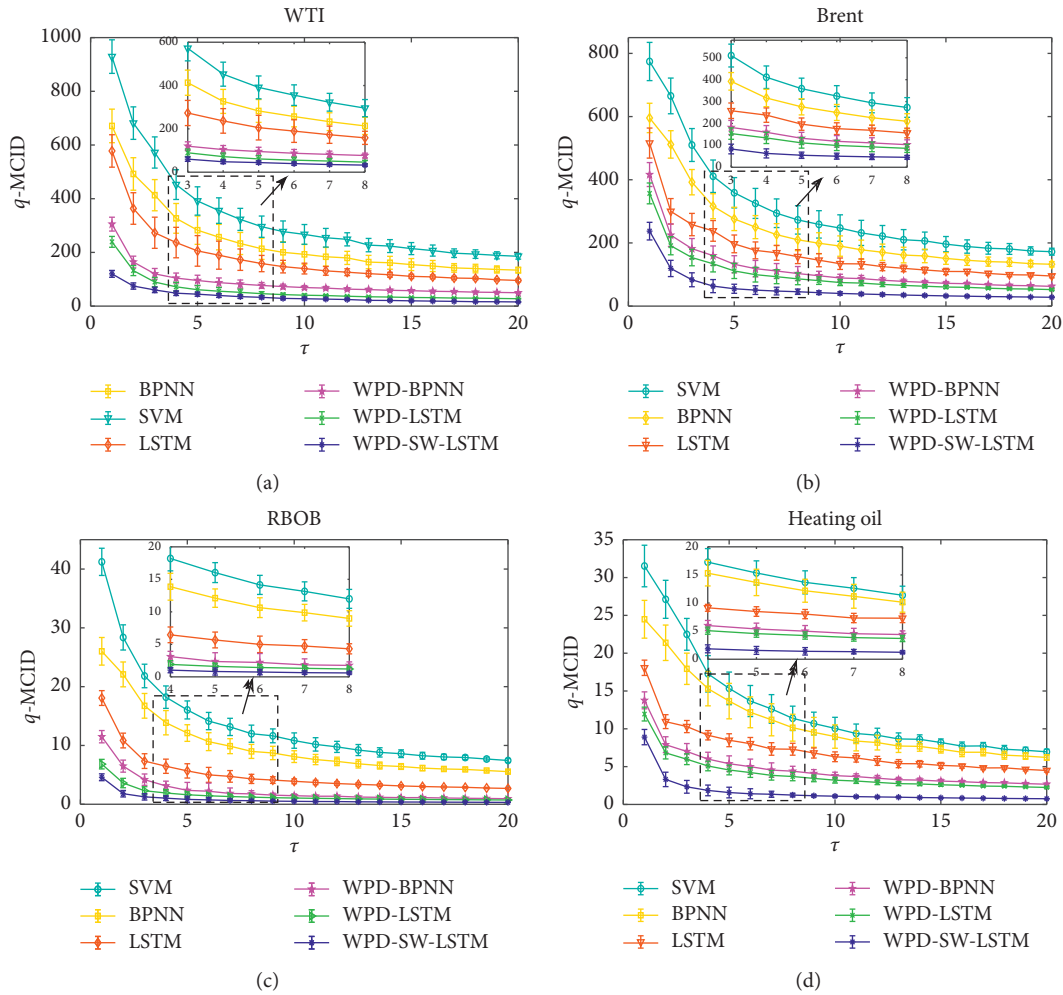


FIGURE 12: The MMCID curves of the actual futures data points and the forecasting results from different forecasting models. (a) WTI. (b) Brent. (c) RBOB. (d) Heating oil.

where  $ED^q(R, S)$  between two time series  $R$  and  $S$  indicates complexity invariant by introducing a correction index.  $CF^q$  is a complexity correction index, and  $CE^q(T)$  is a complexity evaluation of time series  $T$ . Moreover,  $CF^q$  gives reasons for complexity differences of different datasets into comparison. It separates time series with distinctly different complexities to be further apart. And multiorder parameter  $q$  is applied to enlarge the performance of great changes in the process of error evaluation.

When evaluating with the MMCID method, the actual value can be regarded as series  $R$  and the predicted results as

the series  $S$ . According to the theory of the MMCID, the predicted effectiveness is better when the MMCID value is smaller. It also indicates that the fluctuation trends of the prediction are almost consistent with the actual data. In this study, the parameter  $q$  is set to 2 and  $\tau$  is from 1 to 20. Table 8 shows the specific MMCID values between the forecasting results and the actual values from the six mentioned models when the scale parameter  $\tau = 1$ . The empirical results from the four different types of experiment data demonstrate that the proposed hybrid model performs much better than the other five forecasting models. Figure 12 shows MMCID results

TABLE 9: Prediction performance evaluation of hybrid forecasting models.

Errors	MAE	RMSE	MAPE	SMAPE	TIC
Index			WTI		
CEEMDAN-LSTM	1.2017	2.1849	1.7099	1.8371	0.0167
VMD-LSTM	1.3158	2.5268	2.2632	2.4196	0.0192
ST-GRU	1.1701	2.2165	1.6895	1.9726	0.0151
WPD-SW-LSTM	0.8283	1.8493	1.4329	1.3143	0.0130
Index			Brent		
CEEMDAN-LSTM	0.9782	2.3274	1.3412	1.5177	0.0228
VMD-LSTM	1.2814	2.5638	1.4101	1.7065	0.0239
ST-GRU	1.1267	2.2680	1.2826	1.4371	0.0215
WPD-SW-LSTM	0.6756	1.8180	0.9579	1.2339	0.0191
Index			RBOB		
CEEMDAN-LSTM	0.0269	0.0381	1.6774	1.5205	0.0168
VMD-LSTM	0.0368	0.0558	1.9783	1.8152	0.0175
ST-GRU	0.0326	0.0328	1.6186	1.4322	0.0116
WPD-SW-LSTM	0.0122	0.0302	1.0350	1.0012	0.0085
Index			Heating oil		
CEEMDAN-LSTM	0.0376	0.0805	1.2376	1.5278	0.0183
VMD-LSTM	0.0497	0.1014	1.8509	2.0125	0.0252
ST-GRU	0.0408	0.0732	1.0338	1.2957	0.0171
WPD-SW-LSTM	0.0212	0.0642	0.4775	0.6298	0.0143

TABLE 10: Wilcoxon signed rank test for proposed model with different hybrid models.

		WTI	Brent	RBOB	Heating oil
	$H$	1	1	1	1
CEEMDAN-LSTM	$z$ value	-38.4806	3.4128	23.3774	3.0441
	Prob. $p$	$3.1639e^{-10}$	$6.4302e^{-4}$	$7.2652e^{-12}$	0.0023
	$H$	1	1	1	1
VMD-LSTM	$z$ value	-20.6386	7.2933	21.2588	-2.7126
	Prob. $p$	$1.2432e^{-9}$	$3.0256e^{-13}$	$2.7349e^{-10}$	0.0067
	$H$	1	1	1	1
ST-GRU	$z$ value	-23.4035	-15.9011	-24.8309	-5.3581
	Prob. $p$	$3.9388e^{-12}$	$6.2225e^{-15}$	$4.1571e^{-13}$	$8.4113e^{-8}$

between the actual futures prices series and the corresponding prediction of them from each predictive model. It is distinctly noticed that the MMCID value between actual data and the prediction ones by the WPD-SW-LSTM model is the smallest one of all, and the results from hybrid models are much better than those from single models for all the four contemplated futures indices. With the novel estimation method, the forecasting merits of the proposed WPD-SW-LSTM model are further manifested, and the productiveness of the SW method added to WPD-LSTM model is also revealed distinctively. In view of the above empirical analysis, the established new hybrid forecasting approach is effective for improving the accuracy of energy futures prices.

### 5.3. Comparative Analysis with Existing Hybrid Models.

In this section, the latest hybrid models are considered as the benchmark models to make predictions on the selected four energy futures indexes. Recently, many researchers have combined decomposition methods with machine learning algorithm to establish hybrid forecasting models. Lin et al. [34] proposed the CEEMDAN-LSTM model to the forecast

of exchange rate. Niu et al. [32] and He et al. [45] applied the VMD-LSTM model to the forecasting fields of stock prices and exchange rate movements. Li and Wang [62] developed a novel model ST-GRU by embedding stochastic time intensity function into gated recurrent unit model (GRU). Therefore, this section makes comparative analysis between the WPD-SW-LSTM model with the CEEMDAN-LSTM, VMD-LSTM, and ST-GRU models, respectively. Table 9 has listed the error evaluation results of the four hybrid forecasting models. Table 10 is the hypothesis test results of Wilcoxon signed rank test for different paired models. The  $p$  values are all close to 0 and the  $H$  values are 1 through calculation by hypothesis test, indicating that test rejects null hypothesis. Hence, the prediction error of the WPD-SW-LSTM model is significantly different (under the significance level of 0.05) from the error of the other three hybrid models. Furthermore, compared with the results of other models, all the error evaluations of the forecasting performances in Table 9 are very close, but those of the proposed model are smaller than the errors of the other models. Combined with the results of the statistical test in Table 10, it can be deduced that the prediction efficiency of the proposed model is more



superior to the latest three hybrid models for energy futures prices forecasting.

## 6. Conclusion

In this research, a new hybrid forecasting model, WPD-SW-LSTM, has been set up by integrating the wavelet packet decomposition based on LSTM with stochastic time strength weight function method. After decomposing the primitive futures series into several subseries, each forecasting model for the different subseries  $SS_i$  has been established according to its own frequency band properties. The correlation coefficient values ( $R$ ) from four energy futures series are all above 0.98 and extremely near 1, which implies that the proposed model performs great prediction effect. Furthermore, compared with the empirical results of SVM, BPNN, LSTM, WPD-BPNN, and WPD-LSTM forecasting models, the predicted values and different error evaluation reveal that the proposed WPD-SW-LSTM forecasting model has strong points in upgrading the accuracy of energy futures prices. In addition, according to the evaluation errors of MAE, RMSE, MAPE, SMAPE, and TIC, the hybrid models WPD-SW-LSTM, WPD-LSTM, and WPD-BPNN have better prediction performance than the individual models, LSTM, BPNN, and SVM. The effectiveness of stochastic time strength weight function is the key that the accuracy of the WPD-SW-LSTM model is far more than the other five models. By introducing the novel evaluation error, MMCID method and the forecasting effectiveness of the proposed model are further confirmed. At the last section, compared with the recent hybrid CEEMDAN-LSTM, VMD-LSTM, and ST-GRU models, by Wilcoxon test, the proposed model is significantly different from the forecasting errors of the other three models. Combined with the error evaluation results, it can be referred that the forecasting accuracy of the proposed model is the highest among the other benchmark models for energy futures prices forecasting.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank the financial supports from the funds of North China University of Technology, China (no. 110051360002).

## References

- [1] K. Lang and B. R. Auer, "The economic and financial properties of crude oil: a review," *The North American Journal of Economics and Finance*, vol. 52, Article ID 100914, 2020.
- [2] D. Faruk, "A hybrid neural network and ARIMA model for water quality time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 86–94, 2010.
- [3] H. Liu, H.-Q. Tian, and Y.-F. Li, "Comparison of two new ARIMA-ANN and ARIMA-kalman hybrid methods for wind speed prediction," *Applied Energy*, vol. 98, pp. 415–424, 2012.
- [4] H. Abdollahi and S. B. Ebrahimi, "A new hybrid model for forecasting Brent crude oil price," *Energy*, vol. 200, Article ID 117520, 2020.
- [5] J. Li, S. Zhu, and Q. Wu, "Monthly crude oil spot price forecasting using variational mode decomposition," *Energy Economics*, vol. 83, pp. 240–253, 2019.
- [6] Z. Liao and J. Wang, "Forecasting model of global stock index by stochastic time effective neural network," *Expert Systems with Applications*, vol. 37, no. 1, pp. 834–841, 2010.
- [7] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.
- [8] J. Wang and J. Wang, "Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks," *Neurocomputing*, vol. 156, pp. 68–78, 2015.
- [9] J. Wang and J. Wang, "Forecasting energy market indices with recurrent neural networks: case study of crude oil price fluctuations," *Energy*, vol. 102, pp. 365–374, 2016.
- [10] B. Wang and J. Wang, "Deep multi-hybrid forecasting system with random EWT extraction and variational learning rate algorithm for crude oil futures," *Expert Systems with Applications*, vol. 2020, Article ID 113686, 2020.
- [11] P. Du, J. Wang, W. Yang, and T. Niu, "A novel hybrid model for short-term wind power forecasting," *Applied Soft Computing*, vol. 80, pp. 93–106, 2019.
- [12] H. Su, E. Zio, J. Zhang, M. Xu, X. Li, and Z. Zhang, "A hybrid hourly natural gas demand forecasting method based on the integration of wavelet transform and enhanced Deep-RNN model," *Energy*, vol. 178, pp. 585–597, 2019.
- [13] L. Q. Han, *Theory, Design and Application of Artificial Neural Network*, Chemical Industry Press, Beijing, China, 2002.
- [14] Z. Yang and J. Wang, "A hybrid forecasting approach applied in wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm," *Energy*, vol. 160, pp. 87–100, 2018.
- [15] M. Ghiassi, H. Saidane, and D. K. Zimbra, "A dynamic artificial neural network model for forecasting time series events," *International Journal of Forecasting*, vol. 21, no. 2, pp. 341–362, 2005.
- [16] H. H. H. Aly, "A proposed intelligent short-term load forecasting hybrid models of ANN, WNN and KF based on clustering techniques for smart grid," *Electric Power Systems Research*, vol. 182, Article ID 106191, 2020.
- [17] Z. Berradi and M. Lazaar, "Integration of principal component analysis and recurrent neural network to forecast the stock price of Casablanca stock exchange," *Procedia Computer Science*, vol. 148, pp. 55–61, 2019.
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [19] E. Azari and S. Vrudhula, "An energy-efficient reconfigurable LSTM accelerator for natural language processing," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 4450–4459, Los Angeles, CA, USA, December 2019.
- [20] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

- [21] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN-LSTM model for gold price time-series forecasting," *Neural Computing and Applications*, vol. 32, no. 5, pp. 1–10, 2020.
- [22] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neuro-computing*, vol. 323, pp. 203–213, 2019.
- [23] T. Hussain, K. Muhammad, A. Ullah et al., "Cloud-assisted multiview video summarization using CNN and bidirectional LSTM," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 77–86, 2019.
- [24] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.
- [25] L. Yu, S. Wang, and K. K. Lai, "Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm," *Energy Economics*, vol. 30, no. 5, pp. 2623–2635, 2008.
- [26] A. Safari and M. Davallou, "Oil price forecasting using a hybrid model," *Energy*, vol. 148, pp. 49–58, 2018.
- [27] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, Article ID 117794, 2020.
- [28] J. Wang and J. Wang, "Forecasting stochastic neural network based on financial empirical mode decomposition," *Neural Networks*, vol. 90, pp. 8–20, 2017.
- [29] D. Wang, H. Luo, O. Grunder, Y. Lin, and H. Guo, "Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm," *Applied Energy*, vol. 190, pp. 390–407, 2017.
- [30] A. A. Abdoos, "A new intelligent method based on combination of VMD and ELM for short term wind power forecasting," *Neurocomputing*, vol. 203, pp. 111–120, 2016.
- [31] H. Liu and C. Chen, "Data processing strategies in wind energy forecasting models and applications: a comprehensive review," *Applied Energy*, vol. 249, pp. 392–408, 2019.
- [32] H. Niu, K. Xu, and W. Wang, "A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network," *Applied Intelligence*, vol. 50, no. 12, pp. 1–14, 2020.
- [33] M. A. Jallal, A. Gonzalez-Vidal, A. F. Skarmeta et al., "A hybrid neuro-fuzzy inference system-based algorithm for time series forecasting applied to energy consumption prediction," *Applied Energy*, vol. 268, Article ID 114977, 2020.
- [34] H. Lin, Q. Sun, and S.-Q. Chen, "Reducing exchange rate risks in international trade: a hybrid forecasting approach of CEEMDAN and multilayer LSTM," *Sustainability*, vol. 12, no. 6, Article ID 2451, 2020.
- [35] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. De Souza, "CID: an efficient complexity-invariant distance for time series," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 634–669, 2014.
- [36] R. R. Coifman, Y. Meyer, and V. Wickerhauser, "Wavelet analysis and signal processing," in *Wavelets and their Applications* Jones and Barlett, Sudbury, MA, USA, 1992.
- [37] J. D. Wu and C. H. Liu, "An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network," *Expert Systems with Applications*, vol. 36, Article ID 42788C4286, 2009.
- [38] J. Zarei and J. Poshtan, "Bearing fault detection using wavelet packet transform of induction motor stator current," *Tribology International*, vol. 40, Article ID 7638C769, 2007.
- [39] L. Y. Zhao, L. Wang, and R. Q. Yan, "Rolling bearing fault diagnosis based on wavelet packet decomposition and multi-scale permutation entropy," *Entropy*, vol. 17, no. 9, pp. 6447–6461, 2015.
- [40] N. Amjady and F. Keynia, "Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm," *Energy*, vol. 34, no. 1, pp. 46–57, 2009.
- [41] H. Liu, X. Mi, and Y. Li, "Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network," *Energy Conversion and Management*, vol. 166, pp. 120–131, 2018.
- [42] J. Schmidhuber, S. Hochreiter, and Y. Bengio, "Evaluating benchmark problems by random guessing," in *A Field Guide to Dynamical Recurrent Networks*, J. Kolen and S. Cremer, Eds., Wiley-IEEE Press, Hoboken, NJ, USA, 2001.
- [43] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
- [44] T. N. Sainath, O. Vinyals, A. Senior et al., "Convolutional, long short-term memory, fully connected deep neural networks," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, IEEE, South Brisbane, Australia, April 2015.
- [45] F. He, J. Zhou, Z.-k. Feng, G. Liu, and Y. Yang, "A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm," *Applied Energy*, vol. 237, pp. 103–116, 2019.
- [46] D. Dufresne, "The integral of geometric brownian motion," *Advances in Applied Probability*, vol. 33, no. 1, pp. 223–241, 2001.
- [47] J. Gatheral and A. Schied, "Optimal trade execution under geometric Brownian motion in the Almgren and Chriss framework," *International Journal of Theoretical and Applied Finance*, vol. 14, no. 03, pp. 353–368, 2011.
- [48] G. Dudek, "Generating random weights and biases in feed-forward neural networks with random hidden nodes," *Information Sciences*, vol. 481, pp. 33–56, 2019.
- [49] M. Abdechiri, M. R. Meybodi, and H. Bahrami, "Gases brownian motion optimization: an algorithm for optimization (GBMO)," *Applied Soft Computing*, vol. 13, no. 5, pp. 2932–2946, 2013.
- [50] Y. Yang, J. Wang, and B. Wang, "Prediction model of energy market by long short term memory with random system and complexity evaluation," *Applied Soft Computing*, vol. 95, Article ID 106579, 2020.
- [51] R. H. Abiyev, "Fuzzy wavelet neural network based on fuzzy clustering and gradient techniques for time series prediction," *Neural Computing and Applications*, vol. 20, no. 2, pp. 249–259, 2011.
- [52] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527–529, 1993.
- [53] X. T. Liu, "Study on data normalization in bp neural network," *Mechanical Engineering & Automation*, vol. 3, pp. 122–123, 2010.
- [54] S. Lahmiri, "A comparison of PNN and SVM for stock market trend prediction using economic and technical information," *International Journal of Computer Applications*, vol. 29, pp. 24–30, 2011.

- [55] R. Rosillo, J. Giner, and D. De la Fuente, "Stock market simulation using support vector machines," *Journal of Forecasting*, vol. 33, no. 6, pp. 488–500, 2014.
- [56] T. Papadimitriou, P. Gogas, and E. Stathakis, "Forecasting energy markets using support vector machines," *Energy Economics*, vol. 44, pp. 135–142, 2014.
- [57] G.-F. Fan, S. Qing, H. Wang, W.-C. Hong, and H.-J. Li, "Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting," *Energies*, vol. 6, no. 4, pp. 1887–1901, 2013.
- [58] Y. Chen, W.-C. Hong, W. Shen, and N. Huang, "Electric load forecasting based on a least squares support vector machine with fuzzy time series and global harmony search algorithm," *Energies*, vol. 9, no. 2, p. 70, 2016.
- [59] M.-W. Li, Y.-T. Wang, J. Geng, and W.-C. Hong, "Chaos cloud quantum bat hybrid optimization algorithm," *Non-linear Dynamics*, vol. 103, no. 1, pp. 1167–1193, 2021.
- [60] J. A. Rodger, "A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1813–1829, 2014.
- [61] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of biological signals," *Physical Review E*, vol. 71, Article ID 021906, 2005.
- [62] J. Li and J. Wang, "Forecasting of energy futures market and synchronization based on stochastic gated recurrent unit model," *Energy*, vol. 213, Article ID 118787, 2020.

## Research Article

# A Robust Context-Based Deep Learning Approach for Highly Imbalanced Hyperspectral Classification

Juan F. Ramirez Rochac <sup>1</sup>, Nian Zhang <sup>2</sup>, Lara A. Thompson <sup>3</sup> and Tolessa Deksissa <sup>4</sup>

<sup>1</sup>Department of Computer Science & Information Technology, University of the District of Columbia, Washington, DC 20008, USA

<sup>2</sup>Department of Electrical & Computer Engineering, University of the District of Columbia, Washington, DC 20008, USA

<sup>3</sup>Biomedical Engineering Program, Department of Mechanical Engineering, University of the District of Columbia, Washington, DC 20008, USA

<sup>4</sup>Water Resources Research Institute, University of the District of Columbia, Washington, DC 20008, USA

Correspondence should be addressed to Juan F. Ramirez Rochac; [jrochac@udc.edu](mailto:jrochac@udc.edu)

Received 18 March 2021; Revised 13 April 2021; Accepted 25 June 2021; Published 7 July 2021

Academic Editor: Anastasios D. Doulamis

Copyright © 2021 Juan F. Ramirez Rochac et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hyperspectral imaging is an area of active research with many applications in remote sensing, mineral exploration, and environmental monitoring. Deep learning and, in particular, convolution-based approaches are the current state-of-the-art classification models. However, in the presence of noisy hyperspectral datasets, these deep convolutional neural networks underperform. In this paper, we proposed a feature augmentation approach to increase noise resistance in imbalanced hyperspectral classification. Our method calculates context-based features, and it uses a deep convolutional neuronet (DCN). We tested our proposed approach on the Pavia datasets and compared three models, DCN, PCA + DCN, and our context-based DCN, using the original datasets and the datasets plus noise. Our experimental results show that DCN and PCA + DCN perform well on the original datasets but not on the noisy datasets. Our robust context-based DCN was able to outperform others in the presence of noise and was able to maintain a comparable classification accuracy on clean hyperspectral images.

## 1. Introduction

Advances in data collection and data warehousing technologies have led to a wealth of massive repositories of data. Together with active research in artificial intelligence, big data science promises mountain ranges of unexplored datasets and the smart tools to extract relevant information. An important goal in computer-based hyperspectral imaging is to be able to accurately perform this information mining without human work. Government, industry, and academia sectors seek to automate this process. They find it valuable for their future to be able to reduce the human requirement in core processing tasks, such as segmentation, classification, and its applications.

Ever since Vapnik's [1, 2] work transformed the statistical learning theory community, research has indicated the considerable potential of SVM in supervised

classification. However, in many real-world classification problems such as remote sensing, medical diagnosis, object recognition, and business decision-making, the costs of selecting a poor kernel for high dimensional data is too high in terms of computational performance and a handicap to robust, real-time hyperspectral classification and segmentation.

More recently, deep networks have dominated classification problems, such as image segmentation. Convolutional-based neural networks or CNNs are driving advances in recognition. CNNs are not only improving for all domains of image classification [3–7] but also making progress on object detection [8–10], key-point-based prediction [11, 12], and local correspondence [13]. The natural next step in the progression from coarse to fine inference is to make a prediction at every pixel. Prior approaches have used Deep CNNs for image segmentation

[14–20], in which each pixel is labeled, but with shortcomings that this work addresses.

Typically, DCN-based algorithms use the output of the last layer of the network to assign category labels. Imposing a softmax layer on top of a fully-connected dense layer, DCN focuses on semantic information. However, when the task we are interested in is more granular, such as one of classifying mixed pixels or dealing with imbalanced multiclass classification of hyperspectral images, these last layers are not optimal.

Image segmentation faces yet another challenging gap: global information answers the what, while local information provides the where. It is not immediately clear that deep convolutional neural networks for image classification yield a structure sound enough for accurate, pixel-wise multiclass classification. Moreover, when working with high dimensional features, there is often no go-to algorithm that is exact and has acceptable performance. To obtain a speed improvement, many practical applications are forced to settle for approximation approaches, in which they do not return exact answers. In practice, numerical optimizations and fast approximation saturate the spectrum of algorithms and research. However, image segmentation can also be explored as the reconstruction to a low-quality image from its high quality observations. This point of view has many important applications, such as low-level image processing, remote sensing, medical imaging, and surveillance.

There are also paramount applications that would benefit from advances in unsupervised image segmentation, such as medical applications and homeland security. Early detection of tumors, kidney disease, heart disease, microbleeds, and microdamages is critical to worldwide public health. There is significant research and new investments for advancing magnetic resonance imaging technology that can accurately aid in early diagnosis. The authors in [21] reviewed the principles and applications of a gradient echo MRI, the so called  $T2^*$  weighted. During COVID, the pharmaceutical industry joins forces with academia to develop algorithms for automated assessment of large-scale datasets [22]. Detection of illicit drugs, warfare agents, and dangerous substances is critical to security. The authors in [23] introduced a new technology that can rapidly detect explosives using a thermal imager. This thermal spectroscopy pushes the boundaries of traditional image and signal processing techniques.

The problem is that the state-of-the-art in machine learning and data science demands for abundance of labeled samples, which require domain expert input. This is not feasible to spend time and effort labeling training samples. It is more efficient to develop a new method that scales and requires small number of labeled training samples.

Moreover, noise is a challenging variable, specially within imbalanced data. Hyperspectral imaging is such a data containing highly-imbalanced classes. Multiclass classification using DCN suffers from the presence of noise. Therefore, this study proposes a method that can address these challenges using a deep learning-based image clustering model that combines both an adaptive dimensionality reduction approach and a robust feature augmentation

approach which can cluster different types of imaging datasets with high positive predictive value.

The main contribution of this paper is a new pre-processing approach to deal with noisy, highly-imbalanced hyperspectral classification. In Section 2, we present a literature review. In Section 3, we explain our approach. In Section 4, we explain our experiments, while in Section 5, we compare our results. And in Section 6, we present our conclusions and future lines of research.

## 2. Related Works

This section presents previous works and relevant literature in the areas of dimensionality reduction, feature augmentation, noise reduction, and hyperspectral image classification.

*2.1. Dimensionality Reduction.* As big data, cloud computing becomes the standard for data storage, and high dimensional datasets are more and more commonplace. To process such large oceans of data, dimensionality reduction offers two options: feature projection and feature selection. Feature projection techniques transform data from a highly dimensional space to a new space with a lower dimensionality. Principal Component Analysis is one of the most popular linear transformations. In [24] the authors effectively conducted a dimension reduction by applying the principal component analysis to highly overlapped photo-thermal infrared imaging dataset. Feature selection techniques are an alternative that aims to choose the most information-rich features and discard irrelevant features and noise. The authors in [25, 26] present different feature selection techniques to integrate spectral band selection and hyperspectral image classification in an adaptive fashion, with the ultimate goal of improving the analysis and interpretation of hyperspectral imaging.

Recent literature [27] proposes a Kronecker-decomposable component analysis model that combines dictionary learning and component analysis with great results on low rank modeling. The Kronecker product is compatible with the most common matrix decomposition. Therefore, it can be used to learn low-ranking dictionaries in tensor factorization. It also can effectively remove noise.

Principal Component Analysis [28] or PCA is a classical dimensionality reduction with multiple implementations. One intuitive implementation consists of six steps: standardization, covariance, eigenvalues, eigenvectors, reduction, and projection. This formulation is based on maximizing variance within a low-dimensional projection. There are other formulations that scale better to high dimensionality. One of such solver implementations consists of breaking down PCA into two easy-to-calculate sub-problems: alternating least square linear regressions [29] using an iterative algorithm based on the idea that the product of principal orthogonal components can be an approximation to the original data.

Despite the fact that PCA is among the most established techniques for dimensionality reduction, the story does not

end here. There are many other techniques that show great empirical applications and theoretical guarantees. The authors in [30] introduced a Forward Selection Component Analysis and obtained comparable results to PCA and Sparse PCA. And in [31, 32], anomaly and change detection was carried out with great success in hyperspectral imaging. Yet, [33] suggests PCA as yet a powerful preprocessing step to denoise data. Similarly to numerous other noise reduction methods including patents [34], PCA works under the assumption that the signal needs to be cleaned from the same global noise.

*2.2. Image Classification.* Deep learning and big data science are the state-of-the-art in image classification. From support vector machines to convolutional neural networks to spectral clustering, both academia and industry keep pushing for more innovative research. Collaborative and in particular interdisciplinary research is needed to bring these advances to other fields and transform innovations into applications. The authors in [35] and [36] bear witness to the benefits of incorporating diversity to research teams. With authors with top degrees in civil engineering, computer science, and communications and graduate and undergraduate authors, these teams show that in order to push the science forward we need the help of everyone.

There are many classic image segmentation algorithms, from simple thresholding to similarity-based clustering to connectedness and discontinuity-based detection. Threshold-based image segmentation seeks to divide the scale range into background and a set of target foregrounds based on global or local information, for instance, minimizing their interclass variance, maximizing entropy, and/or fuzzy sets theory. One big advantage of using these simple methods is the low computational cost in terms of code complexity which is evident in fast speed operation. This is mainly because thresholding does not take into account spatial information. One drawback is that in the presence of noise, results are not optimal. Similarity-based segmentation uses the idea of clustering based on certain aggregation in feature space. K-means clustering is one of the most well-known unsupervised algorithms. K-means groups together pixels based on their distance; hence, it is considered a distance-based partition method. Connectedness-based image segmentation is a region growing approach that links together points with similar features creating homogeneous and smoothly-connected segments. Discontinuity-based image segmentation seeks to detect object edges or high changes in intensity. Its motivation comes from the idea that there is always a discontinuity between different regions or segments. These discontinuities can be detected using derivatives. Prewitt, Sobel, and Laplacian operators are among the most popular differential operators for spatial domain edge detection which can be applied using convolution for image segmentation.

There are also emerging machine learning and deep learning approaches. Support Vector Machines or SVM is a machine learning algorithm that models classification tasks as optimization problems subject to inequality constraints.

The original algorithm [1] was invented by Vapnik and Chervonenkis in 1963. SVM uses a dual Lagrangian, which depends only on labeled samples. The traditional SVM philosophy consists of finding the hyperplane that maximizes the margin between points of different classes. Note that the hyperplane is at the centre of the margin that separates the two classes. The kernel trick was introduced in [2] by Cortes in 1995. This hyperplane is denoted by the perpendicular vector  $w$  from the origin and it is characterized by (12). Introduce a new variable  $Y$  subscript  $i$ -th such that  $Y_i$  is positive (+1) for gray samples and it is negative (-1) for yellow samples. This optimization problem is solved using a Lagrangian multiplier (13). After applying the partial derivatives, it is evident that the solution only depends on the inner product of the supporting vectors  $x_i$ . Different kernel functions SVM may be employed to solve nonlinearly separable samples. Thus, SVM performs so well on binary classification.

Deep Convolutional Neuronets or DCN is a deep learning algorithm that models a classification task as series of convolutional layers, pooling layers, dropout, and an activation layer usually consisting of a softmax function. CNN-based learning has recently achieved expert level performance in various applications. In [37] the authors present a deep fully convolutional neural network for semantic pixel-wise segmentation. Evaluation of the decoder variants shows that accuracy increases for larger decoders for a given encoder network. Experimental results on road scenes and indoor scenes show that the proposed SegNet outperforms other segmentation benchmarks.

Some other applications of DCN-based segmentation are listed in [38, 39] and [40]. In [38], the authors extended the original DeepLab with more speed, accuracy, and simplicity by compiling a comprehensive evaluation on benchmark and challenging datasets, such as PASCAL VOC 2012, Cityscapes, among others. In [39] the authors present a new unsupervised image segmentation based on the centre of a local region. The authors validated their work on 2D and 3D medical images. MATLAB was used to implement the approach on X-rays, abdominal and cardiovascular MRI images. In [40] the authors present an image segmentation approach that recasts the problem into a binary pairwise classification of pixels.

Deep learning high speed and accuracy come with a price: subject matter expert labor to label. DCN-based approaches are supervised learning and labeled samples are needed in abundance which results in a high demand for SME input. Despite the shortcomings, multiple research initiatives are pushing the boundaries of noninvasive medicine, remote sensing, and natural language processing. Deep learning-based models stand at the core of these emerging applications.

*2.3. Applications in Medical Image Processing.* U-NET deep FCN structure is highly applicable for medical image segmentation. Multiple U-NET variants [41–43] and domain specific models [44] have been applied to process medical images. For instance, [41] presents a U-Net variant for image

segmentation on brain tumor MRI scans while [42] presents another U-Net variant based on nested and dense skip connections for medical image segmentation. Moreover, [43] introduces a robust self-adapting U-Net-based framework for medical image segmentation. And [44] adds the emerging attention mechanism to a nested U-Net architecture for image segmentation on liver CT scans. One interesting medical application of image segmentation using a deep learning model is presented in [45]. A new hybrid of the classic V-Net architecture is used to help detect kidney and renal tumors on CT imaging with successful performance of medical segmentation. This wealth of deep learning research branches out from the U-Net model and provides expert-level solutions to medical image segmentation.

Recently, one shot learning models have been proposed to detect COVID-19 using medical images. Signoroni et al. [46] introduced a learning-based solution designed to assess the severity of COVID-19 disease by means of automated X-ray image processing, a domain specific implementation of [42]. Furthermore, [47] compiles an early survey of medical imaging research toward COVID-19 detection, diagnosis, and follow-up. One of their findings is the proliferation of AI-empowered applications which use X-rays and/or CT scans to provide partial information about patients with COVID-19. This reinforces the sense that deep learning-based solutions are widely used in medical image processing.

Tensor-based learning has also been incorporated into medical image processing and hyperspectral imaging. An et al. [48] presented a tensor-based low rank decomposition model for hyperspectral images and evaluates its classification accuracy on hyperspectral cubes. Moreover, the authors in [49] proposed another tensor-based representation to better preserve the spatial and spectral information and capture the local and global structures of hyperspectral images. Yet these models do not focus on imbalanced datasets nor try to solve the denoising problem. Recently, in the field of optical coherence tomography (OCT) [50] has introduced a tensor-based learning model, which tackles the denoising problem on high resolution OCT medical images with great results. However, it is unclear how well tensor-based models would represent the structure of imbalanced datasets and will remain outside the scope of our work.

*2.4. Applications in Natural Language Processing.* Natural language processing (NLP) is a field with multiple-machine-learning- (ML-) and deep-learning- (DL-) based research initiatives. With sentiment analysis as a fundamental task of NLP, researchers have proposed several domain specific applications of ML- and DL-based frameworks. The main challenge encountered in machine-learning-based sentiment classification is the unmanageable amount of data. To address this challenge, [51] presents an ensemble learning (EL) approach for feature selection, which successfully aggregates several different feature selection results, so that we can obtain a more robust and efficient feature subset. Moreover, [52] also explores the predictive performance of different feature engineering schemes, four supervised ML-based algorithms and three EL-based methods obtaining

experimental results that yield higher predictive performance compared to the individual feature sets. Furthermore, in [53], the author presents yet another comprehensive analysis this time of keyword extraction approaches with empirical results that indicate an enhanced predictive performance and scalability of keyword-based representation of text documents in conjunction with EL-based models.

Sentiment analysis is a critical task of extracting subjective information from online text documents, mainly based on feature engineering to build efficient sentiment classifiers. To improve the feature selection process, [54] proposes and validates the effectiveness of a hybrid ensemble pruning scheme based on clustering and randomized search for text sentiment classification. Sentiment analysis can be reduced to a text classification problem. However, the text classification problem suffers from the curse of high dimensional feature space and feature sparsity problems. To mitigate and lift this curse, [55] explores several classification algorithms and EL-based methods on different datasets.

To recognize sentiment in information-rich but unstructured text, [56] presents a DL-based approach to sentiment analysis on product reviews with outperforming results. Since Twitter can serve as an essential source for several applications, including event detection, news recommendation, and crisis management, in [57], the author presents a DL-based scheme for sentiment analysis on Twitter messages with consistent and encouraging results.

ML- and DL-based models are at the core of NLP research. For instance, Onan [58] indicated that DL-based methods outperform EL-based methods and supervised ML-based methods for the task of sentiment analysis on educational data mining. And the list does not stop here. Onan [59] indicated that topic-enriched word embedding schemes utilized in conjunction with conventional feature sets can yield promising results for sarcasm identification. Onan [60] presented first usage of supervised clustering to obtain diverse ensemble for text classification and compare it to ML- and DL-based models. Onan and Toçoğlu [61] employed a three-layer stacked bidirectional long short-term memory architecture to identify sarcastic text documents with promising classification accuracy results. Onan [62] presented an extensive comparative analysis of different feature engineering schemes and five different ML-based learners in conjunction with EL-based methods.

### 3. Methodology

The main objective of our proposed approach is to optimize the performance of DCN on hyperspectral images. We developed a context-based feature augmentation approach to provide resistance against noise to deep learning classification of highly imbalanced hyperspectral images. The classification apparatus used in this study relies on a deep convolutional neuronet (DCN) to perform multi-class classification based on findings in [63]. The input to this network is a highly imbalanced hyperspectral image or cube. Figure 1 shows a hyperspectral cube. Figure 2 shows a 1-by-1 column along the spectral dimension.

Our proposed approach will be a preprocessing module in this classification apparatus as shown in Figure 3. Our

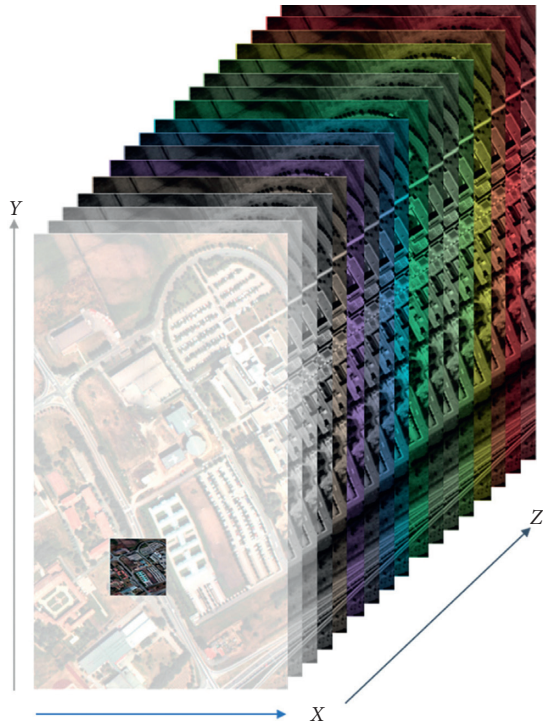


FIGURE 1: A hyperspectral image, where  $x$  and  $y$  are spatial dimensions and  $z$  is the spectral dimension.

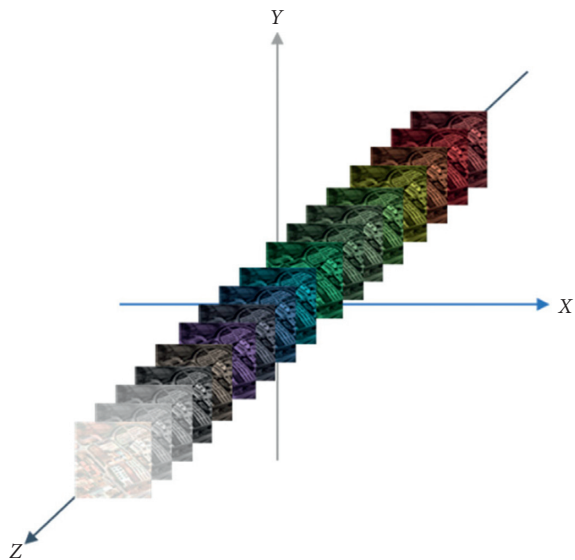


FIGURE 2: A hyperspectral column, where  $z$  is the spectral dimension.

four-step approach is introduced as follows. Full details are presented in Sections 3.1 through 3.2.

- (i) Local gradients are *feature vectors* of differences, defined in Section 3.1. In this step, we calculate these *feature vectors* for each pixel  $p$  in the hyperspectral

cube, as differences between the pivotal pixel  $p$  and its surrounding pixels in a 3-by-3-by-3 *local neighborhood*. This set of differences will constitute the *local gradients* of  $p$ .

- (ii) Reference clusters are *feature vectors* of high and low thresholds, defined in Section 3.2. In this step, we calculate these *feature vectors* for each pixel  $p$  in the hyperspectral cube, as statistical thresholds of the surrounding 9-by-9 *reference neighborhood*. This set of thresholds will constitute the *reference clusters* of  $p$ .
- (iii) Prototype contexts are *feature vectors* of similarity, defined in Section 3.3. In this step, we calculate these *feature vectors* for each pixel  $p$  in the hyperspectral cube, as the degree of membership of the *local gradients* to the *reference clusters*. This set of similarity degrees will constitute the *prototype contexts* of  $p$ .
- (iv) Concatenated features are all *feature vectors*, defined in Sections 3.1 and 3.2. In this step, we concatenate *local gradients*, *reference clusters*, and *prototype contexts* into one context-based *feature vector* for each pixel  $p$  in the hyperspectral cube.

**3.1. Calculate Local Gradients.** The first step of our approach is to calculate the *local gradients* [64]. Figure 4 shows a pivotal pixel  $p(1, 1, 1)$  in its 3-by-3-by-3 *local neighborhood*. The *local gradient*  $\chi$  is the set of gradient differences  $\{d_1, d_2, d_3, \dots, d_{13}\}$ , where  $d_i$  is the magnitude of the differences between  $p$  and its direct neighbors for each discrete direction  $i$ . For instance, in direction  $i=1$ ,  $d_1$  is equal to  $|p_{1,1,1} - p_{2,1,1}| + |p_{1,1,1} - p_{0,1,1}|$ , whereas, in direction  $i=10$ ,  $d_{10}$  is equal to  $|p_{1,1,1} - p_{2,2,2}| + |p_{1,1,1} - p_{0,0,0}|$ . Such *local gradients* are calculated for each pixel  $p_{i,j,k}$  within the hyperspectral cube.

It is important to note that this moving cubic-shaped *local neighborhood* only uses partial data around the borders of the hyperspectral image. Thus the indexes,  $i, j, k$ , will only run from 1 to the dimension length  $-1$  for each dimension  $x, y, z$ .

**3.2. Calculate Reference Clusters.** The second step of our approach is to calculate the *reference clusters* [64]. Figure 5 shows a pivotal pixel  $p(5, 5, 5)$  in its 9-by-9 *reference neighborhood*. The *reference clusters*  $\zeta$  is the sets of high and low thresholds  $\{hi_1, hi_2, hi_3, \dots, hi_{13}\}, \{lo_1, lo_2, lo_3, \dots, lo_{13}\}$ , where  $hi_i$  is the central value of the high-valued gradients and  $lo_i$  is the central value of the low-valued gradients within  $p$ 's *reference neighbors* for each discrete direction  $i$ . We calculate these central values using the mean  $\mu$  and variance  $\sigma^2$  equations presented in (1) and (2) to set  $hi = \mu + 2\sigma$  and  $lo = \mu - 2\sigma$ . Such *reference clusters* are calculated for each pixel  $p_{i,j,k}$  within the hyperspectral cube.



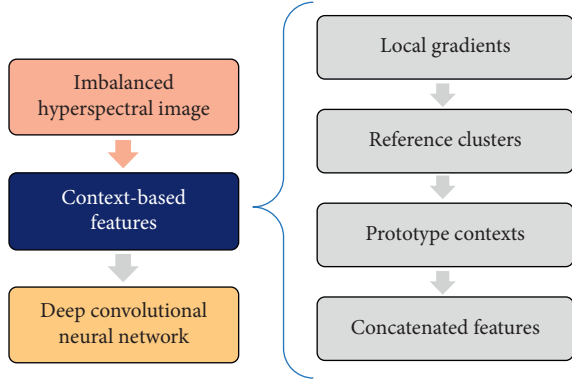


FIGURE 3: Overview of our deep learning hyperspectral classification apparatus.

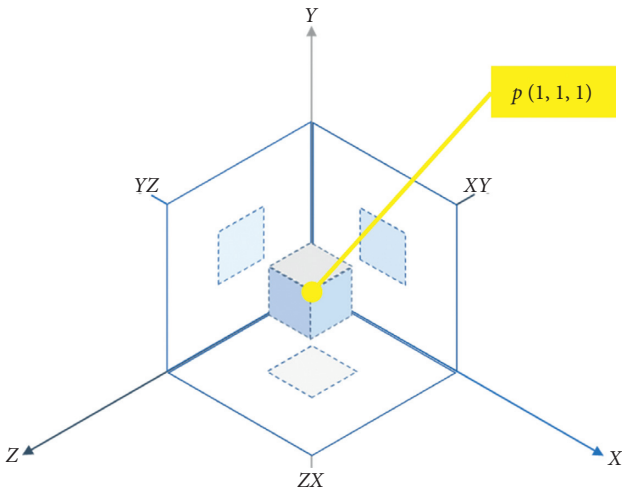


FIGURE 4: Pivotal pixel  $p$  inside its local neighborhood.

$$\mu_{i,j,k,d} = \frac{1}{N \times M} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{i+n,i+m,k,d}, \quad (1)$$

$$\sigma^2 = \frac{1}{N \times M} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (x_{i+n,i+m,k,d} - \mu_{i,j,k,d})^2. \quad (2)$$

It is important to note that this moving square-shaped *reference neighborhood* only uses partial data around the borders of the hyperspectral image. Thus the indexes,  $i, j$  will only run from 5 to the dimension length  $-5$  for each spatial dimensions. It will use however all the spectral bands on the  $z$  dimension.

**3.3. Construct Prototype Contexts.** The third step of our approach is to construct the *prototype contexts*. The *prototype contexts*  $\kappa$  is the sets of similarity features  $\{c_1, c_2, c_3, \dots, c_{13}\}$  where  $c_i$  is the *prototype context* with the highest degree of membership for each discrete direction  $i$ . We calculate this degree of membership  $M$  with the equation presented in (3)–(6) where  $D^2$  is the square of the Mahalanobis distance,  $\chi$  is the vector of local gradients,  $\kappa$  is the vector of prototype contexts,  $W$  is the inverse pooled covariance matrix, and the

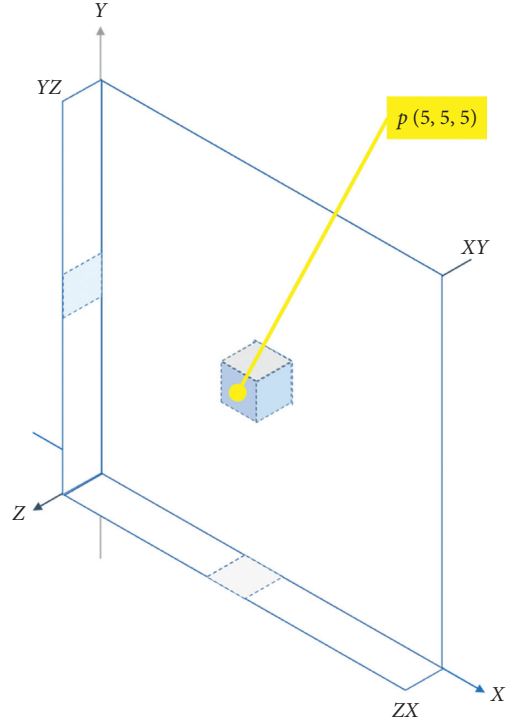


FIGURE 5: Pivotal pixel  $p$  inside its reference neighborhood.

$K$  factor is equal to the square root of the product between the highest value in  $\chi$  and the highest value in  $\kappa$ . Such *prototype contexts* are calculated for each pixel  $p_{i,j,k}$  within the hyperspectral cube.

$$M(\chi) = \max\left(0, 1 - \frac{D}{\sqrt{K}}\right), \quad (3)$$

$$D^2(\chi, \kappa) = (\chi - \kappa)^T W^{-1} (\chi - \kappa), \quad (4)$$

$$W(\chi, \kappa) = \frac{1}{2} \text{cov}(\chi) + \frac{1}{2} \text{cov}(\kappa), \quad (5)$$

$$K(\chi, \kappa) = \max(\chi) \times \max(\kappa). \quad (6)$$

**3.4. Concatenated Augmented Features.** The fourth step of our approach is to concatenate all *features vectors*. These *feature vectors* consist of the *local gradients*, *reference clusters*, and *prototypes contexts*. Such *context-based feature vectors* are concatenated for each pixel  $p_{i,j,k}$  within the hyperspectral cube.

Figure 6 shows how our context-based approach integrates into a deep learning classification model. Note that to evaluate the robustness of our approach, we added a synthetic noise to the original datasets. This noise was generated using a Gaussian equation. And classification accuracy was used as the main measurement to compare the performance of the model and in particular the resistance to noise in imbalanced hyperspectral images. Details are presented in the following section.

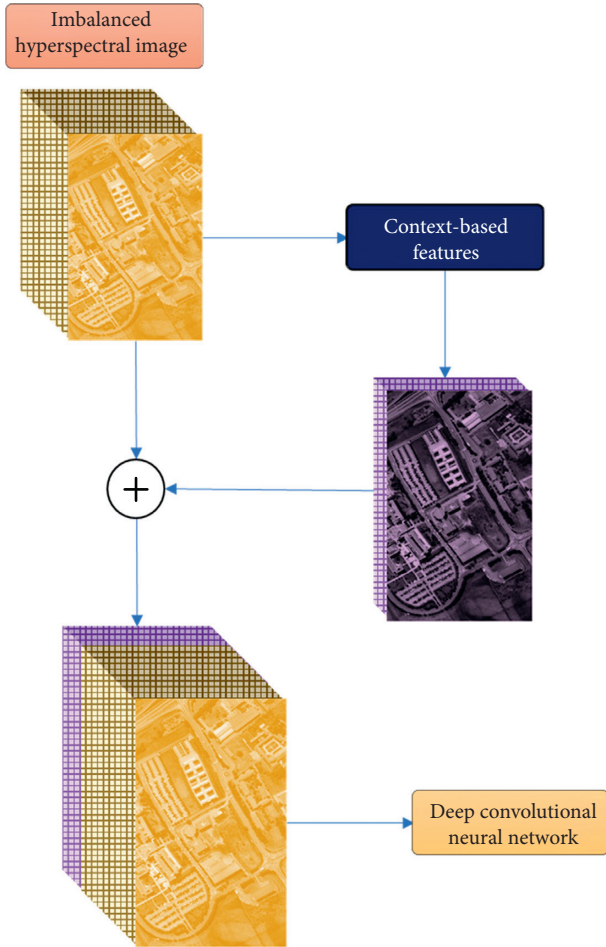


FIGURE 6: Overview of our approach as a preprocessing module.

## 4. Experiments

In this section, we describe the datasets, dataset partition policy, and experimental settings. Multiple settings are designed to evaluate the performance of our approach on noisy and clean data, as well as on imbalanced and balanced data.

**4.1. Datasets.** Four datasets were used in our experiments. The first two are the Pavia Centre and Pavia University datasets. These two datasets were acquired by the ROSIS sensor during a flight campaign over Pavia, Italy. The original Pavia Centre dataset is a hyperspectral cube with a spatial resolution of  $1096 \times 715$  and 102 spectral bands, and the original Pavia University dataset is a hyperspectral cube with a spatial resolution of  $610 \times 340$  spatial pixels and 103 spectral bands. The corresponding ground truths differentiate nine classes. For more details, please visit the following link. This link was last accessed on February 1, 2021 ([http://www.ehu.es/ccwintco/index.php/Hyperspectral\\_Remote\\_Sensing\\_Scenes#Pavia\\_Centre\\_and\\_University](http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Pavia_Centre_and_University)).

It is important to note that the Pavia Centre data are considered a balanced hyperspectral cube, whereas the Pavia University data are considered an imbalanced hyperspectral

cube. It is clear from Figure 7 that the Pavia Centre samples are evenly distributed between classes. But, in Figure 8, the majority of Pavia University samples belong to one single class, namely the class *Meadows*. Thus, this predominant class dwarfs minority classes, such as *Shadows*, *Bitumen*, and *Painted Metal Sheets*. This disparity is what makes Pavia University data imbalanced.

To evaluate the robustness of our approach, we added a synthetic noise to the original “clean” datasets and produced two additional synthetic datasets. Thus, together with the two clean datasets, two noisy datasets were used in our experiments, corresponding to the noisy Pavia Centre and the noisy Pavia University datasets. Identically to their clean counterparts, the noisy Pavia Centre dataset is a hyperspectral cube with a spatial resolution of  $1096 \times 715$  pixels, 102 spectral bands and 9 distinct classes, and the noisy Pavia University dataset is a hyperspectral cube with a spatial resolution of  $610 \times 340$  pixels, 103 spectral bands and 9 distinct classes.

To produce these noisy datasets, an intermittent irregular noise was incorporated. Equations (7)–(9) were used to generate a noise signal corresponding to a signal-to-noise value of  $\text{SNR}_{\text{dB}} = 120$ . In (7),  $G$  and  $F$  are random variables and  $N$  follows a Gaussian distribution with a probability density function presented in (8). Similarly to [65], this weighted random noise will follow a Gaussian normal distribution  $N(\mu, \sigma)$ , where the mean  $\mu$  is zero and the variance  $\sigma$  is determined from the signal-to-noise ratio ( $\text{SNR}_{\text{dB}}$ ) formula presented in (9).

$$G(a, b) \leftarrow F(m, n) + N(\mu, \sigma), \quad (7)$$

$$N(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad (8)$$

$$\text{SNR}_{\text{dB}} = 20 \log_{10} \frac{\mu_{\text{signal}}}{\sigma_{\text{noise}}}. \quad (9)$$

**4.2. Dataset Partition Policy.** Datasets were divided into training and testing sets; 80% of the data was used during the training (a.k.a. model-fitting) phase while the remaining 20% of the data was used for testing (a.k.a. model-prediction) phase. One-fourth of the training set was used as validation set during the fitting phase. Figure 9 shows the full-partition schema.

To rank our context-based DCN approach, two additional models are implemented: (i) a baseline deep learning approach, namely, DCN, and (ii) a benchmark approach, that is PCA + DCN. And classification metrics are used to evaluate and compare the performance and effectiveness of our approach.

**4.3. Baseline Experiments.** As a baseline, we observe the performance of a deep learning model without any preprocessing on the different hyperspectral datasets. Four types of experiments are included in this section. First, we work on clean data, running individual experiments for

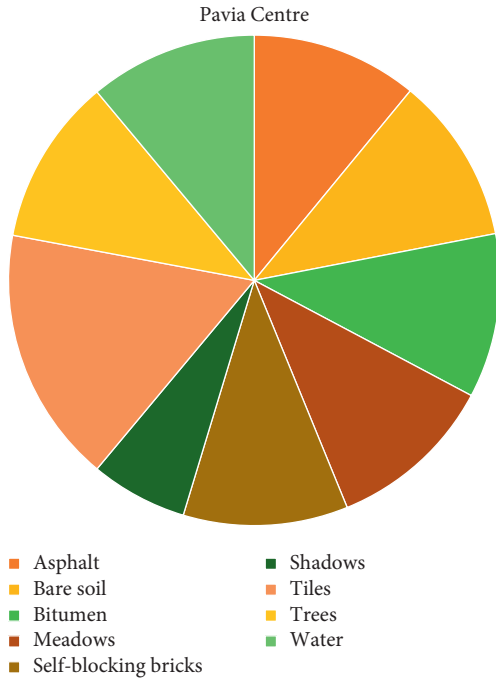


FIGURE 7: Class distribution for Pavia Centre. This dataset is considered balanced because for each class, there is relatively the same number of samples.

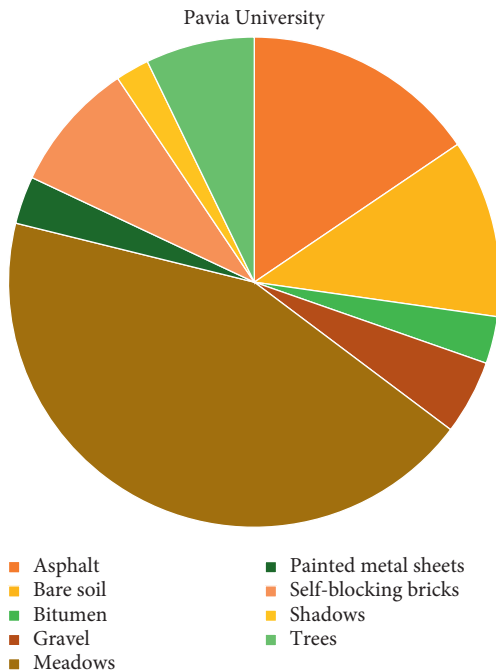


FIGURE 8: Class distribution for Pavia University. This dataset is considered imbalanced because for each class, there is not the same number of samples.

balanced and imbalanced datasets. Then, we focus on noisy data, and again we run individual experiments for balanced and imbalanced datasets.

A Deep Convolutional Neuronet (DCN) was used as a baseline to perform the classification. We used a DCN which

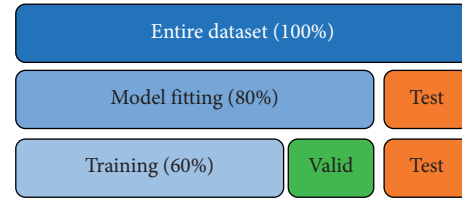


FIGURE 9: Partition policy: datasets are divided into 3 parts (20%, 20%, and 60%). The training task uses 60% of the samples. The validation task uses 20%. The testing task uses the remaining 20%.

consists of three types of layers, namely, input layer, hidden convolutional layer(s), and output layer. In Figure 10, the input dataset is shown as a cube. Similarly to [40], the hidden convolutional layers are shown as flat squares, the max-pooling layers in whiter color, and the dropout layer in pale. Straight lines are used to depict fully-connected layers or dense layers. Finally, for multiclass classification, the activation function is based on a softmax function.

During the model-fitting phase, we run for 20 epochs. At this point, the network achieves stability without running into overfitting. DCN used the two original datasets and the two noisy datasets. The results of our fitting phase are presented in Figures 11 to 14. The average classification accuracy on clean test data was  $86.1 \pm 3.9$  percent, whereas in noisy data was  $66.9 \pm 2.9$  percent. These results suggest an adversary effect of noise on our basic model.

**4.4. Benchmark Experiments.** As a benchmark comparison, we observe the performance of a deep learning model with noise reduction model as a preprocessing on the different hyperspectral datasets. Similarly, to the previous section, this section presents four types of experiments. First, we work on clean data, running individual experiments for balanced and imbalanced datasets. Then, we focus on noisy data, and again we run individual experiments for balanced and imbalanced datasets.

Principal Component Analysis (PCA) together with DCN was used as a benchmark to perform the classification. Ten principal components are sufficient to represent 99% variability of the data. Figure 15 shows the Scree Curves for both the Pavia Centre dataset in Figure 15(a) and the Pavia University dataset in Figure 15(b).

As suggested by the Scree Curves, PCA + DCN was implemented using only the first ten principal components. Twenty epochs were used during the model-fitting phase, a.k.a. training phase. In our experimental runs, the dataset partition policy was maintained the same and both the original datasets and the noisy datasets were randomly selected into training, validation, and testing sets.

The results of our fitting phase are presented in Figures 16 to 19. The average classification accuracy on clean test data was  $84.1 \pm 6.1$  percent, whereas on noisy data was  $37.3 \pm 4.7$  percent. Compared to the results for vanilla DCN, these results strongly suggest an adversary effect of noise on the principal component-based model. Another important point to analyze is that during training of PCA + DCN on

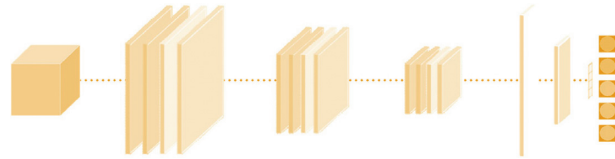


FIGURE 10: Overview of our deep convolutional neural network.

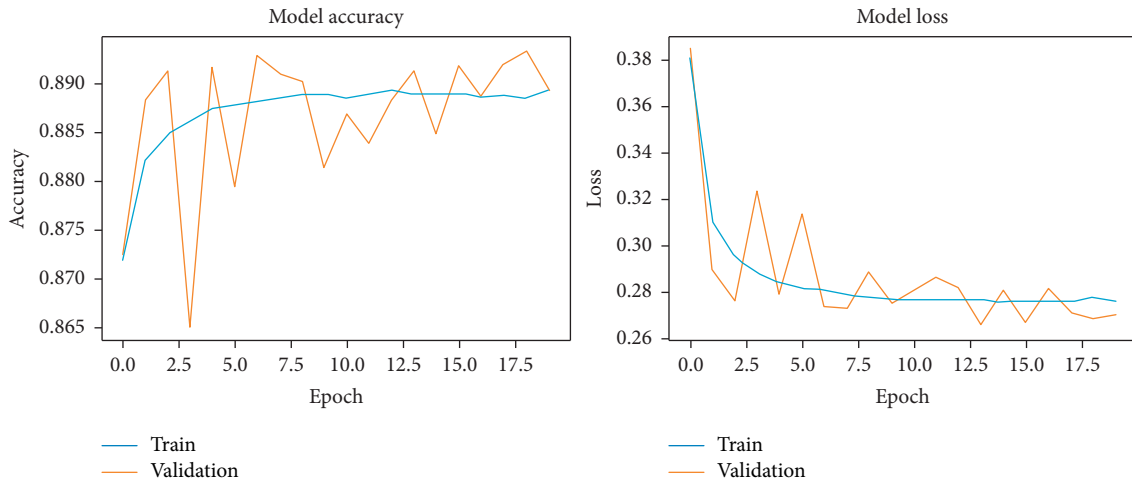


FIGURE 11: DCNN accuracy and loss during the model-fitting phase using the original Pavia Centre dataset.

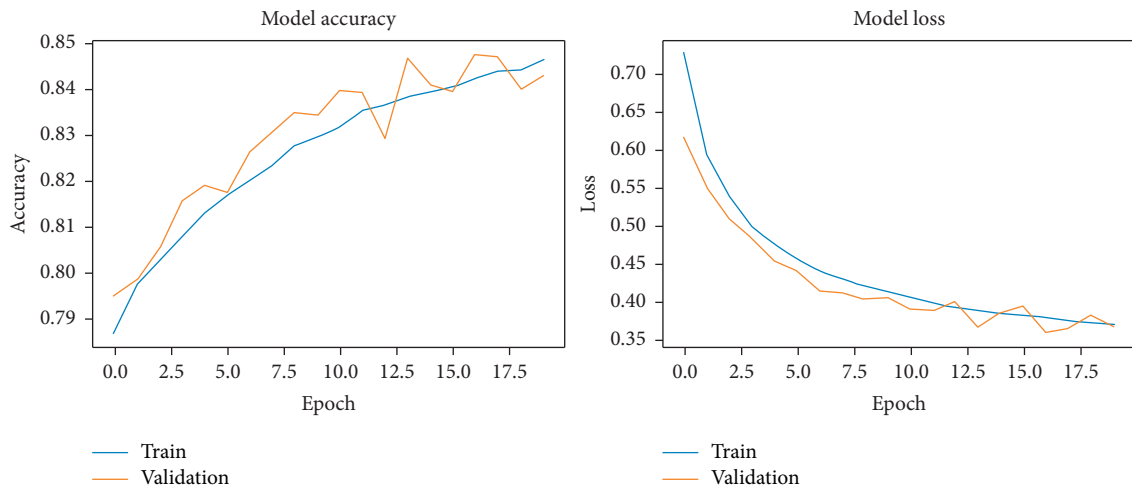


FIGURE 12: DCNN accuracy and loss during the model-fitting phase using the original Pavia University dataset.

noisy data, the model suffered from overfitting after the 4 epochs as shown in Figure 18.

**4.5. Enhanced Experiments.** We integrate our context-based feature augmentation module as a preprocessing step to the deep learning model. We observe the performance of a context-based deep learning model on the original highly imbalanced hyperspectral dataset. Then, we observe the

performance of our enhanced model in the presence of noise. We also run our context-based DCN for 20 epochs using the two original datasets and the two noisy datasets. All context-based features were used to achieve better noise resistance.

The results of the model-fitting phase are presented in Figures 20 to 23. The average classification accuracy on clean test data was  $87.5 \pm 3.4$  percent, whereas on noisy data was  $85.0 \pm 4.2$  percent. Compared to previous results, these

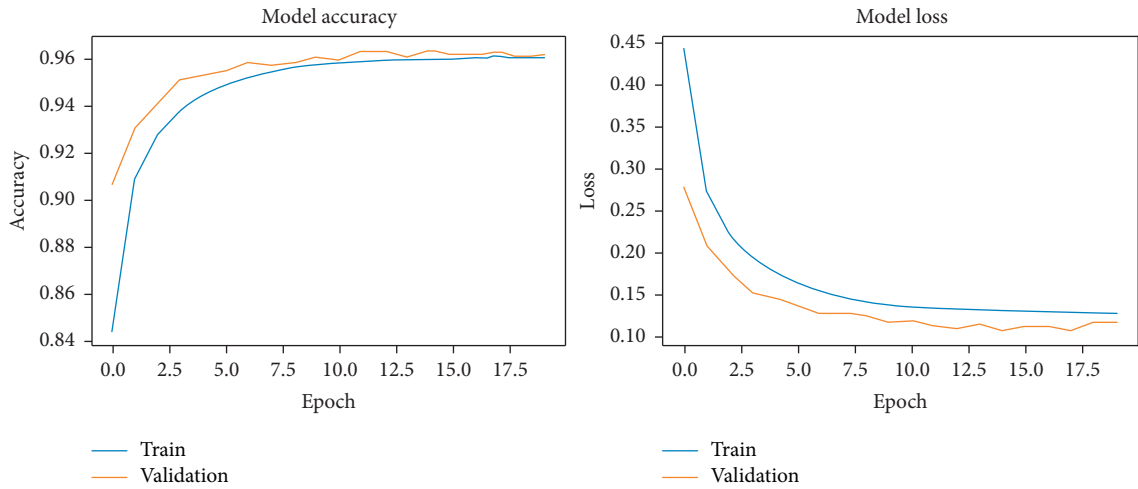


FIGURE 13: DCNN accuracy and loss during the model-fitting phase using the noisy Pavia Centre dataset.

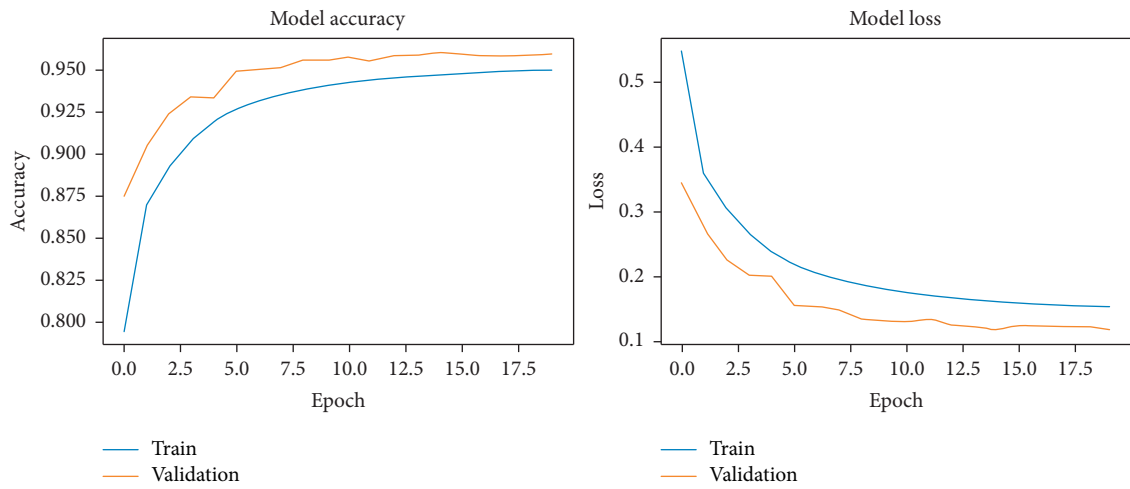


FIGURE 14: DCNN accuracy and loss during the model-fitting phase using the noisy Pavia University dataset.

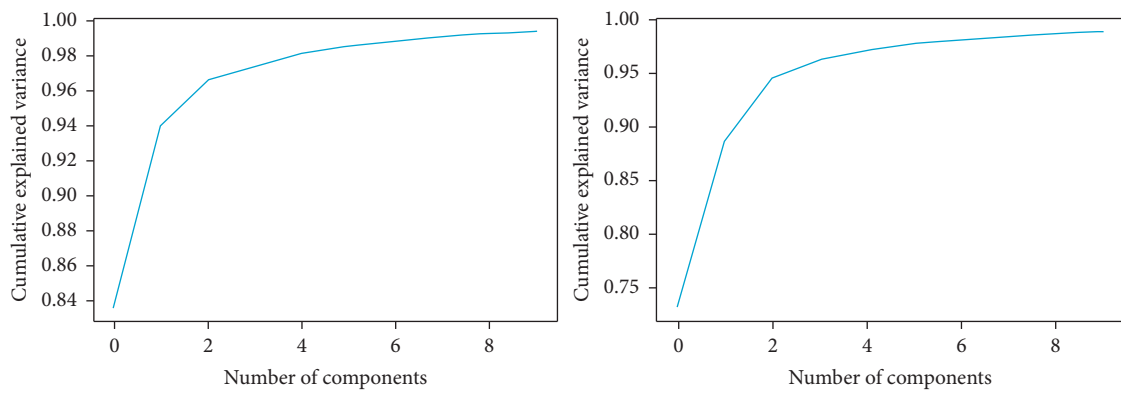


FIGURE 15: Scree curves for the (a) Pavia University dataset and (b) Pavia Centre dataset.

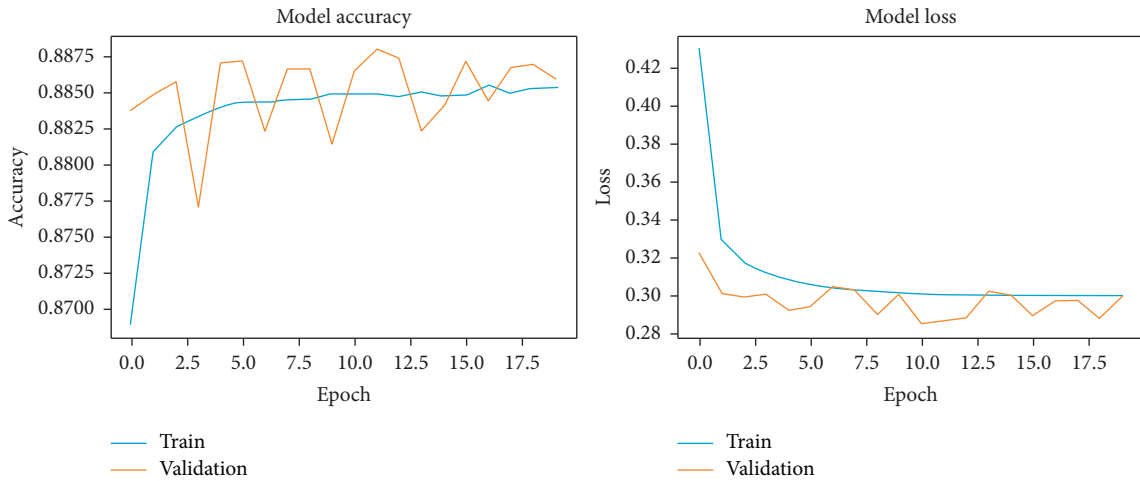


FIGURE 16: PCA + DCNN accuracy and loss during the model-fitting phase using the original Pavia Centre dataset.

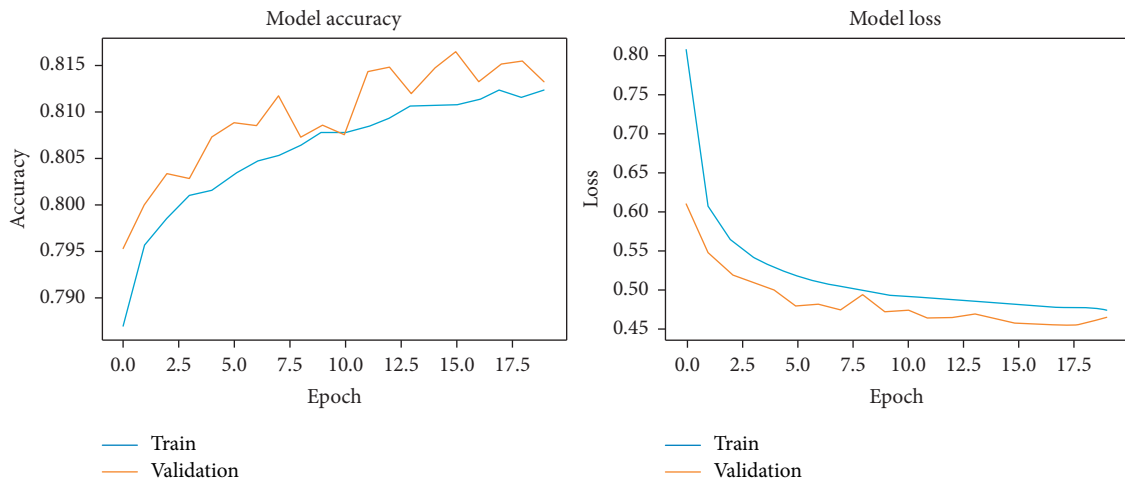


FIGURE 17: PCA + DCNN accuracy and loss during the model-fitting phase using the original Pavia University dataset.

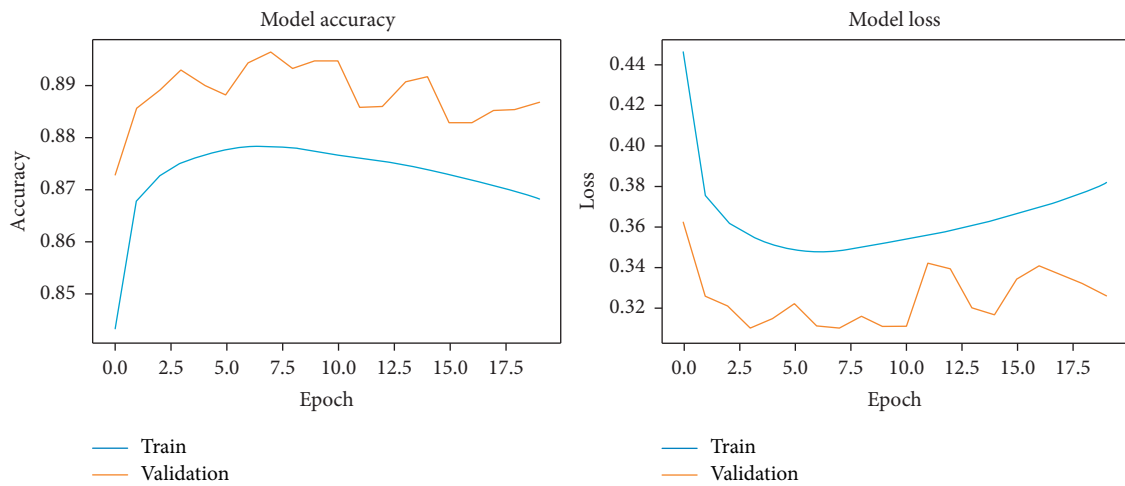


FIGURE 18: PCA + DCNN accuracy and loss during the model-fitting phase using the noisy Pavia Centre dataset.

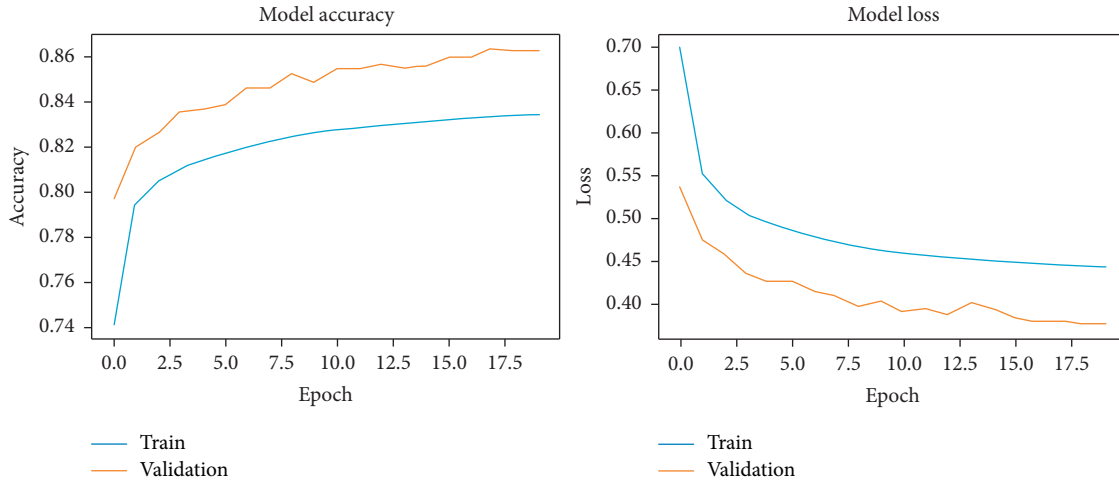


FIGURE 19: PCA + DCNN accuracy and loss during the model-fitting phase using the noisy Pavia University dataset.

percentages suggest that our proposed approach exhibits a high-level of accuracy on clean data and robustness against noise on both the Pavia University and the Pavia Centre datasets.

## 5. Results and Discussion

**5.1. Performance Metrics.** Receiver operating characteristic (ROC) curves are used to provide a graphical summary of the performance of our classification model. In this Cartesian plane graph, the  $x$ -axis denotes the False Positive Rate and the  $y$ -axis denotes the True Positive Rate. Thus, ROC curves depict False Positive Rate vs. True Positive Rate, where we have the following:

- (i) True Positive Rate is equal to True Positives (TP) divided by the addition of True Positives (TP) and False Negatives (FN), that is,  $TP/(TP + FN)$
- (ii) False Positive Rate is equal to False Positives (FP) divided by the addition of False Positives (FP) and True Negatives (TN), that is,  $FP/(FP + TN)$

Precision-Recall (PR) curves provide another graphical tool to evaluate performance of a classification model. In this Cartesian plane graph, the  $x$ -axis denotes the Recall and the  $y$ -axis denotes the Precision. Thus, PR curves depict Recall vs. Precision, where we have the following:

- (i) Recall is equal to True Positives (TP) divided by the addition of True Positives (TP) and False Negatives (FN), that is,  $TP/(TP + FN)$
- (ii) Precision is equal to True Positives (TP) divided by the addition of True Positives (TP) and False Positives (FP), that is,  $TP/(TP + FP)$

Finally, to compare the performance of each model dataset side by side, we compile a table using the ROC Area under Curve (AUC) Score for each model dataset. To this end, we used the following metrics:

- (i) Accuracy is equal to the quotation between the addition of True Positives and True Negatives

divided by the Total Population, that is,  $(TP + TN)/(TP + TN + FP + FN)$

- (ii) F1-score is equal to two times Precision ( $P$ ) times Recall ( $R$ ) divided by the addition of Precision ( $P$ ) and Recall ( $R$ ), that is,  $2PR/(P + R)$

**5.2. Prediction Results.** The following detail the classification results during the model-prediction phase. The following present the weighted averages for all performance metrics. First, Tables 1 and 2 present the classification results on the original, “clean datasets”, Pavia Centre and Pavia University, correspondingly. Then, Tables 3 and 4 present the classification results on the synthetic, “noisy datasets”, Pavia Centre with noise and Pavia University with noise, correspondingly.

Our experimental results suggest that all models suffer in the presence of noise, but the negative impact of noise can be mitigated with our proposed context-based approach. Tables 3 and 4 present the precision, recall, F1-score, and overall accuracy scores for DCN, PCA + DCN and our context-based DCN. Table 3 focuses on the noisy Pavia Centre dataset, while Table 4 focuses on the noisy Pavia University dataset. In both tables, we can observe that our proposed model achieves better results.

**5.3. Tabular Summary and Analysis.** Comprehensive summary tables are presented as follows. A total of three approaches were analyzed: a basic DCN with no preprocessing, a PCA + DCN, and a context-based DCN. They are listed on different rows. Four datasets were used: two without noise referenced as “clean data” and the same ones with random noise referenced as “noisy data”. Imbalanced datasets are listed on shaded columns of the tables. The values in each cell represent overall classification accuracy. Table 5 summarizes the overall accuracy of each model during the fitting/learning phase, whereas Table 6 summarizes the overall accuracy of each model during the testing/prediction phase.

It is important to note that during training on labeled samples as well as during testing on new samples, our proposed context-based DCN outperformed both DCN and

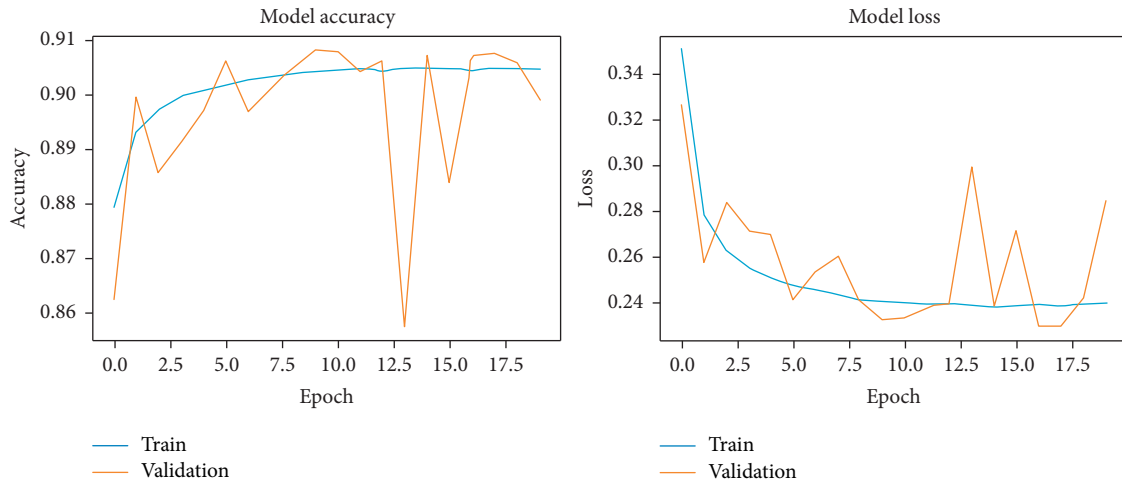


FIGURE 20: Context-based DCNN accuracy and loss during the model-fitting phase using the original Pavia Centre dataset.

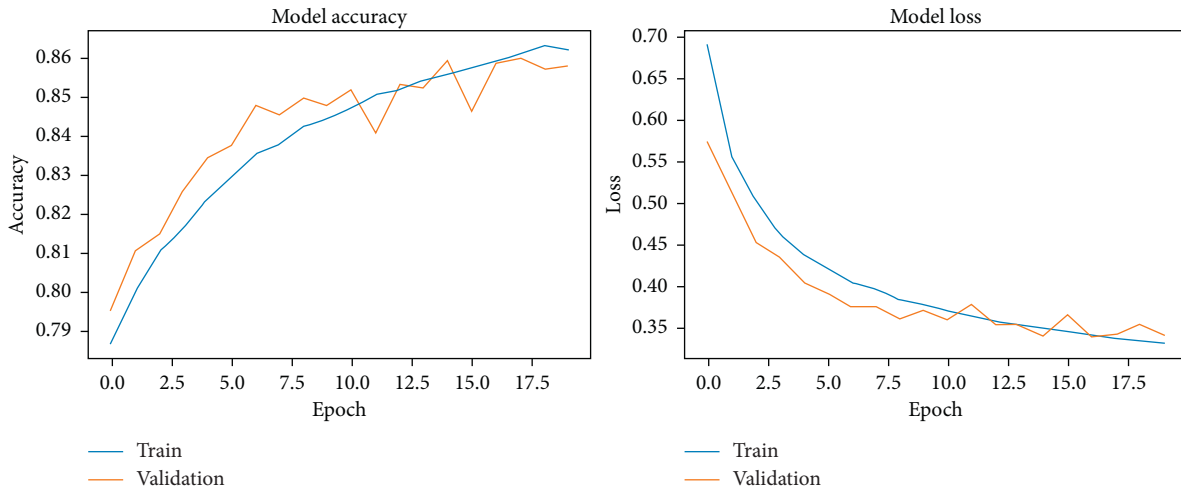


FIGURE 21: Context-based DCNN accuracy and loss during the model-fitting phase using the original Pavia University dataset.

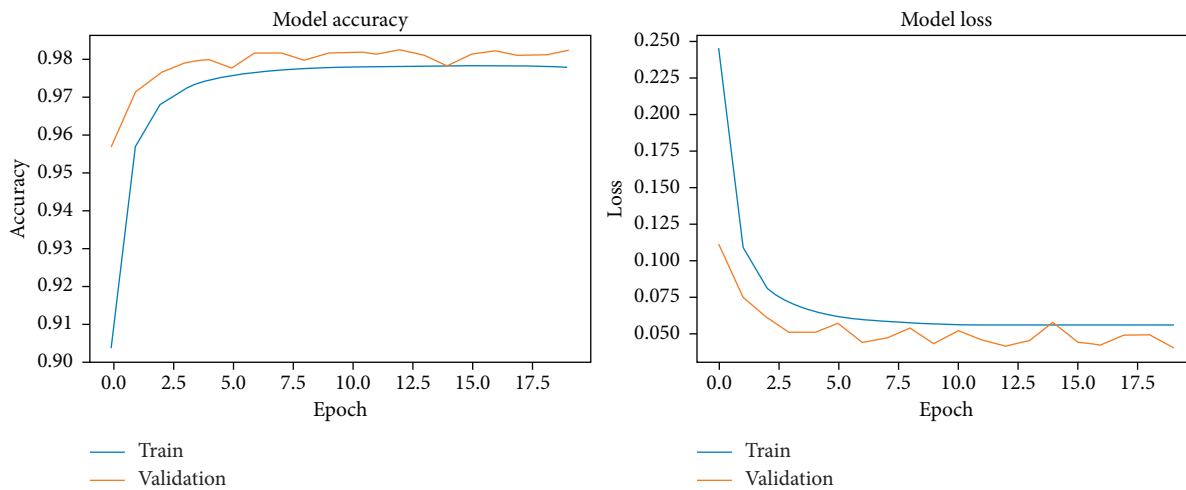


FIGURE 22: Context-based DCNN accuracy and loss during the model-fitting phase using the noisy Pavia Centre dataset.



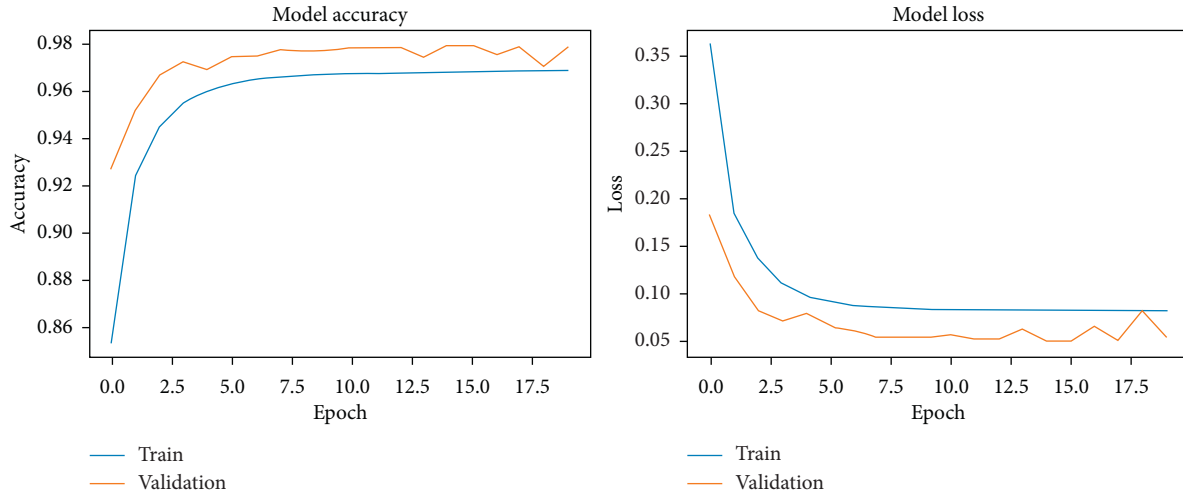


FIGURE 23: Context-based DCNN accuracy and loss during the model-fitting phase using the noisy Pavia University dataset.

TABLE 1: Model comparison based on prediction results using the original Pavia Centre dataset.

Models	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
DCN	86.70	89.15	85.11	88.92
PCA + DCN	79.71	88.72	83.82	88.52
Context-based DCN	88.35	89.95	88.05	89.88

TABLE 2: Model comparison based on prediction results using the original Pavia University dataset.

Models	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
DCN	83.99	83.16	83.08	84.28
PCA + DCN	80.68	79.89	78.37	81.29
Context-based DCN	86.37	85.00	85.50	85.78

TABLE 3: Model comparison based on prediction results using the noisy Pavia Centre dataset.

Models	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
DCN	85.97	65.14	69.00	68.98
PCA + DCN	84.70	34.10	37.26	40.62
Context-based DCN	86.37	82.14	83.40	88.01

TABLE 4: Model comparison based on prediction results using the noisy Pavia University dataset.

Models	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
DCN	89.72	67.81	73.22	64.79
PCA + DCN	89.45	40.86	46.02	33.93
Context-based DCN	89.48	88.59	88.50	81.99

TABLE 5: Highest validation accuracy during the training phase (model-fitting).

Models	Clean datasets		Noisy datasets	
	Pavia Centre (%)	Pavia University (%)	Pavia Centre w/noise (%)	Pavia University w/noise (%)
DCN	88.93	84.28	96.44	96.00
PCA + DCN	88.69	81.29	88.66	86.24
Context-based DCN	89.92	85.78	98.22	97.83

TABLE 6: Average accuracy score during the testing phase (model-prediction).

Models	Clean datasets		Noisy datasets	
	Pavia Centre (%)	Pavia University (%)	Pavia Centre w/noise (%)	Pavia University w/noise (%)
DCN	88.92	83.37	68.98	64.79
PCA + DCN	88.52	79.76	40.62	33.93
Context-based DCN	89.88	85.02	88.01	81.99

PCA + DCN, especially in the presence of random noise. PCA + DCN did not perform well for noisy cases because it was not able to remove our synthetic noise signal, which was not just random but also intermittent and irregular.

## 6. Conclusions

Hyperspectral imaging is an area of active research. Deep learning-based approaches to classification are the current state-of-the-art. However, our experimental results showed that in the presence of noisy hyperspectral datasets, these expert-level models underperform. To address this shortcoming, this paper presented a context-based feature augmentation approach to increase noise resistance in highly-imbalanced hyperspectral classification.

On noisy datasets, our robust approach outperformed a basic deep learning model and outclassed a combination of PCA and DCN approach. In addition, on highly-imbalanced noisy data, our context-based DCN approach suffered significant loss in terms of classification accuracy (less than 10%), whereas DCN and PCA + DCN suffered from an alarming 25% and 50% cuts in classification accuracy respectively.

Future lines of research should focus on applying our context-based approach to other noisy datasets in areas such as MRI and other highly imbalanced 3D medical images.

## Data Availability

The datasets used to support the findings of this study are available at [http://www.ehu.es/ccwintco/index.php/Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Science Foundation (NSF) (Grant no. 2011927), the United States Department of Defense (DOD) (Grant nos. W911NF1810475 and W911NF2010274), the National Institutes of Health (NIH) (Grant no. 1R25AG067896-01), and the United States Geological Survey and State Water Resources Research Institute Partnership (USGS-WRRI) (Grant no. 2020DC142B).

## References

- [1] V. N. Vapnik and A. Y. Chervonenkis, "On a perceptron class," *Automation and Remote Control*, vol. 25, no. 1, pp. 103–109, 1964.
- [2] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [3] J. F. Ramirez Rochac, L. Thompson, N. Zhang, and T. Oladunni, "A data augmentation-assisted deep learning model for high dimensional and highly imbalanced hyperspectral imaging data," in *Proceedings of the 9th International Conference on Information Science and Technology ICIST*, Kopaonik, Serbia, March 2019.
- [4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 2015.
- [5] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, June 2015.
- [6] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: integrated recognition, localization and detection using convolutional networks," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.
- [7] J. F. Ramirez Rochac, L. Liang, N. Zhang, and T. Oladunni, "A Gaussian data augmentation technique on highly dimensional, limited labeled data for multiclass classification using deep learning," in *Proceedings of the Tenth International Conference on Intelligent Control and Information Processing ICICIP*, Marrakesh, Morocco, December 2019.
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE TPAMI*, vol. 38, no. 1, pp. 142–158, 2015.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proceedings of the Computer Vision—ECCV 2014*, pp. 346–361, Zurich, Switzerland, September 2014.
- [10] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proceedings of the Computer Vision—ECCV 2014*, pp. 834–849, Zurich, Switzerland, September 2014.
- [11] J. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" *Advances in Neural Information Processing Systems*, vol. 2, pp. 1601–1609, 2014.
- [12] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to SIFT," 2014, <https://arxiv.org/abs/1405.5769>.
- [13] F. Feng Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano, "Toward automatic phenotyping of developing embryos from videos," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1360–1371, 2005.
- [14] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," *Advances in Neural Information Processing Systems*, vol. 25, pp. 2852–2860, 2012.
- [15] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on*

- Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.
- [16] PH. Pinheiro and R. Collobert, “Recurrent convolutional neural networks for scene labeling,” in *Proceedings of the 31st International Conference on Machine Learning*, pp. 82–90, Beijing, China, June 2014.
  - [17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, “Simultaneous detection and segmentation,” in *Proceedings of the Computer Vision—ECCV 2014*, pp. 297–312, Zurich, Switzerland, September 2014.
  - [18] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” in *Proceedings of the Computer Vision—ECCV 2014*, pp. 345–360, Zurich, Switzerland, September 2014.
  - [19] Y. Ganin and V. Lempitsky, “N4-fields: neural network nearest neighbor fields for image transforms,” in *Proceedings of the Asian Conference on Computer Vision*, pp. 536–551, Singapore, November 2014.
  - [20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
  - [21] G. B. Chavhan, P. S. Babyn, and B. Thomas, “Principles, techniques, and applications of T2\*-based MR imaging and its special applications,” *Radiographics*, vol. 29, pp. 1433–1449, 2009.
  - [22] N. Arora, A. K. Banerjee, and M. L. Narasu, “The role of artificial intelligence in tackling COVID-19,” *Future Virology*, vol. 15, no. 11, pp. 1–8, 2020.
  - [23] R. Furstenberg, C. A. Kendziora, J. Stepnowski et al., “Stand-off detection of trace explosives via resonant infrared photothermal imaging,” *Applied Physics Letters*, vol. 93, Article ID 224103, 2008.
  - [24] N. Audebert, B. Le Saux, and S. Lefevre, “Deep learning for classification of hyperspectral data: a comparative review,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 159–173, 2019.
  - [25] C. Xing, L. Ma, and X. Yang, “Stacked denoise autoencoder based feature extraction and classification for hyperspectral images,” *Journal of Sensors*, vol. 2016, Article ID e3632943, 2015.
  - [26] J. F. Ramirez Rochac and N. Zhang, “Feature extraction in hyperspectral imaging using adaptive feature selection approach,” in *Proceedings of the Eighth International Conference on Advanced Computational Intelligence ICACI*, pp. 36–40, Chiang Mai, Thailand, February 2016.
  - [27] M. Bahri, Y. Panagakis, and S. Zafeiriou, “Robust Kronecker component analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2365–2379, 2019.
  - [28] I. Jolliffe, *Principal Component Analysis*, Wiley, Hoboken, NJ, USA, 2005.
  - [29] M. Harandi, M. Salzmann, and R. Hartley, “Dimensionality reduction on SPD manifolds: the emergence of geometry-aware methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 48–62, 2018.
  - [30] L. Puggini and S. McLoone, “Forward selection component analysis: algorithms and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2395–2408, 2017.
  - [31] J. Zhou, C. Kwan, B. Ayhan, and M. T. Eismann, “A novel cluster kernel RX algorithm for anomaly and change detection using hyperspectral images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6497–6504, 2016.
  - [32] C. C. Olson and T. Doster, “A novel detection paradigm and its comparison to statistical and kernel-based anomaly detection algorithms for hyperspectral imagery,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 302–308, Honolulu, HI, USA, July 2017.
  - [33] A. V. Krysko, J. Awrejcewicz, I. V. Papkova, O. Szymanowska, and V. A. Krysko, “Principal component analysis in the nonlinear dynamics of beams: purification of the signal from noise induced by the nonlinearity of beam vibrations,” *Advances in Mathematical Physics*, vol. 2017, Article ID 3038179, 9 pages, 2017.
  - [34] C. Kwan and J. Zhou, “Method for image denoising,” US Patent 9,159,121, 2015.
  - [35] N. Zhang and K. Leatham, “A neurodynamics-based non-negative matrix factorization approach based on discrete-time projection neural network,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–9, 2019.
  - [36] J. F. Ramirez Rochac, N. Zhang, and P. Behera, “Design of adaptive feature extraction algorithm based on fuzzy classifier in hyperspectral imagery classification for big data analysis,” in *Proceedings of the 2016 12th World Congress on Intelligent Control and Automation WCICA*, Guilin, China, June 2016.
  - [37] LC. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and Y. AL, “DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.
  - [38] I. Aganj, M. G. Harisinghani, R. Weissleder, and B. Fischl, “Unsupervised medical image segmentation based on the local center of mass,” *Scientific Reports*, vol. 8, p. 13012, 2018.
  - [39] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, “Deep adaptive image clustering,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5879–5887, Venice, Italy, October 2017.
  - [40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1106–1114, 2012.
  - [41] N. Micallef, D. Seychell, and C. J. Bajada, “A nested U-net approach for brain tumour segmentation,” in *Proceedings of the IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, pp. 376–381, Palermo, Italy, June 2020.
  - [42] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “U-Net++: a nested U-net architecture for medical image segmentation,” *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, DLMIA 2018, ML-CDS 2018, Lecture Notes in Computer Science*, vol. 11045 Cham, Switzerland, Springer.
  - [43] F. Isensee, J. Petersen, A. Klein et al., “nnU-Net: self-adapting framework for u-net-based medical image segmentation,” 2018, <https://arxiv.org/abs/1809.10486>.
  - [44] C. Li, Y. Tan, W. Chen et al., “Attention U-Net++: a nested attention-aware U-net for liver CT image segmentation,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 345–349, Abu Dhabi, UAE, October 2020.
  - [45] F. Türk, M. Lüy, and N. Barışçı, “Kidney and renal tumor segmentation using a hybrid V-Net-Based model,” *Mathematics*, vol. 8, no. 10, p. 2020.
  - [46] A. Signoroni, M. Savardi, S. Benini et al., “Learning COVID-19 pneumonia severity on a large chest X-ray dataset,” *Elsevier, Medical Image Analysis*, vol. 71, Article ID 102046, 2021.
  - [47] F. Shi, J. Wang, J. Shi et al., “Review of artificial intelligence techniques in imaging data acquisition, segmentation, and

- diagnosis for COVID-19,” *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 4–15, 2021.
- [48] J. An, X. Zhang, H. Zhou, and L. Jiao, “Tensor-based low-rank graph with multi-manifold regularization for dimensionality reduction of hyperspectral images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, 2018.
- [49] K. Makantasis, A. D. Doulamis, N. D. Doulamis, and A. Nikitakis, “Tensor-based classification models for hyperspectral data analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, 2018.
- [50] P. G. Daneshmand, A. Mehridehnavi, and H. Rabbani, “Reconstruction of optical coherence tomography images using mixed low-rank approximation and second order tensor based total variation method,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 3, 2021.
- [51] A. Onan and S. Korukoğlu, “A feature selection model based on genetic rank aggregation for text sentiment classification,” *Journal of Information Science*, vol. 43, no. 1, pp. 25–38.
- [52] A. Onan, “Sentiment analysis on Twitter based on ensemble of psychological and linguistic feature sets,” *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 69–77.
- [53] A. Onan, “Ensemble of keyword extraction methods and classifiers in text classification,” *Expert Systems with Applications*, vol. 57, pp. 232–247.
- [54] A. Onan, S. Korukoğlu, and H. Bulut, “A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification,” *Information Processing & Management*, vol. 53, no. 4, pp. 814–833.
- [55] A. Onan, S. Korukoğlu, and H. Bulut, “LDA-based topic modelling in text sentiment classification: an empirical analysis,” *International Journal of Linguistics and Computer Applications*, vol. 7, no. 1, pp. 101–119.
- [56] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” *Concurrency and Computation: Practice and Experience*, p. e5909.
- [57] A. Onan, “Deep learning based sentiment analysis on product reviews on Twitter,” in *International Conference on Big Data Innovations and Applications*, pp. 80–91, Springer, Istanbul, Turkey, August 2019.
- [58] A. Onan, “Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach,” *Computer Applications in Engineering Education*, vol. 29, no. 3, pp. 572–589.
- [59] A. Onan, “Topic-enriched word embeddings for sarcasm identification,” *Software Engineering Methods in Intelligent Algorithms. CSOC 2019. Advances in Intelligent Systems and Computing*, Springer, pp. 293–304, Cham, Switzerland.
- [60] A. Onan, “Hybrid supervised clustering based ensemble scheme for text classification,” *Kybernetes*, vol. 46, no. 2, pp. 330–348, 2017.
- [61] A. Onan and M. A. Toçoğlu, “A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification,” *IEEE Access*, vol. 9, pp. 7701–7722.
- [62] A. Onan, “An ensemble scheme based on language function analysis and feature engineering for text genre classification,” *Journal of Information Science*, vol. 44, no. 1, pp. 28–47, 2016.
- [63] S. P. Sabale and C. R. Jadhav, “Hyperspectral image classification methods in remote sensing—a review,” in *Proceedings of the First International Conference on Computing Communication Control and Automation ICCUBEA*, pp. 679–683, Pune, India, February 2015.
- [64] J. F. Ramirez Rochac and N. Zhang, “Reference clusters based feature extraction approach for mixed spectral signatures with dimensionality disparity,” in *Proceedings of the 10th Annual IEEE International Systems Conference SYSCON*, pp. 1–5, Orlando, FL, USA, April 2016.
- [65] J. F. Ramirez Rochac, N. Zhang, J. Xiong, J. Zhong, and T. Oladunni, “Data augmentation for mixed spectral signatures coupled with convolutional neural networks,” in *Proceedings of the 9th International Conference on Information Science and Technology ICIST*, Kopaonik, Serbia, March 2019.

## Research Article

# Detection of Oil Spill Using SAR Imagery Based on AlexNet Model

**Xinzhe Wang,<sup>1</sup> Jiayu Liu,<sup>1,2</sup> Shuai Zhang,<sup>1,2</sup> Qiwen Deng,<sup>1,2</sup> Zhuo Wang,<sup>1</sup> Yunhao Li,<sup>2,3</sup> and Jianchao Fan<sup>2</sup>**

<sup>1</sup>*Institute of Information Science and Engineering, Dalian Polytechnic University, Dalian 116034, China*

<sup>2</sup>*Department of Marine Remote Sensing, National Marine Environmental Monitoring Center, Dalian 116023, China*

<sup>3</sup>*Institute of Geography Science, Liaoning Normal University, Dalian 116029, China*

Correspondence should be addressed to Jianchao Fan; [fjchaonmemc@163.com](mailto:fjchaonmemc@163.com)

Received 13 May 2021; Revised 19 June 2021; Accepted 24 June 2021; Published 6 July 2021

Academic Editor: Nian Zhang

Copyright © 2021 Xinzhe Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Synthetic aperture radar (SAR) plays an irreplaceable role in the monitoring of marine oil spills. However, due to the limitation of its imaging characteristics, it is difficult to use traditional image processing methods to effectively extract oil spill information from SAR images with coherent speckle noise. In this paper, the convolutional neural network AlexNet model is used to extract the oil spill information from SAR images by taking advantage of its features of local connection, weight sharing, and learning for image representation. The existing remote sensing images of the oil spills in recent years in China are used to build a dataset. These images are enhanced by translation and flip of the dataset, and so on and then sent to the established deep convolutional neural network for training. The prediction model is obtained through optimization methods such as Adam. During the prediction, the predicted image is cut into several blocks, and the error information is removed by corrosion expansion and Gaussian filtering after the image is spliced again. Experiments based on actual oil spill SAR datasets demonstrate the effectiveness of the modified AlexNet model compared with other approaches.

## 1. Introduction

Oil resources are the most important resources in the process of human industrialization. While the exploitation of marine oil enriches the oil resources, the marine oil spill caused by many factors has caused great harm to the environment. A large area of marine oil spill has caused a lot of economic losses, but at the same time, it has also caused great damage to the ecosystem. Regarding the Gulf of Mexico oil spill [1], Penglai 19-3 oil spill, and so on, those oil spill events caused damage to the local marine ecosystem and caused serious economic, ecological, and social impacts. Furthermore, the oil spill area will spread to other places with the current and wind and eventually affect a large area of the sea. Moreover, polluted marine organisms will enter the human body through the food chain, leading to a variety of diseases and even casualties [2]. Thus, the marine oil spill is one of the most serious problems of marine pollution in the world today. It should be monitored by an efficient and real-time method to extract timely information such as

location and area before the oil spill is spread over a large area [3].

Because of its own characteristics, remote sensing oil spill detection technology has been a hot research field in recent years. Because aerial remote sensing, satellite remote sensing, and other remote sensing monitoring methods have the characteristics of high timeliness, high resolution, large monitoring range, not affected by regional factors, image, and graphic data are easy to process and interpret. Remote sensing monitoring provides a lot of technical support for oil spill risk inspection, oil spill pollution monitoring, early warning, emergency response, oil spill ecological damage assessment, and remediation [4].

Remote sensing images represent the differences of different ground objects through the differences of brightness value or pixel value (reflecting the spectral information of ground objects) and spatial changes (reflecting the spatial information of ground objects) [5]. In remote sensing images, the background of oil spill and seawater are different in features such as grayscale, texture, shape, and brightness.

Therefore, oil spills can be identified by analyzing the feature changes of remote sensing images [6]. Early marine oil spill monitoring mainly through visual interpretation, through direct observation, or with the aid of auxiliary interpretation instrument to obtain specific target information in remote sensing images. Because the visual interpretation needs less equipment and is simple and convenient, it can obtain a lot of thematic information from remote sensing images at any time. Therefore, visual interpretation is the main method to interpret the image in the process of oil spill monitoring for a long time. However, the amount of remote sensing data is increasing year by year. Visual interpretation alone cannot meet the growing demand for monitoring. Moreover, the visual interpretation depends entirely on the experience of the interpreter, so interpretation errors are prone to occur. Therefore, computer vision is introduced into the oil spill remote sensing image monitoring. The computer or related equipment is used to simulate the biological vision, and the oil spill remote sensing image is processed to obtain the corresponding scene information. At present, an image segmentation algorithm is mainly used to extract information from oil spill remote sensing image or ENVI and another commercial remote sensing digital image processing software is used to classify sample pixels by built-in classification method [7].

The classification basis of the image segmentation algorithm is not unified, and the selection of segmentation algorithm largely depends on the shape of the image to be segmented, pixel distribution characteristics, and other factors, mainly divided into threshold segmentation, clustering segmentation, region growth, and so on.

Xu et al. applied the OTSU algorithm to oil spill monitoring [8]. Jin et al. used FCM to extract oil spill dark spots in SAR images [9]. Zou et al. used the SVM supervised classification method to complete the task of extracting oil spill information [10]. OTSU algorithm, also known as the maximum interclass variance method, was proposed by Japanese scholar OTSU in 1979. It assumes that the image to be processed only contains foreground image and background image and realizes image segmentation by calculating the threshold, which can make the maximum difference between the two types of pixels. There is also the optimal entropy threshold method proposed by Kaotur et al. and Ptile method proposed by Detcoyle et al. Clustering segmentation method is based on basic features such as grayscale pixels to divide the image according to certain rules. Then, the clustering method also developed the HCM clustering method based on a fuzzy theory proposed by RsuPini and fuzzy C-means (FCM) algorithm proposed by Dunn. The commonly used method is fuzzy C-means. The basic idea of the region growing segmentation method is to start from a group of growing points and merge the similar pixels until they cannot continue to grow.

Marine remote sensing techniques can be divided into laser fluorescence sensing, visible sensing, infrared, and microwave remote sensing [11]. The detection ability of the visible light sensor is limited due to the small contrast between the oil spill and the background. At this stage, only visible light sensor with high spatial resolution can detect oil

spill effectively. However, limited by the platform, it can only be carried out in limited scenarios. Compared with laser fluorescence sensor, visible light remote sensing, infrared remote sensing, and SAR in microwave remote sensing are not limited by weather, light, and other external conditions and can monitor the target all-weather, long-term, and real-time [12]. The SAR remote sensing image with good imaging conditions has the outstanding advantages of fast, all-day, all-weather, high precision, which can penetrate the surface and vegetation to obtain the information that optical photogrammetry is difficult to obtain. The visible light remote sensing image is greatly affected by the weather, and the factors such as light, cloud, and atmospheric particles will affect the remote sensing image. Therefore, SAR has many advantages over other methods in monitoring marine oil spill and other natural disasters. In general, the most important contribution of SAR in oil spill monitoring is that it cannot be affected by rainy or cloudy weather [13]. The classical segmentation algorithms mentioned in this paper have been relatively mature and stable, but most of them have the problems of a large amount of computation and time consumption. The number of classifications (except threshold segmentation) is affected by the image itself, and it is sensitive to noise and other factors. Therefore, most of them can only be used for optical remote sensing images acquired by visible light sensors, while the classification effect of SAR images is general and unstable [14].

In recent years, deep learning has attracted extensive attention in various fields. This method mainly uses neural network to supervise the learning of samples. Deep learning has been widely used in the field of object detection. Kwan et al. used YOLO to track and classify targets [15]. As a one-stage target detection algorithm, YOLO can directly predict the whole picture. Deep learning contains many algorithms, but the most representative one is the convolutional neural network, which can be traced back to 1962, Hubel and Wiesel's research on the visual system in the cat's brain [16]. Then, the neocognitron model was proposed by Kuniyuki Fukushima in 1979 and 1980. Neocognitron was a neural network with a deep structure, and it was one of the earliest deep learning algorithms. The first convolutional neural network was a time delay network proposed by Alexander Waibel in 1987 [17]. CNN is a kind of artificial neural network, and its weight-sharing network structure reduces the complexity of the network model and the number of weights. This advantage is particularly obvious when the input of the network is a multidimensional image. It can effectively learn the corresponding features from a large number of samples and extract the features better than the artificial design. Moreover, the larger the number of samples is, the better the extracted features are for classification and recognition. Meanwhile, the CNN structure has strong expansibility, and it can use a very deep number of layers. Therefore, training convolutional neural network for SAR image recognition can greatly reduce the interference caused by a lot of noise. At the same time, due to the stronger expression ability of the depth model, it has more advantages than the current mature algorithm in dealing with more complex classification problems such as remote sensing

image. Convolutional neural network is used to simulate the human brain's perception of image representation. With the increasing number of iterations and the application of the optimization algorithm, the model will have better robustness. When processing SAR image, it has better performance in the face of information interference brought by more complex sea conditions. Traditional semantic segmentation algorithms are based on artificially designed features to perform segmentation, which has poor accuracy and robustness. For remote sensing images, remote sensing images have rich features. Different remote sensing images may have different characteristics of oil spills. It is difficult to find a feature that can be used to segment remote sensing images accurately [18]. However, convolutional neural network can find a feature that can segment remote sensing image accurately. Convolutional neural network is an effective method for SAR image recognition. Compared with the traditional methods, this method has better robustness and generalization ability, and the accuracy is also improved.

In SAR images, oil spill can be identified from the perspective of features such as geometry, grayscale, and texture [19]. Moreover, for different SAR remote sensing images, their characteristics are also very different. It is difficult to find suitable features that can identify oil spills. Using convolutional neural networks can avoid the process of manually searching for features. It saves the time of manually searching for suitable features and can also improve detection accuracy. Inevitably, because the imaging principle of SAR is coherent microwave imaging, this imaging principle causes the existence of coherent speckle noise in SAR remote sensing images, and the existence of coherent speckle noise makes it particularly difficult to interpret SAR images [20].

Due to the existence of coherent speckle noise, it is difficult to classify each pixel in the SAR remote sensing image like semantic segmentation. Therefore, to reduce the interference of coherent speckle noise, the larger picture is divided into many smaller pictures, and then the convolutional neural network is used to classify the cropped smaller pictures and replace the classification of each pixel in semantic segmentation by classifying the smaller pictures after cropping. In short, it uses the classification of small images to segment the original image semantically. This method not only greatly reduces the interference of speckle noise on SAR remote sensing images but also improves the detection accuracy. Because the input image is a very small image, so using a shallow network has been able to meet the requirements. The model used in this paper is the classic network model AlexNet [21]. Considering the input image is a smaller image, the model is adjusted to fit the smaller image input.

The remainder of this paper is organized as follows. Section 2 describes the principle of convolutional neural network. The experimental methods are reported and discussed in Section 3. In Section 4, the experimental results are provided. Finally, the conclusion is given in Section 5.

## 2. Preliminary

The traditional unsupervised classification methods, such as image threshold segmentation and image edge extraction, are mainly based on the color features or texture features of the image. The extraction of image color features is to convert pixel values in digital images to corresponding values. Since color features are essentially pixel-based features, pixels in all regions of the image have corresponding contributions. As a global feature, color features are insensitive to changes in the direction and size of the image region and cannot well capture local features in the image. For texture features, this method is not like color features based on pixels but is calculated in areas containing multiple pixels. As a statistical feature, texture features usually have rotation invariance and are more resistant to noise. Texture feature is an efficient method for processing images with different thicknesses and densities. However, when the information difference between the thickness and density of the image is small, it is difficult to accurately reflect the difference between different textures perceived by human vision through texture features. When the wind wave is small, the noise is small, and the texture features of the image are relatively obvious. Images in areas with high winds and waves will appear like oil spills in texture and color. When the resolution of remote sensing images is high, this phenomenon is more obvious.

Convolutional neural network is usually composed of input layer, hidden layer, and output layer [22]. The input layer mainly performs some preprocessing operations on the picture, such as filtering and normalization. In this way, the model can be more robust.

The hidden layer usually includes the convolutional layer, the pooled layer, and the fully connected layer. The hidden layer is the key reason why CNN can extract the features of the images of the marine oil spill. Since the input image is a small image cut from the original large image, a simple convolutional neural network model can meet the requirements of the input image. For this reason, the classic AlexNet model is used. According to the size of the input image, the model is adjusted. The adjusted model includes five convolution layers: one pooling layer and three fully connected layers. After a series of convolution and pool operations, the input image is connected to the full connection layer, and two final prediction results are output, which corresponds to the score of the corresponding category. Figure 1 shows the adjusted AlexNet model.

Finally, there is the output layer. In this part, the classification labels are output by using logical functions or normalized exponential functions, or like semantic image segmentation, the output layer directly outputs the classification results of each pixel [23].

The core of the convolutional neural network is the hidden layer. The hidden layer mainly contains three parts, the convolutional layer, the pooling layer, and the fully connected layer.

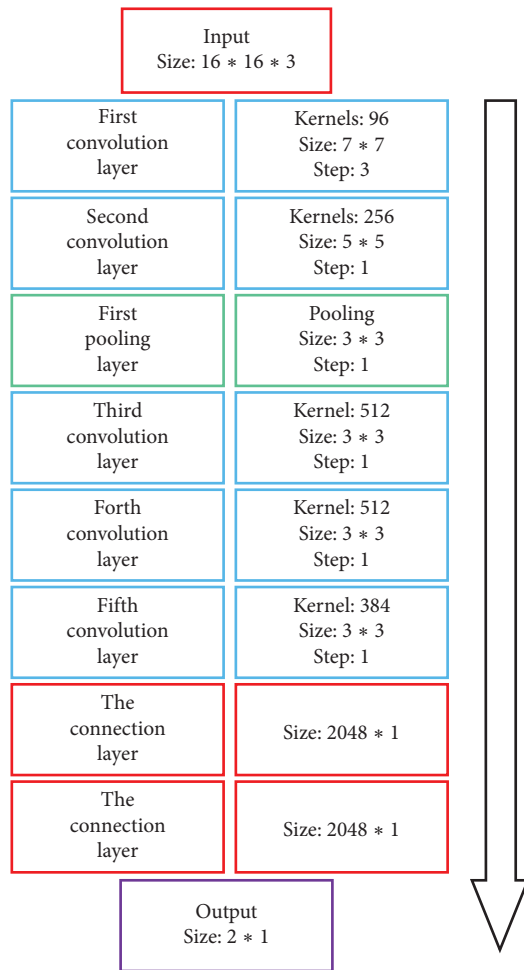


FIGURE 1: AlexNet network model.

The convolution layer is the set of a series of filters, and the output of the convolution layer is called the feature map. In the process of image processing, the extraction of image features can be regarded as the solution of feature vectors. The essence of an image is a matrix composed of pixel values. Decomposition of the eigenvalues of the image is extracting the features in the image. Among them, the convolution kernel can be regarded as the eigenvector set, and the backpropagation in the hidden layer can be considered as the process of solving the eigenvector set. CNN has a better effect on Marine oil spill processing because it has certain translation invariance and rotation invariance of the image. Compared with traditional classification methods, targets can still be accurately identified when affected by unpredictable environmental factors such as thermal noise and sea waves. In the neural network, the convolution kernel is defined as the feature detector at different positions. That is, no matter where the target appears in the image, it will detect these features and output the same response. The same is true for the pooling layer. For example, the maximum pooling will return the maximum value in the field if the maximum value has been moved, but in the field, the pooling layer will still output the same maximum value. Compared with texture feature extraction, texture feature has better

resistance to noise, although it usually has rotation invariance and has a good effect on processing images with great difference in thickness and density.

The pooling layer, also known as the lower sampling layer, can reduce the amount of data while retaining effective feature information. Due to the nature of the remote sensing image itself and the influence of uncertain factors such as environment, the model is prone to overfitting after processing a large amount of data. After dimensionality reduction and compression of the oil spill features that need to be extracted through the pooling layer, the overfitting will be reduced, and the fault tolerance of the model will be improved. This is especially true when extracting its features from ultrahigh resolution remote sensing images. Sampling can confuse the specific position of a feature. After a feature is found, only the relative position of this feature and other features can be needed to deal with the changes of similar objects caused by deformation and distortion.

The full connection layer is located at the last position of the whole network. From the perspective of representational learning, the full connection layer will conduct a nonlinear combination and output of the previously extracted features. That is, the full connection layer will not extract the features itself but use the extracted features at a higher stage to complete the final learning. ReLU [24] function is generally used for the excitation function of each neuron in the full connection layer. Since the ReLU function is an unsaturated nonlinear function, it can reduce the interdependence between parameters and alleviate the problem of overfitting [25]. For the problem of image classification, the output layer of the convolutional neural network generally uses the normalized exponential function to output the final classification label. Due to the small number of samples available for training oil spill images, overfitting is easy to occur. The dropout [26] operation can be introduced in the output layer to randomly delete the neurons of the neural network. Regularization and other operations can also be used to enhance the robustness of the model and reduce the phenomenon of overfitting so that the model can obtain higher accuracy in the prediction.

### 3. Oil Spill Detection Based on AlexNet

According to the principle of CNN, the extraction of the marine oil spill CNN can be roughly divided into four parts: dataset preparation, network model training, model testing, and model prediction. The specific process is shown in Figure 2.

In the preparation of the dataset, the noise generated in the SAR images in the process of imaging has the wind and waves and the ship, such as the impact of the target; scope of the different characteristics of the original data may have a very big difference, with the goal of the oil spill, features may be varied and very easy to appear in the process of training convergence, and accuracy is not high or gradient to vanish, so usually data in the input network before, need to input data standardization, namely before the convolutional neural network training data input, need in the channel or time/frequency d to normalization of data, on the image



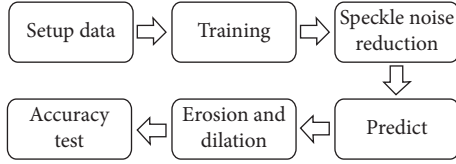


FIGURE 2: Flow chart of oil spill extraction.

processing. The original pixel values distributed in  $[0, 255]$  are usually normalized to the range  $[0, 1]$  so that the network model has a better ability to adapt to uncontrollable factors. At the same time, due to microwave coherent imaging characters, its special imaging mechanism will bring the interference of coherent speckle noise to the image. These images with coherent speckle noise will affect the training results of the model. It is a common practice to use speckle noise reduction methods before and after prediction, which can improve the detection accuracy. Some papers in the literature have incorporated such a practice in SAR image processing [27–32]. Therefore, before the image is input to the network, the noise reduction method is used to process the image to reduce the interference of speckle noise.

In order to increase the amount of training data and improve the generalization ability and robustness of the model, the satellite remote sensing image is enhanced by inversion, translation, and rotation operations [33].

All images of datasets are processed by the cropping method. Because the predicted oil spill area and the normal sea area appear in an image at the same time, the loss of information may occur after the predicted image is restitched. Therefore, the cutting step size during cutting and splicing is smaller than the cutting size. When using the cropping method to crop a picture, if the oil spill area in the cropped picture accounts for more than 60% of the total picture, the picture will be judged as an oil spill. Otherwise, it is not judged. As shown in Figure 3,  $4 \times 4$  extraction with a step size of 2 is carried out for the region of  $7 \times 7$ . The features extracted by the convolutional neural network only retain  $4 \times 4$  information after the stitching is completed. In this way, the interference between the oil spill and the image of the normal sea area is reduced, and the edge of the detection area is smoothed. At the same time, because the image size is smaller than the step size in cutting and stitching, the prediction of the following image can be used to verify the prediction result of the previous image. This method not only improves the detection accuracy but also reduces the interference of speckle noise.

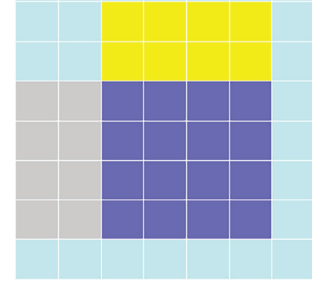


FIGURE 3: Diagram of cutting method.

In the process of the processed oil spill image entering CNN, the features are usually extracted by convolution kernel, and the features are compressed, and further dimensionality is reduced through the pooling layer [34]. These randomly initialized convolution kernels will be updated continuously through backpropagation and will approach the real solution after several iterations. The essence of this method is not to solve the whole image but to iterate out a set of feature vectors consistent with a certain distribution through the backpropagation algorithm, and this set of feature vectors infinitely approximates the conceptual feature vectors so that we can use the mathematical method of feature vectors to solve the image matrix. Therefore, CNN has the advantage that traditional feature extraction methods do not have when processing images based on remote sensing images, such as marine oil spill, whose features are difficult to separate. The convolution layer usually extracts the features of the input image through the convolution operation. The lower convolution layer extracts the relatively low-level features, such as the edges, lines, and corners of the image, while the higher the convolution layer extracts, the more advanced the features. The inner part of the convolution layer is composed of multiple convolution kernels, and each element of convolution kernels corresponds to a weight and a deviation vector. Each neuron in the convolutional layer is connected to multiple neurons in the adjacent layer above. When the convolution kernel is working, it will regularly sweep the input features and perform matrix multiplication on the input features in the receptive field and superposition of the deviation vector. The specific mathematical expression is shown as follows [35]:

$$Z^{(l+1)}(i, j) = [Z^l \otimes w^{(l+1)}](i, j) + b, \quad (1)$$

$$[Z^l \otimes w^{(l+1)}](i, j) + b = \sum_{k=1}^{K_i} \sum_{x=1}^J \sum_{y=1}^J [Z_k^l(s_0 i + x, s_0 j + y) w_k^{(l+1)}(x, y)] + b. \quad (2)$$

In the formula,  $b$  is the deviation vector,  $Z^l$  and  $Z^{l+1}$  represent the convolution input and output of  $l+1$  layer,

also known as feature map, and  $L^{l+1}$  is the size of  $Z^{l+1}$ . It is assumed that the feature map has the same length and width.

$Z(i, j)$  corresponds to the pixels of the feature graph, where  $(i, j) \in \{0, 1, \dots, L_{l+1}\}$ ,  $L_{l+1} = (L_l + 2p - f/sn) + 1$ .  $K$  is the number of channels of the feature graph.  $f$ ,  $s_0$ , and  $p$  are parameters of the convolution layer; they correspond to the size of the convolution kernel, the stride of the convolution, and the number of padding layers.

After the convolution layer, the output feature map is usually sent to the pooling layer for subsampling. The mathematical representation of  $L_p$  pooling is shown as follows [36]:

$$A_k^l(i, j) = \left[ \sum_{x=1}^f \sum_{y=1}^f A_k^l(s_0i + x, s_0j + y)^p \right]^{(1/p)}. \quad (3)$$

In the formula, step size  $s_0$  and pixel  $Z(i, j)$  have the same meanings as the convolution layer, and  $p$  is the prespecified parameter. When  $p = 1$ ,  $L_p$  pooling takes the mean value in the pooling region, which is called mean pooling. When  $p$  goes to infinity,  $L_p$  pooling takes the maximum value in the pooling region, which is called maximum pooling. The pooling layer aims to obtain spatially invariant features by reducing the resolution of the feature surface, and the pooling layer plays the role of secondary feature extraction. The pooling layer also plays the role of reducing computational complexity.

The model will be tested after training is completed every epoch. The difference between the prediction process and the final big image prediction process is that at this stage, only the small images are predicted, and the best time for model training is found by comparing the predicted results. The main purpose of this part is to measure the quality of the model.

The model prediction stage is mainly divided into 4 steps when predicting real remote sensing images. In the first step, the preprocessing stage, we will perform filtering operations on the picture to reduce the interference of coherent speckle noise in the picture. In the second step, the image is cropped according to the size of  $16 * 16$  and the step size of 7. The third step is model prediction. The cropped images are sent with a size of  $16 * 16$  to the trained AlexNet network for prediction. The fourth step is to generate a mask image based on the prediction result because oil spills usually exist continuously and in pieces on the sea. Therefore, a small oil spill area in the prediction result is likely to be the result of the wrong prediction. So, the oil spill area with the wrong prediction can be removed by corrosion expansion. In order to improve accuracy and reduce errors, a corrosion expansion operation is performed. This can reduce the interference to the prediction result due to the prediction error and then maps the result back to the original image to obtain the final predicted result. In this part, the adjusted AlexNet model plays the most important role, extracting features, and classification, through the classification results to determine whether it is oil spill, to generate mask image. Figure 4 shows the detailed forecast flow chart.

In order to measure the error between the measured data and the manual calibration data, the difference between the binary image and the manually calibrated truth graph was

calculated, that is, (the number of same pixels)/(the total number of pixels) \* 100%, which is the accuracy. At the same time, the kappa index based on the confusion matrix is used. The kappa coefficient reflects the accuracy of classification. Under normal circumstances, the kappa coefficient range is 0-1, divided into five groups, which are very low agreement (slight) between 0.0 and 0.2 and general agreement (fair) between 0.21 and 0.40, moderate consistency between 0.41 and 0.60, substantial consistency between 0.61 and 0.80, and almost perfect between 0.81 and 1. When calculating the kappa coefficient, we need to get four data, which are the basis of the confusion matrix. They are true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP and TN mean that the predicted category of the pixel is similar to the actual category, but TP is the predicted positive category, and TN is the predicted negative category. FP and FN indicate that the predicted pixel category is not the same as the real category. The difference is that FP is the predicted positive category, while FN predicts the negative category. After obtaining the four data,  $P_0$  can be obtained by (4), which is the accuracy. It reflects the correct proportion of the classification. Then,  $P_e$  is obtained by (5), which can be called bias index.  $P_e$  is the product of the actual and predicted quantity/the square of the total number of samples. It reflects the balance of the results. The higher it is, the more unbalanced the confusion matrix is. And finally, kappa is obtained by combining (6). It is the evaluation index of reaction consistency standard. Also, oil spill detection aims to identify the oil spill area accurately, so the evaluation standard of recall rate shown in (7) will be used, reflecting the proportion of predicted positive samples to actual positive samples:

$$P_0 = \frac{tp + tn}{tp + tn + fp + fn}, \quad (4)$$

$$P_e = \frac{(tn + fn) * (tn + fp) + (fp + tp) * (fn + tp)}{(tp + tn + fp + fn)^2}, \quad (5)$$

$$\text{kappa} = \frac{P_0 - P_e}{1 - P_e}. \quad (6)$$

$$\text{recall} = \frac{tp}{tp + fn}. \quad (7)$$

## 4. Experiments

In order to ensure the validity of this experiment, the images selected in the experiment are all SAR images with the real oil spill events.

**4.1. Data Processing.** The image is cropped by Photoshop software, and the  $2109 \times 2109$  resolution image of some sea areas is intercepted to establish the training set. The cropped  $2109 \times 2109$  oil spill image is cropped into multiple parts according to the method of length 16, width 16, and Step 7. In other words, the  $2109 \times 2109$  resolution image is cropped into 300 lines  $\times$  300 columns, and the single image size is

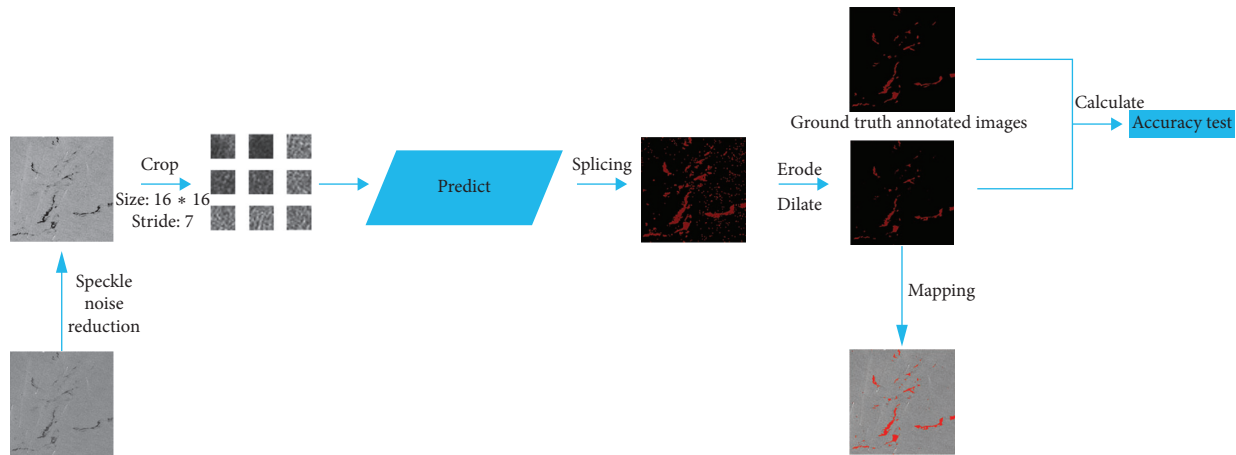


FIGURE 4: Oil spill prediction flow chart.

$16 \times 16$  resolution image. When making the dataset, if more than 60% of the image belongs to the oil spill area, the picture is judged to be oil spill. Otherwise, it is not. During Mosaic, part of the information of a single image will be overwritten by the next adjacent image. This can improve the accuracy of detection. When making labels, if more than 60% of the part in the image belongs to the oil spill area, it is classified as the oil spill part, and the rest is classified as the nonoil spill part. After image enhancement, 90% of the labeled  $16 \times 16$  datasets (17,100 images in total) were randomly selected as training samples. A total of 1,710 images of 10% were used as test samples, and the test samples and training samples were not duplicated. Before the image is input, in order to reduce the interference caused by noise, the convolution kernel size is 3, and the mean filtering method is adopted to smooth the image. Some training samples are shown in Figure 5.

As for the model, the basic learning rate is 0.0005, and the input size is adjusted to  $16 \times 16 \times 3$ . Every 50 epoch, the learning rate is adjusted once, and the adjustment magnification is 0.99, that is, adjusted to 0.99 times the previous time. The network consists of 8 layers, and the five layers are the convolution layer. After the first and second convolution layer, a maximum pooling layer is used to extract the features of the image. The last three layers are fully connected layers, which are used to classify the extracted features without updating the parameters. In order to further improve the generalization ability of the model,  $L_2$  regularization and dropout were used in the full connection layer during training, and the dropout parameter was set to 0.6. Table 1 shows the detailed parameter settings of the model.

The algorithm of AlexNet is compared with OTSU, SVM, and FCM. Before the prediction, the parameters of the algorithm are set. Table 2 shows the parameter setting of the FCM algorithm, and Table 3 shows the parameter setting of the SVM algorithm.

**4.2. Result.** The following results are from the same hardware condition. The processor is Intel(R) Core (TM) i5-10200H CPU @2.40 GHz 2.4 GHz, memory 16 GB.

There are many denoising algorithms that can deal with speckle noise, such as frost filtering and median filtering. Before image prediction, filtering the image can reduce the interference of speckle noise on the image and improve the accuracy of detection. Different denoising algorithms will produce different results. In order to select an appropriate denoising algorithm, different filtering algorithms are used for the same image, and then AlexNet is used for detection. In the detection process, only the filtering algorithm before detection is changed, and other parameters remain unchanged. Finally, recall, accuracy, and kappa coefficient are used to evaluate the results. Recall rate is the evaluation of positive samples, and oil spill detection is more concerned about the recognition effect of the oil spill area, so the use of recall rate can better compare the effect of different noise reduction algorithms. The specific results are shown in Table 4. It is not difficult to find from the table that after filtering the image, the detection accuracy is improved compared with no filtering operation, and the highest detection recall is the Lee-Sigma filter, so the Lee-Sigma filter is selected as the filtering algorithm used before image prediction.

Figure 6 shows the loss curve and precision curve. Figure 6(a) shows the loss function and precision curve in the training process, and Figure 6(b) shows the loss function and precision curve in the verification process. It is not difficult to find that in the training process, the model's loss function is close to 0, and the accuracy is close to 1, which indicates that the model has well fitted the data of the training set. In the verification set, with the increase of epoch, the model's loss function is close to 0. The accuracy is relative to 0.96-0.97, which shows that the model has good robustness and generalization ability. The network can still achieve higher accuracy and lower loss value in few iterations at the current depth. The survey surface proves that such excellent features as CNN weight sharing greatly reduce the calculation amount during training.

In the experimental comparison results, the predicted image is obtained after corrosion expansion and smooth filtering. By macroscopic comparison between the extracted

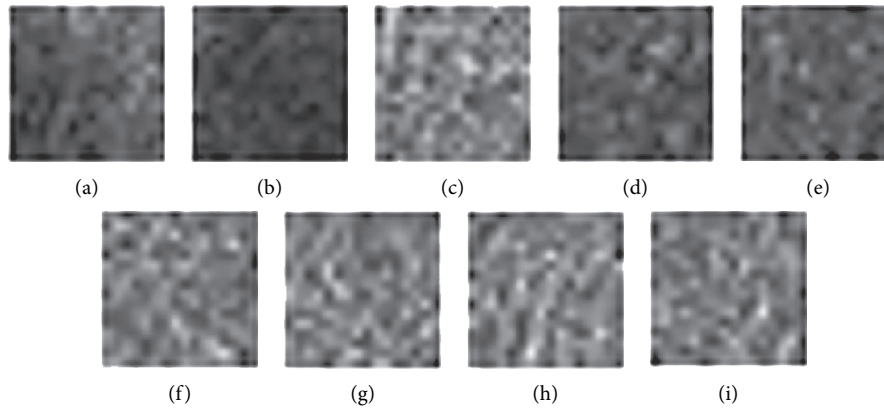


FIGURE 5: Partial training sample examples.

TABLE 1: AlexNet parameters configuration.

Epoch	2500
Batch size	256
Learning rate	0.0005
Learning rate decay	0.99
Dropout	0.6
Weight decay	0.0001

TABLE 2: FCM parameters configuration.

Number of categories	2
Maximum number of iterations	50
Threshold of membership change	1e-5
Class center change threshold	1e-5

TABLE 3: SVM parameters configuration.

Kernel type	Radial basis function
Gamma in kernel function	0.333
Penalty parameter	100
Pyramid levels	0
Classification probability threshold	0

TABLE 4: Results of different filtering methods.

	Recall (%)	Accuracy (%)	Kappa
No filtering	97.88	98.52	0.78
Frost filtering	98.2	98.73	0.80
Gamma-MAP filtering	98.24	98.64	0.79
Local filtering	97.94	98.68	0.80
Mean filtering	98.26	98.58	0.79
Median filtering	98.13	98.60	0.79
Lee filtering	98.19	98.71	0.80
Lee-Sigma	98.3	98.57	0.79

results (only after corrosion expansion treatment) and the original image, the network extraction is more accurate. Some areas that have not been accurately calibrated by human calibration can still be extracted through the network. Two criteria, accuracy and kappa coefficient, are used for image segmentation evaluation. Figures 7 and 8 are two examples of detection using different methods. They are

called Sample 1 and Sample 2, respectively. Samples 1 and 2 compare the results using AlexNet, Otsu, FCM, and SVM methods, respectively. In order to maintain the fairness of the result, the same operation is performed on the predicted image before and after the prediction.

Figures 7 and 8 show the processing results of the different oil spill images by different methods. The red part is the oil spill area, and other areas are nonoil spill areas such as sea or land. Figures 7(a) and 8(a) are the original images of two different oil spill images, respectively, while Figures 7(b) and 8(b) are the results of manual annotation of these two images, respectively. Figures 7(d), 7(e), 8(d), and 8(e) show the extraction results of the same oil spill area using traditional unsupervised algorithms such as OTSU and FCM, respectively. Due to the influence of speckle noise, many speckle noises are identified as oil spill areas, and the oil spill part cannot be accurately extracted. Figures 7(f) and 8(f) show the extraction results of the SVM algorithm commonly used in supervised learning, which has high accuracy in visible light remote sensing classification. However, the algorithm is still seriously affected by noise and cannot obtain a clean oil spill area through corrosion expansion filtering. Figures 7(c) and 8(c) are the oil spill results extracted by the improved AlexNet method. It is not difficult to find that the image extraction effect is better than the other three methods. This method can not only improve the detection accuracy but also improve the detection accuracy. Because the size of clipping and stitching is smaller than the step size, the final stitching result is smoother than other methods.

After comparing the AlexNet method with OTSU, FCM, and SVM methods, the accuracy and kappa coefficient is used to evaluate the result. The specific numerical results are shown in Table 5. The most common evaluation index is accuracy, which can directly reflect the correct proportion of the results, and the calculation is straightforward. When detecting oil spills from remote sensing images, due to the uneven distribution of oil spills and nonoil spills, the oil spills only account for a small part of the whole remote sensing image. In this case, it will lead to high accuracy, which cannot well reflect the results of oil spill detection. Therefore, the kappa coefficient is added as the evaluation index. Kappa coefficient, as an index of consistency test, can better evaluate the test results.

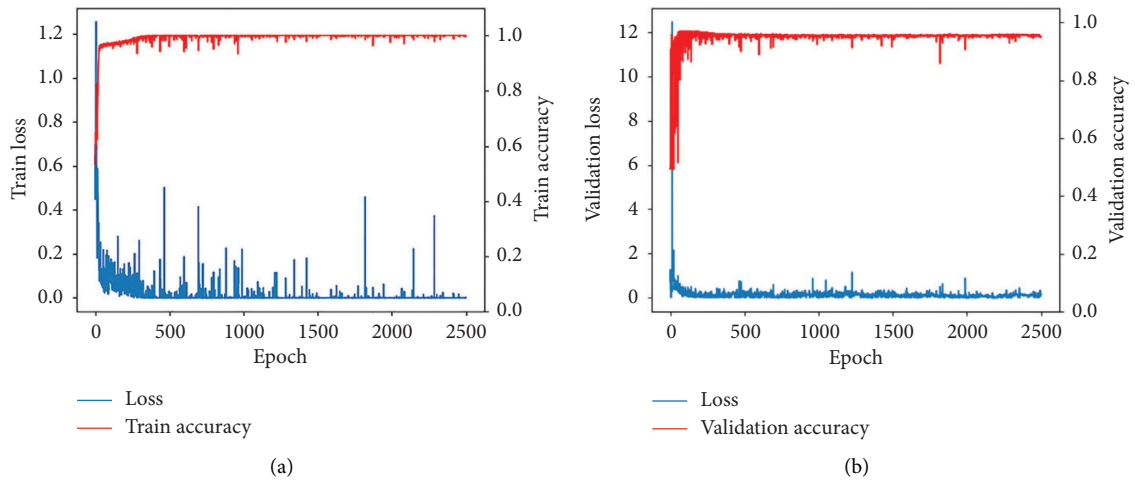


FIGURE 6: Loss and accuracy curves. (a) Train and (b) validation.

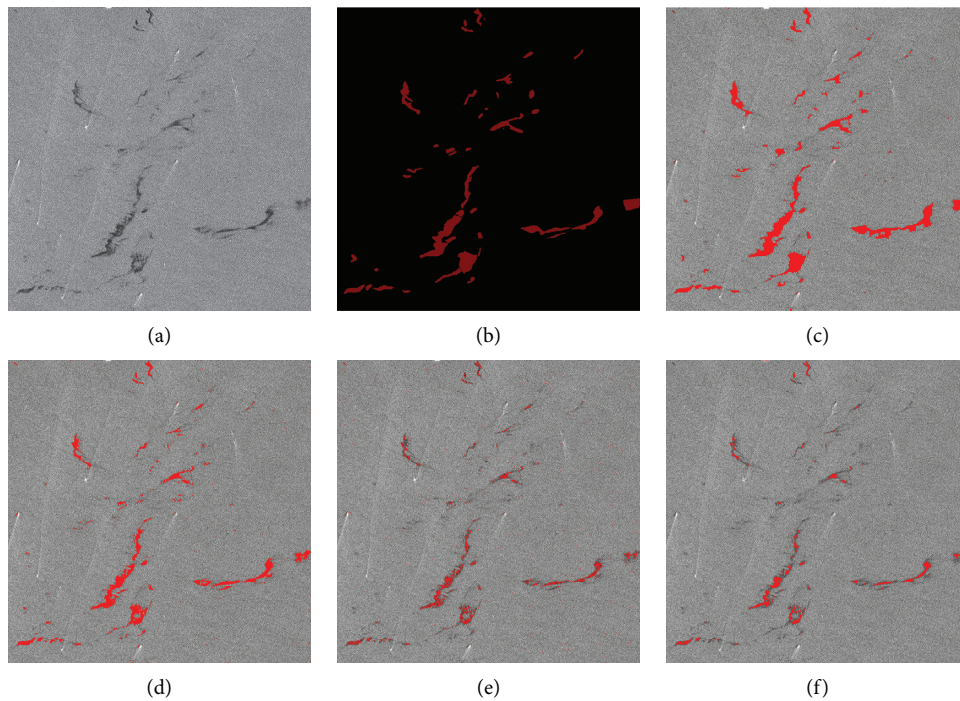


FIGURE 7: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results. (d) OTSU. (e) FCM. (f) SVM.

The accuracy and kappa coefficient are used to evaluate the overall results, while in oil spill detection, the detection of oil spill area is more important than that of the nonoil spill area, so the recall rate is used to evaluate the detection results further. Recall rate reflects the proportion of predicted positive samples in actual positive samples. It only cares about positive samples, so it can better evaluate the results of oil spill detection. The specific results are shown in Table 6. It is not difficult to find that compared with other methods, the detection accuracy of AlexNet for oil spill area is much higher than the other three methods.

In addition, the time required by several detection methods is calculated. The specific time required is shown in

Table 7. It is not difficult to find that the AlexNet algorithm takes the longest time. But for the oil spill detection, the detection accuracy is more important. It takes a long time to obtain higher accuracy, which is acceptable.

In order to test the generalization ability of the model, several other oil spill image input models are selected to test. These test images have different sizes and characteristics. In SAR images, the brightness of remote sensing images is different because of the different backscattering coefficients. The larger the coefficient, the brighter the image. The lower the coefficient, the darker the image. Therefore, before the image is predicted, the image's brightness to be predicted needs to be adjusted according to the image brightness

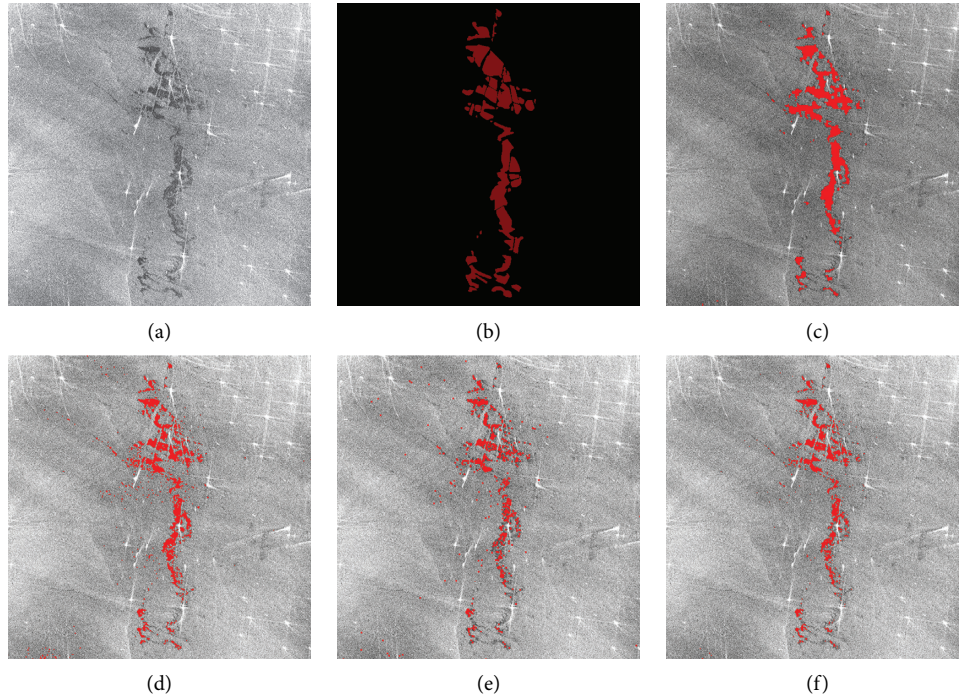


FIGURE 8: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results. (d) OTSU. (e) FCM. (f) SVM.

TABLE 5: Test results of four methods.

Images	Overall accuracy (%)				Kappa			
	AlexNet	OTSU	FCM	SVM	AlexNet	OTSU	FCM	SVM
Sample 1	99.74	98.62	97.66	97.78	0.79	0.73	0.41	0.41
Sample 2	97.54	97.09	96.47	96.88	0.67	0.59	0.45	0.51

TABLE 6: Recall results of four methods.

Images	Recall (%)			
	AlexNet	OTSU	FCM	SVM
Sample 1	97.21	63.14	60.9	26.46
Sample 2	78.50	47.27	32.7	36.33

TABLE 7: Time of four methods.

Images	Time (s)			
	AlexNet	OTSU	FCM	SVM
Sample 1	74.32	1.63	18.36	46.16
Sample 2	70.24	1.72	21.95	43.57

standard in the training set. This can reduce the detection error caused by different image brightness. The experimental images are shown in Figure 9–13. They are called Samples 3, 4, 5, 6, and 7, respectively.

After the picture is predicted, the results are tabulated in Table 8. It is seen that the AlexNet model has achieved good results in the picture. It has good robustness and generalization ability in detecting oil spill area.

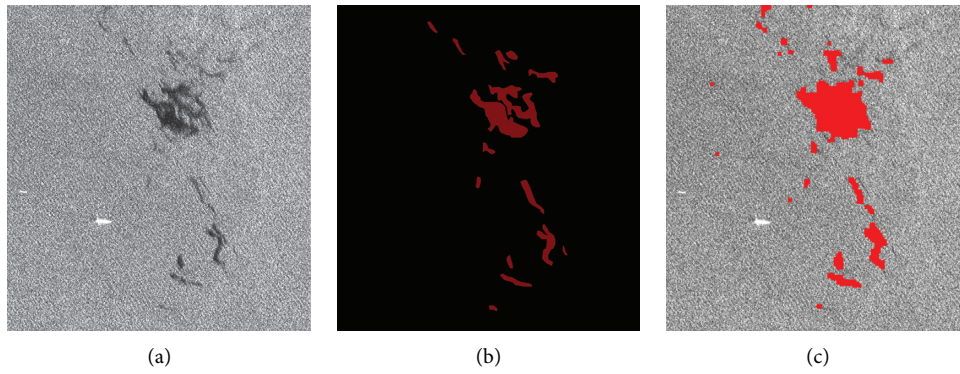


FIGURE 9: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results.

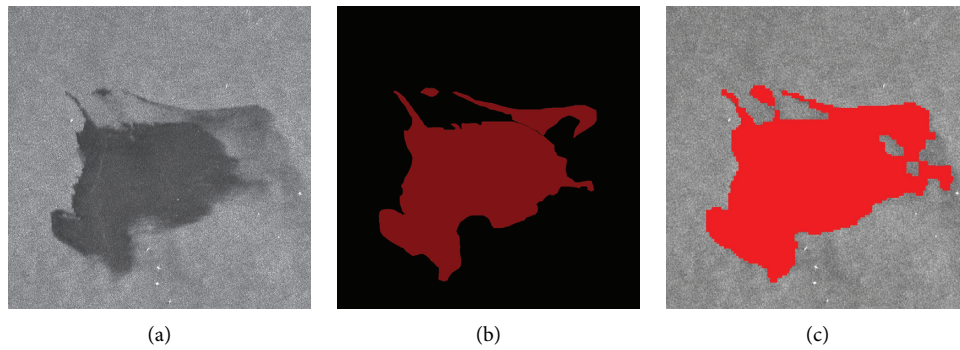


FIGURE 10: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results.

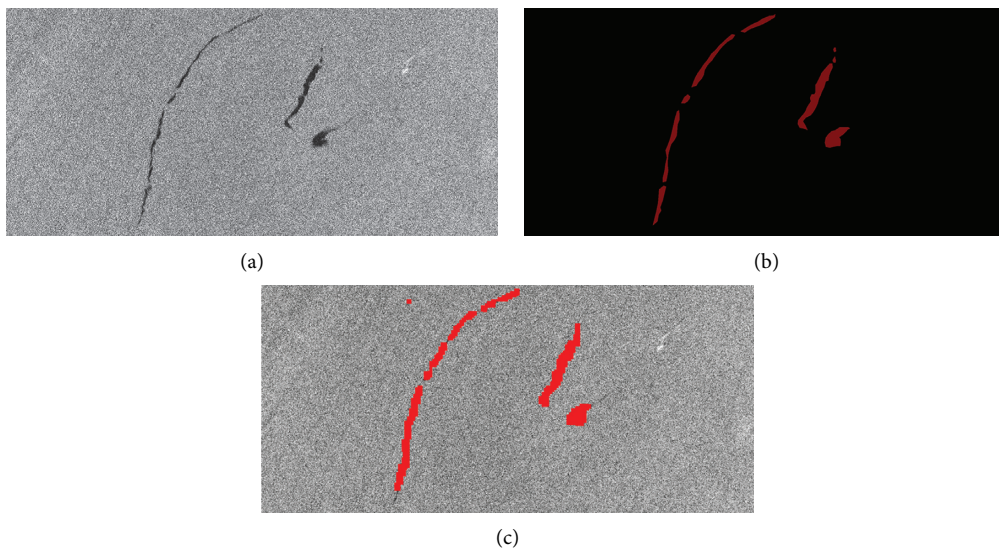


FIGURE 11: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results.

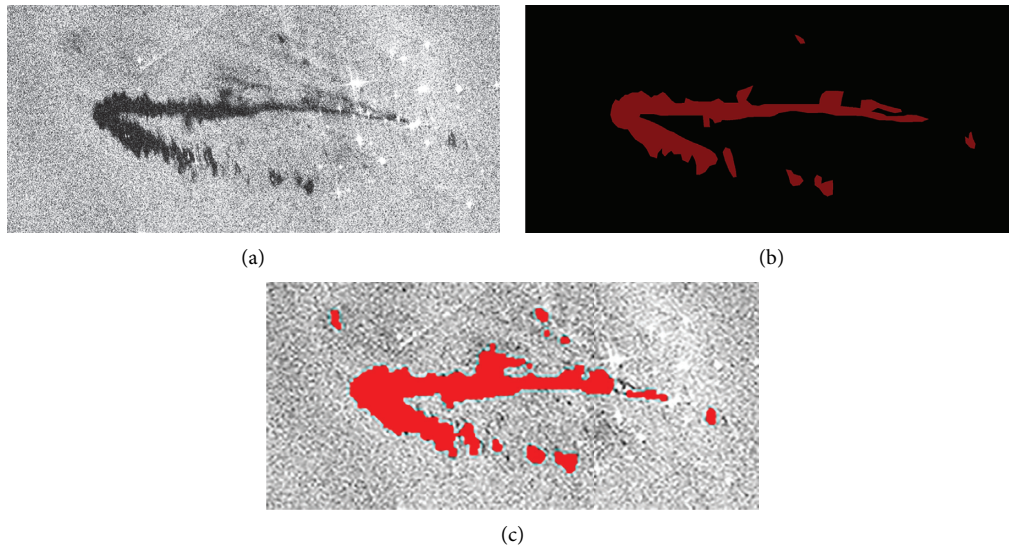


FIGURE 12: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results.

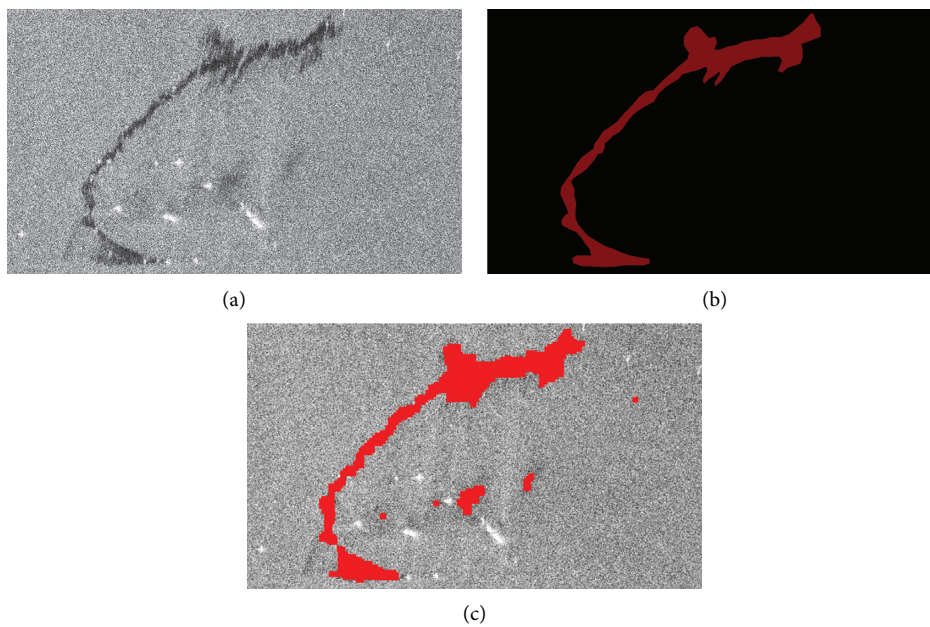


FIGURE 13: Comparison of experimental results. (a) Original SAR image. (b) The true value figures. (c) AlexNet prediction results.

TABLE 8: Experimental evaluation results.

Images	Recall (%)	Overall accuracy (%)	Kappa
Sample 3	96.32	97.73	0.70
Sample 4	99.92	93.86	0.83
Sample 5	99.76	98.91	0.73
Sample 6	96.14	96.72	0.78
Sample 7	97.24	97.93	0.83



## 5. Conclusion

By the advantage of its characterization learning ability, the CNN model achieves a high identification accuracy under the condition of small training sample size and only data enhancement and expansion capacity and extracts the oil spill areas that have not been calibrated by manual calibration. At the same time, by the advantage of its translation invariance and scaling invariance, the CNN model has good generalization ability and robustness and can still extract the oil spill area with high precision even when there is a certain difference between the two images. According to the experiment, it is feasible to use the convolutional neural network to extract the marine oil spill. YOLO model, as another small target detection scheme, can directly process the whole remote sensing imagery effectively. Thus, YOLO model can be used to detect marine oil spill in the future.

## Data Availability

All datasets in the experiment are based on GF-3 and Radarsat-2 SAR images, which are not freely available. These datasets can be checked and ordered from China Centre For Resources Satellite Data and Application and Canadian Space Agency. All datasets in the experiment are based on GF-3 and Radarsat-2 SAR images, which are not freely available. These datasets can be checked and ordered from China Centre For Resources Satellite Data and Application and Canadian Space Agency.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 42076184, 41876109, 41806207, and 41706195, in part by the National Key Research and Development Program of China under grant nos. 2017YFC1404902 and 2016YFC1401007, in part by the National High Resolution Special Research under grant no. 41-Y30F07-9001-20/22, and Dalian University of Technology Innovation Training Program for College Students.

## References

- [1] F. Cui and S. S. Zhang, "Ocean exploitation and environmental risk: an analysis of the Gulf of Mexico oil spill in USA," *Journal of Ocean University of China (Social Sciences)*, vol. 5, pp. 6–10, 2011.
- [2] C. Tao, "A study on types, characteristics and social effects of ocean pollution incidents," *Pacific Journal*, vol. 23, no. 3, pp. 87–96, 2015.
- [3] Y. N. Zhang, Q. Ding, and Q. J. Li, "A study on monitoring of oil spill at sea by satellite remote sensing," *Journal of Dalian Maritime University*, vol. 25, no. 3, pp. 1–5, 1999.
- [4] L. C. Sun, Q. Zhou, and J. Wang, "The present situation and forecast of marine oil spill detection technology by using remote sensing," *Ocean Development and Management*, vol. 36, no. 3, pp. 49–53, 2019.
- [5] F. Y. Zhou, L. P. Jin, and J. Dong, "Review of convolutional neural network," *Chinese Journal of Computers*, vol. 40, no. 6, pp. 1229–1251, 2017.
- [6] S. L. Song, Y. Y. Zhu, and M. H. Zhang, "Advances in marine oil spill monitoring using remote sensing," *Shanxi Architecture*, vol. 43, no. 20, pp. 205–206, 2017.
- [7] N. Li and X. L. Wang, "Research on classification method of remote sensing image based on ENVI," *Technology Innovation and Productivity*, vol. 5, pp. 63–65, 2020.
- [8] J. Xu, B. Li, C. Cui, P. Liu, and X. Y. Zhu, "Research on marine radar oil spill monitoring technology," *Marine Environmental Science*, vol. 37, no. 1, pp. 16–20, 2018.
- [9] J. Jin, Y. N. Wu, and Z. L. Kang, "Feature extraction of oil spill dark spot based on multi-feature in SAR image," *Geomatics & Spatial Information Technology*, vol. 41, no. 2, pp. 53–56, 2018.
- [10] Y. R. Zou, C. Liang, and T. Zeng, "Oil spill identification using SVM based on polarization parameters," *Journal of Marine Sciences*, vol. 31, no. 3, pp. 71–75, 2013.
- [11] J. C. Hu and D. F. Wang, "Monitoring method of ocean oil spilling based on remote sensing," *Environmental Protection Science*, vol. 40, no. 1, pp. 68–73, 2014.
- [12] Y. Yan, X. L. Dong, and Y. Li, "The comparative study of remote sensing image supervised classification methods based on ENVI," *Beijing Surveying and Mapping*, vol. 3, pp. 14–16, 2011.
- [13] M. G. Gong, L. Z. Su, and H. Li, "A survey on change detection in synthetic aperture radar imagery," *Journal of Computer Research and Development*, vol. 53, no. 1, pp. 123–137, 2016.
- [14] C. Liu, C. W. Qu, Q. Zhou, and Z. Li, "SAR images target classification algorithm optimization based on convolutional neural network," *Radar Science and Technology*, vol. 15, no. 4, pp. 362–367, 2017.
- [15] C. Kwan, B. Chou, J. Yang, and T. Tran, "Deep learning based target tracking and classification for infrared videos using compressive measurements," *Journal of Signal and Information Processing*, vol. 10, no. 4, pp. 167–199, 2019.
- [16] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
- [17] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [18] H. L. Luo and Y. Zhang, "A survey of image semantic segmentation based on deep network," *Acta Electronica Sinica*, vol. 47, no. 10, pp. 2211–2220, 2019.
- [19] C. Shu and S. S. Sha, "Characteristics analysis of the oil spill and looks-alike for sea surface oil spill automatic monitoring," *Ship Ocean Engineering*, vol. 49, no. 2, pp. 64–67, 2020.
- [20] Z. L. Lu, X. Jia, W. G. Zhu, and C. Z. Zeng, "Study on SAR image despeckling algorithm," *Journal of Sichuan Ordnance*, vol. 38, no. 6, pp. 104–108, 2018.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 60, no. 6, pp. 84–90, 2017.
- [22] B. Z. Li, K. Liu, J. J. Gu, and W. Z. Jiang, "Review of the researches on convolutional neural networks," *Computer Era*, vol. 4, pp. 12–17, 2021.
- [23] M. Y. Ji, X. M. Xi, and Z. L. Yu, "A review of semantic segmentation based on deep learning," *Information Technology & Informatization*, vol. 10, pp. 137–140, 2017.

- [24] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Artificial Intelligence and Statistics*, vol. 15, no. 8, pp. 315–323, 2011.
- [25] J. Y. Qu, X. Sun, and X. Gao, "Remote sensing image target recognition based on CNN," *Foreign Electronic Measurement Technology*, vol. 35, no. 8, pp. 45–50, 2016.
- [26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [27] B. Ayhan and C. Kwan, "Practical considerations in unsupervised change detection using SAR images," in *Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0334–0339, New York, NY, USA, 2019.
- [28] S. Temitope Yekeen, A. L. Balogun, and K. B. Wan Yusof, "A novel deep learning instance segmentation model for automated marine oil spill detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 190–200, 2020.
- [29] W. C. Xiong, C. Q. Wu, B. Wei, W. M. Shen, and Z. P. Sun, "Oil spill detection with SAR in South Korea's oil leak," *Remote Sensing Technology and Application*, no. 4, pp. 410–413+358, 2008.
- [30] L. J. Shi, C. F. Zhao, and P. Liu, "Oil spill identification in marine SAR images based on texture feature and artificial neural network," *Periodical of Ocean University of China*, vol. 39, no. 6, pp. 1269–1274, 2009.
- [31] F. Yang, J. Yang, J. J. Yin, and J. S. Song, "Spill detection based on polarimetric SAR decomposition models," *Journal of Tsinghua University (Science and Technology)*, vol. 55, no. 8, pp. 854–859, 2015.
- [32] J. Sun, Y. Xu, F. X. Chen, and Z. R. Peng, "Research on offshore petroleum oil spilling detection using SAR echo signal," *Acta Oceanologica Sinica*, vol. 36, no. 9, pp. 103–111, 2014.
- [33] X. Zhang and B. W. Liu, "Research on SAR target recognition based on convolutional neural networks," *Electronic Measurement Technology*, vol. 44, no. 14, pp. 92–96, 2018.
- [34] J. Huang, Z. G. Jiang, H. P. Zhang, and Y. Yao, "Ship object detection in remote sensing images using convolutional neural networks," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 43, no. 9, pp. 1841–1848, 2017.
- [35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, pp. 326–366, MIT press, Cambridge, UK, 2016.
- [36] J. B. Estrach, A. Szlam, and Y. Lecun, "Signal recovery from pooling representations," in *Proceedings of the International Conference on Machine Learning*, pp. 307–315, Beijing, China, 2014.

## Research Article

# Quadruplet-Based Deep Cross-Modal Hashing

Huan Liu,<sup>1</sup> Jiang Xiong ,<sup>1</sup> Nian Zhang,<sup>2</sup> Fuming Liu,<sup>1</sup> and Xitao Zou <sup>1</sup>

<sup>1</sup>Key Laboratory of Intelligent Information Processing and Control, Chongqing Municipal Institutions of Higher Education, Chongqing Three Gorges University, Chongqing 40044, China

<sup>2</sup>Department of Electrical and Computer Engineering, University of the District of Columbia, Washington, D. C., SC 20008, USA

Correspondence should be addressed to Jiang Xiong; [xjcq123@126.com](mailto:xjcq123@126.com)

Received 18 March 2021; Revised 24 May 2021; Accepted 14 June 2021; Published 2 July 2021

Academic Editor: Raşit Köker

Copyright © 2021 Huan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, benefitting from the storage and retrieval efficiency of hashing and the powerful discriminative feature extraction capability of deep neural networks, deep cross-modal hashing retrieval has drawn more and more attention. To preserve the semantic similarities of cross-modal instances during the hash mapping procedure, most existing deep cross-modal hashing methods usually learn deep hashing networks with a pairwise loss or a triplet loss. However, these methods may not fully explore the similarity relation across modalities. To solve this problem, in this paper, we introduce a quadruplet loss into deep cross-modal hashing and propose a quadruplet-based deep cross-modal hashing (termed QDCMH) method. Extensive experiments on two benchmark cross-modal retrieval datasets show that our proposed method achieves state-of-the-art performance and demonstrate the efficiency of the quadruplet loss in cross-modal hashing.

## 1. Introduction

With the advent of the era of big data, there are surging massive multimedia data on the Internet, such as images, videos, and texts. These data usually exist in diversified modalities, for example, there may exist a textual data and an audio data describing a video data or an image data. As data from different modalities may have compact semantic relevance, cross-modal retrieval [1, 2] is proposed to retrieve semantic similar data from one modality while the querying data is from a distinct modality. Benefitting from the high efficiency and low cost, hashing-based cross-modal retrieval (cross-modal hashing) [3–6] has drew extensive attention. The goal of cross-modal hashing is to map the modal heterogeneous data into a common binary space and ensure that semantic similar/dissimilar cross-modal data have similar/dissimilar hash codes. Cross-modal hashing methods can usually achieve superior performance; nonetheless, most of existing cross-modal hashing methods (such as cross-modal similarity sensitive hashing (CMSSH) [7], semantic correlation maximization (SCM) [8], semantics-preserving hashing (SePH) [9], and generalized semantic preserving hashing (GSPH) [10]) are based on handcrafted feature learning, which cannot effectively capture the heterogeneous

relevance between different modalities and thus may result in inferior performance.

In the last decade, deep convolutional neural networks [11, 12] have been successfully utilized in many computer vision tasks, and therefore, some researchers also deploy it in cross-modal hashing, such as deep cross-modal hashing (DCMH) [13], pairwise relationship guided deep hashing (PRDH) [14], self-supervised adversarial hashing (SSAH) [15], and triplet-based deep hashing (TDH) [16]. Cross-modal hashing methods with deep neural networks efficiently integrate the hash representation learning and the hash function learning into an end-to-end framework, which can capture heterogeneous cross-modal relevance more effectively and thus acquire better cross-modal retrieval performance.

To date, most deep cross-modal hashing methods utilize the pairwise loss (such as [13–15]) or the triplet loss (such as [16]) to preserve semantic relevance during the hash representation learning procedure. Nevertheless, the pairwise loss- and triplet loss-based hash methods suffer from a weak generalization capacity from the training set to the testing set [17, 18], as shown in Figure 1(a). On the contrary, quadruplet loss is proposed and has been utilized in image hashing retrieval [17] and

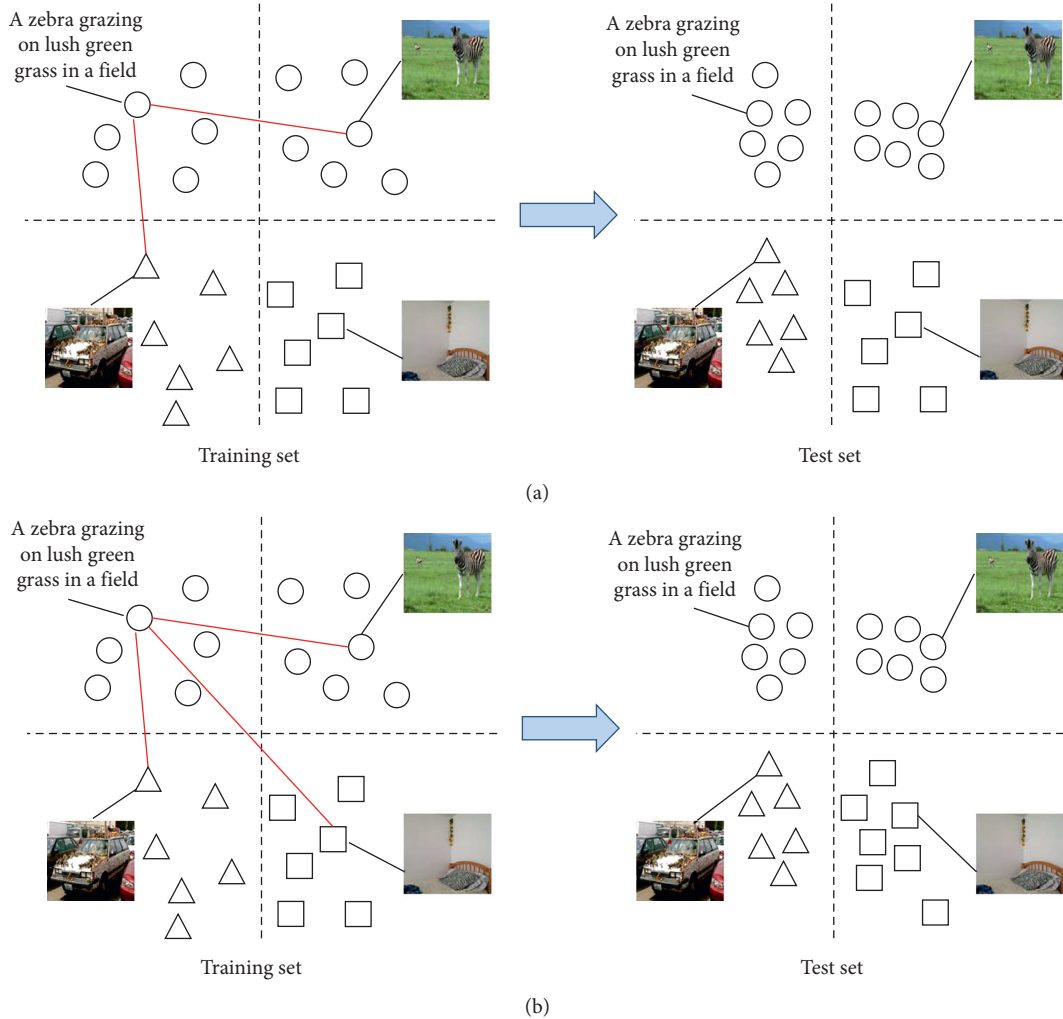


FIGURE 1: (a) Triplet loss-based cross-modal hashing methods suffer from a weak generalization capacity from the training set to the testing set because the test instances belong to the category  $\square$  and cannot be mapped into compact binary codes (see the lower-right corner). (b) Triplet loss-based cross-modal hashing methods can project the test instances, which belong to the category  $\square$ , into compact binary space (see the lower right corner).

person reidentification [18], and in these works, it has been proved that the quadruplet loss-based model can enhance the generalization capability. Therefore, cross-modal hashing combines quadruplet loss as a natural solution to enhance the performance of cross-modal hashing, as shown in Figure 1(b).

To this end, in this paper, we introduce quadruplet loss into cross-modal hashing and propose a quadruplet-based deep cross-modal hashing method (QDCMH). Specifically, QDCMH firstly defines a quadruplet-based cross-modal semantic preserving module. Afterwards, QDCMH integrates this module, hash representation learning, and hash code generation into an end-to-end framework. Finally, experiments on two benchmark cross-modal retrieval datasets are conducted to validate the performance of the proposed method. The main contributions of our proposed QDCMH include the following:

- (i) We introduce quadruplet loss into cross-modal retrieval and propose a novel deep cross-modal hashing method. To the best of our knowledge, this is

the first work to introduce quadruplet loss into cross-modal hashing retrieval.

- (ii) We conduct extensive experiments on benchmark cross-modal retrieval datasets to investigate the performance of our proposed QDCMH.

The remainder of this paper is organized as follows. Section 2 elaborates our proposed quadruplet-based deep cross-modal hashing method. Section 3 presents the learning algorithm of QDCMH. Section 4 is the experimental results and the corresponding analysis. Section 5 concludes our work.

## 2. Proposed Method

In this section, we elaborate our proposed quadruplet-based deep cross-modal hashing (QDCMH) method with the following sections: notations, quadruplet-based cross-modal semantic preserving module, feature learning networks, and hash function learning. Figure 2 presents the flowchart of

our proposed QDCMH, which cooperates quadruplet-based cross-modal semantic preserving module, hash representation learning, and hash codes generation into an end-to-end framework. In our proposed QDCMH method, we assume that each instance has two modalities, i.e., an image modality and a text modality, but they can be easily applied to multimodalities.

*2.1. Notations.* Assume that the training data comprises  $n$  image-text pairs, i.e., the original image features  $V \in R^{n \times d_v}$  and the original text features  $T \in R^{n \times d_t}$ . Besides, there is a label vector associated with each image-text pair and label vectors for all training instances constitute a label matrix  $L \in R^{n \times d_l}$ .  $d_v$  and  $d_t$  are the corresponding original dimensions of image features and text features, respectively, and  $d_l$  is the total number of class categories. If image-text pair  $\{V_i, T_i\}$  attaches to the  $j$ th category, then  $L_{ij} = 1$ , otherwise  $L_{ij} = 0$ . The quadruplet  $(V_q, T_p, T_{n1}, T_{n2})$  denotes that  $V_q$  is a query instance from the image modality, and  $T_p, T_{n1}, T_{n2}$  are three retrieval instances from the text modality, where  $V_q$  and  $T_p$  have at least one common categories, while  $V_q$  and  $T_{n1}$ ,  $V_q$  and  $T_{n2}$ , and  $T_{n1}$  and  $T_{n2}$  are three pairwise instances and the two instances in each pairwise have no common label.

With the known quadruplet  $(V_q, T_p, T_{n1}, T_{n2})$ , the target of our proposed QDCMH is to learn the corresponding hash codes  $(B_{V_q}, B_{T_p}, B_{T_{n1}}, B_{T_{n2}})$ , where  $B_{V_q}, B_{T_p}, B_{T_{n1}}, B_{T_{n2}}$  are the hash codes of instances  $V_q, T_p, T_{n1}, T_{n2}$ , respectively. To learn the above hash codes, we first learn the hash representations  $(F_{V_q}, G_{T_p}, G_{T_{n1}}, G_{T_{n2}})$  from the quadruplet  $(V_q, T_p, T_{n1}, T_{n2})$  with deep neural networks, where  $F_{V_q} = f(V_q, \theta_V)$  and  $G_{T_p} = g(T_p, \theta_T)$  are the hash representations of instance  $V_q$  and  $T_p$ , respectively.  $f(\cdot, \theta_V)$  and  $g(\cdot, \theta_T)$  are the hash representation learning functions for the image modality and the text modality, respectively.  $\theta_V$  and  $\theta_T$  are the parameters of deep neural networks to extract features for the image modality and for the text modality, respectively. Secondly, we can utilize the following sign function to

approximately map the hash representations into the corresponding hash codes, i.e.,  $B_{V_q} = \text{sign}(F_{V_q})$  and  $B_{T_p} = \text{sign}(G_{T_p})$ . In the same way, we can learn the hash codes of quadruplet  $(T_q, V_p, V_{n1}, V_{n2})$ . For convenience, we denote the hash codes of all training image-text pairs, the hash representations of all training image instances, and the hash representations of all training text instances as  $B \in \{-1, 1\}^{n \times k}$ ,  $F \in R^{n \times k}$ , and  $G \in R^{n \times k}$ , respectively, where  $k$  is the length of hash codes:

$$y = \begin{cases} 1, & \text{if } x \geq 0, x \in R, \\ -1, & \text{if } x < 0, x \in R. \end{cases} \quad (1)$$

*2.2. Quadruplet-Based Cross-Modal Semantic Preserving Module.* In cross-modal hashing retrieval, given an image instance  $V_i$  and a text instance  $T_j$ , it is intractable to preserve the semantic relativity during the hash code learning procedure as the huge semantic gap across modalities. To solve this, DCMH [13] defines pairwise loss to map similar/dissimilar image-text pairs into similar/dissimilar hash codes. TDH [16] utilizes triplet loss to learn similar hash codes for similar cross-modal instances and generate distinct hash codes for semantic irrelevant cross-modal instances. Both pairwise loss and triplet loss can preserve the relevance in the original instance space; however, pairwise loss- and triplet loss-based hashing methods often suffer from a weaker generalization capability from the training set to the testing set [17, 18]. To solve this problem, in this section, a quadruplet-based cross-modal semantic preserving module is proposed to boost the generalization capability and better preserve the semantic relevance for cross-modal hashing.

For a quadruplet  $(V_q, T_p, T_{n1}, T_{n2})$ , we should keep the semantic relevance unchanged during the hash representation learning, i.e.,  $F_{V_q}$  should be similar to  $G_{T_p}$ ,  $F_{V_q}$  should be distinct to  $G_{T_{n1}}$  and  $G_{T_{n2}}$ , and  $G_{T_{n1}}$  should be dissimilar with  $G_{T_{n2}}$ . Thus, we can define the following quadruplet loss for cross-modal hashing:

$$J_{\text{quadruplet}}^{I \rightarrow T}(F_{V_q}, G_{T_p}, G_{T_{n1}}, G_{T_{n2}}) = \sum_{V_q, T_p, T_{n1}} \max\left(0, \|F_{V_q} - G_{T_p}\|_2^2 - \|F_{V_q} - G_{T_{n1}}\|_2^2 + \alpha_1\right) + \sum_{V_q, T_p, T_{n1}, T_{n2}} \max\left(0, \|F_{V_q} - G_{T_p}\|_2^2 - \|G_{T_{n1}} - G_{T_{n2}}\|_2^2 + \alpha_2\right), \quad (2)$$

where  $V_q$  is a query instance from the image modality,  $T_p$ ,  $T_{n1}$ , and  $T_{n2}$  are three retrieval instances from the text modality, and  $V_q$  and  $T_p$  are semantic similar. While  $V_q$  and  $T_{n1}$ ,  $V_q$  and  $T_{n2}$ , and  $T_{n1}$  and  $T_{n2}$  are three pairwise instances, and the two instances in each pairwise have distinct semantics. Equation (2) denotes that the distance of hash representations of similar cross-modal pairwise instances

should be smaller than that of dissimilar pairwise instances (both from intermodalities and from intramodalities) with a positive margin ( $\alpha_1$  or  $\alpha_2$ ). This can ensure that similar cross-modal instances have similar hash representations while dissimilar instances have distinct hash representations. By this quadruplet loss, the cross-modal semantic relevance can be preserved during the hash representation learning stage.

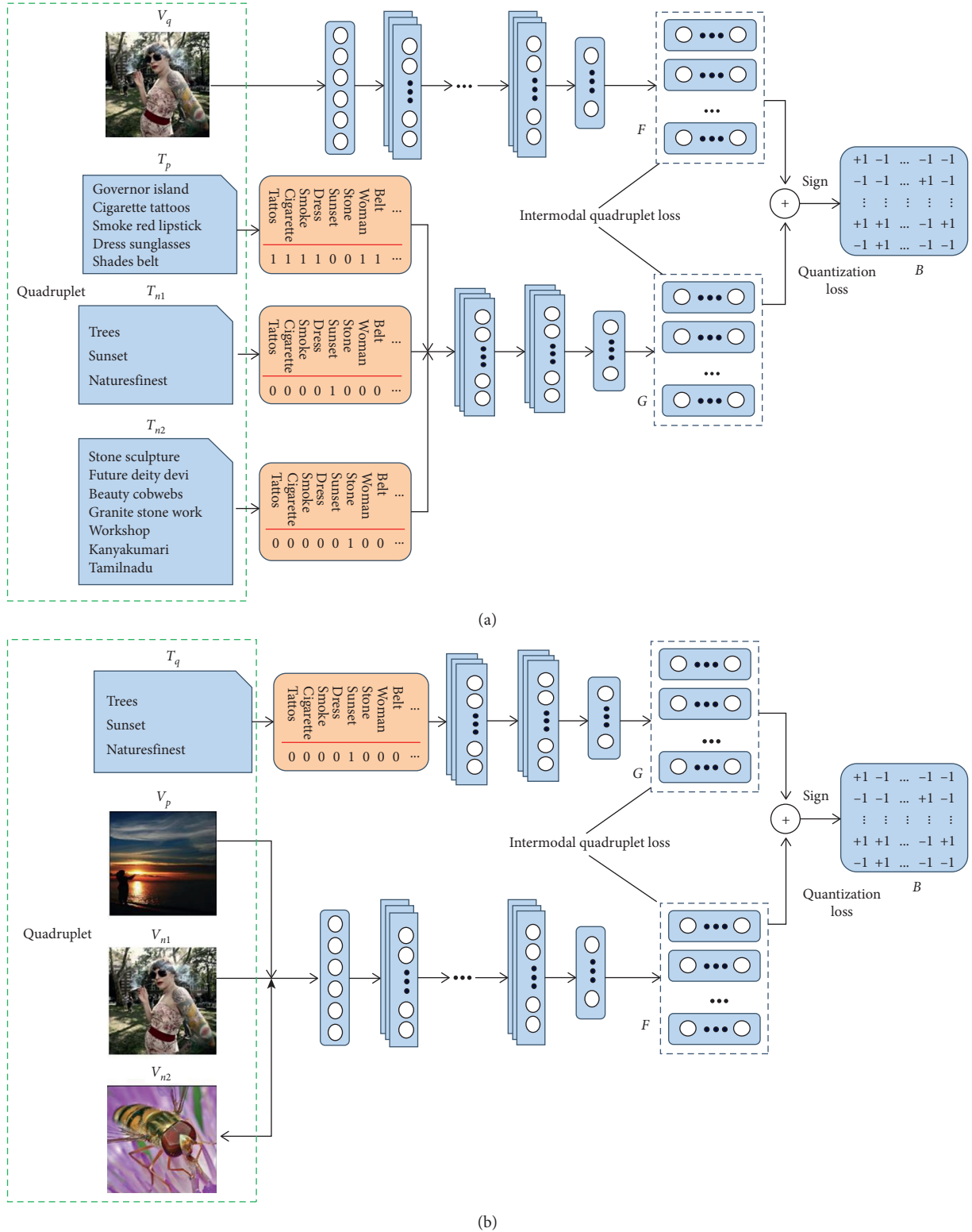


FIGURE 2: Flowchart of the proposed quadruplet-based deep cross-modal hashing (QDCMH) method. QDCMH encompasses three steps: (1) a quadruplet-based cross-modal semantic preserving module, (2) a classical convolutional neural network is used to learn image-modality features and the TxtNet in SSAH [15] is adopted to learn the text-modality features, and (3) an intermodal quadruplet loss is utilized to efficiently capture the relevant semantic information during the feature learning process and a quantization loss is used to decrease information loss during the hash codes generation procedure. (a) Quadruplet ( $V_q, T_p, T_{n1}, T_{n2}$ ), which utilizes an image instance  $V_q$  to retrieve three text instances:  $T_p, T_{n1}$ , and  $T_{n2}$ .  $V_q$  and  $T_p$  have at least one common labels, while  $V_q$  and  $T_{n1}$ ,  $V_q$  and  $T_{n2}$ , and  $T_{n1}$  and  $T_{n2}$  are three pairwise instances and the two instances in each pairwise have no common label. (b) Quadruplet ( $V_q, T_p, T_{n1}, T_{n2}$ ), which utilizes a text instance  $T_q$  to retrieve three image instances:  $V_p, V_{n1}$ , and  $V_{n2}$ .  $T_q$  and  $V_p$  have at least one common labels, while  $T_q$  and  $V_{n1}$ ,  $T_q$  and  $V_{n2}$ , and  $V_{n1}$  and  $V_{n2}$  are three pairwise instances and the two instances in each pairwise have no common label.

Similarly, given a quadruplet  $(T_q, V_p, V_{n1}, V_{n2})$ , we can have the following cross-modal quadruplet loss:

$$J_{\text{quadruplet}}^{T \rightarrow I}(G_{T_q}, F_{V_p}, F_{V_{n1}}, F_{V_{n2}}) = \sum_{T_q, V_p, V_{n1}} \max\left(0, \|G_{T_q} - F_{V_p}\|_2^2 - \|G_{T_q} - F_{V_{n1}}\|_2^2 + \alpha_3\right) + \sum_{T_q, V_p, V_{n1}, V_{n2}} \max\left(0, \|G_{T_q} - F_{V_p}\|_2^2 - \|F_{V_{n1}} - F_{V_{n2}}\|_2^2 + \alpha_4\right), \quad (3)$$

where  $T_q$  is a query instance from the text modality,  $V_p, V_{n1}$ , and  $V_{n2}$  are three retrieval instances from the image modality,  $G_{T_q}, F_{V_p}, F_{V_{n1}}$ , and  $F_{V_{n2}}$  are hash representations for instances  $T_q, V_p, V_{n1}$ , and  $V_{n2}$ , respectively, and  $\alpha_3$  and  $\alpha_4$  are two positive margins. Equation (3) is distinct to equation (2) as the modality of query instance and the modality of retrieval instances are inverse.

**2.3. Hash Representation Learning and Hash Code Learning.** For each quadruplet from training set, it is easy to learn their hash representations and fully protect the semantic similarity with the above quadruplet-based cross-modal semantic relevance preserving module, so we have the following hash representation learning loss:

$$J_{\text{representation}} = \frac{1}{n_{I \rightarrow T}} J_{\text{quadruplet}}^{I \rightarrow T}(F_{V_q}, G_{T_p}, G_{T_{n1}}, G_{T_{n2}}) + \frac{\beta}{n_{T \rightarrow I}} J_{\text{quadruplet}}^{T \rightarrow I}(G_{T_q}, F_{V_p}, F_{V_{n1}}, F_{V_{n2}}), \quad (4)$$

where  $n_{I \rightarrow T}$  is the number of quadruplets for utilizing image to retrieve text,  $n_{T \rightarrow I}$  is the number of quadruplets for utilizing text to retrieve images, and  $\beta$  is a hyperparameter to balance the two parts.

Additionally, to learn high-quality hash codes, we generate hash codes from the learned hash representations with the sign function in equation (1), and the final hash codes matrix for all training image-text pairs are generated as follows:

$$B = \text{sign}\left(\frac{F + G}{2}\right). \quad (5)$$

As  $F$  and  $G$  are real-valued features, to decrease the information loss from  $F$  and  $G$  to  $B$  in equation (5), it is necessary to force  $F$  and  $G$  to be as close as possible to  $B$ ; thus, we introduce the following quantization loss:

$$J_{\text{quantization}} = \frac{\|B - F\|_2^2 + \|B - G\|_2^2}{2nk}. \quad (6)$$

Integrating the hash representation loss and the quantization loss together, the whole loss function is as follows:

$$J = J_{\text{representation}} + \gamma J_{\text{quantization}}, \quad (7)$$

where  $\gamma$  is a hyperparameter to balance the hash representation loss and the quantization loss.

**2.4. Feature Extraction Networks.** In QDCMH, feature extraction includes two deep neural networks: a classical convolutional neural network is used to extract the features of images and a multiscale fusion model is utilized to learn features from texts. Specifically, for image modality, we deploy AlexNet [11] pretrained on the ImageNet [19] dataset. We then fine-tune the last layer using a new fully connected hash layer which consists of  $k$  hidden nodes. Therefore, the learned deep features have been embedded into a  $k$ -dimensional Hamming space. For text modality, the TxtNet in SSAH [15] is used, which comprises a three-layer feedforward neural network and a multiscale (MS) fusion model (Input  $\rightarrow$  MS  $\rightarrow$  4096  $\rightarrow$  512  $\rightarrow$   $k$ ).

### 3. Learning Algorithm of QDCMH

For QDCMH, we utilize alternating strategy to learn parameters  $\theta_V$  of deep neural networks for image modality and parameters  $\theta_T$  of deep neural networks for text modality and hash codes matrix  $B$  for all training image-text pairs. When we learn one of  $\theta_V, \theta_T$ , and  $B$ , we keep the other two fixed. The specific algorithm for QDCMH is depicted in Algorithm 1.

**3.1. Update  $\theta_V$  with  $\theta_T$  and  $B$  Fixed.** When  $\theta_T$  and  $B$  are maintained fixed, we utilize stochastic gradient descent and backpropagation to optimize the deep neural network parameters  $\theta_V$ .

**3.2. Update  $\theta_T$  with  $\theta_V$  and  $B$  Fixed.** When we fix the values of  $\theta_V$  and  $B$ , we use stochastic gradient descent and backpropagation to learn the deep neural network parameters  $\theta_T$ .

**3.3. Update  $B$  with  $\theta_T$  and  $\theta_V$  Fixed.** When the deep neural networks' parameters  $\theta_T$  and  $\theta_V$  are kept unchanged, the hash codes matrix  $B$  can be optimized with equation (5).

## 4. Experiments

**4.1. Datasets.** To investigate the performance of QDCMH, we conduct experiments on two benchmark cross-modal retrieval datasets: MIRFLICKR-25K [20] and Microsoft COCO2014 [21], and the brief descriptions of the datasets are listed in Table 1.

**4.2. Evaluation Metrics.** In our experiments, we utilize mean average precision (MAP), top  $N$ -precision curves (top  $N$

**Input:**

training data set:  $\{V, T, L\}$ . The maximal number of epoches of the algorithm is `max_epoch`. Mini-batch size  $n_{\text{batch}} = 128$ .

**Output:**

Parameters  $\theta_V, \theta_T$  of the deep neural networks, and corresponding hash codes matrix  $B$ .

- (1) Generating  $n_{I \rightarrow T}$  ( $V_q, T_p, T_{n1}, T_{n2}$ ) quadruplets (named  $\text{Quad}_{I2T}$ ) from training set, generating  $n_{T \rightarrow I}$  ( $T_q, V_p, V_{n1}, V_{n2}$ ) quadruplets (named  $\text{Quad}_{T2I}$ ) from training set.
- (2) Initialize the deep neural network parameters  $\theta_V, \theta_T$ , the whole training image hash representations  $F$ , the whole training text hash representations  $G$ , the hash codes matrix  $B$ , and the epoch numbers  $\text{batchnum}_v = \text{batchnum}_t = \lceil (n_{I \rightarrow T} + n_{T \rightarrow I}) / n_{\text{batch}} \rceil$ .
- (3) **repeat**
- (4)   **for**  $j = 1$  to  $\text{batchnum}_v$  **do**
- (5)     Randomly sample  $n_v$  images from  $\text{Quad}_{I2T} \cup \text{Quad}_{T2I}$  to construct a mini-batch of images.
- (6)     For each instance  $V_i$  in the mini-batch, calculate  $F_{V_i} = f(V_i, \theta_V)$  by forward propagation.
- (7)     Update  $F$ .
- (8)     Calculate the derivative of  $\theta_V$  in equation (7).
- (9)     Update the network parameters  $\theta_I$  by utilizing backpropagation.
- (10)   **end for**
- (11)   **for**  $j = 1$  to  $\text{batchnum}_t$  **do**
- (12)     Randomly sample  $n_t$  texts from  $\text{Quad}_{I2T} \cup \text{Quad}_{T2I}$  to construct a mini-batch of texts.
- (13)     For each instance  $T_i$  in the mini-batch, calculate  $G_{T_i} = g(T_i, \theta_T)$  by forward propagation.
- (14)     Update  $G$ .
- (15)     Calculate the derivative of  $\theta_T$  in equation (7).
- (16)     Update the network parameters  $\theta_T$  by using backpropagation.
- (17)   **end for**
- (18)   Update  $B$  using equation (5).
- (19) **until** the max epoch number `max_epoch`.

ALGORITHM 1: QDCMH: quadruplet-based deep cross-modal hashing.

TABLE 1: Brief description of the experimental datasets.

Dataset	Used	Train	Query	Retrieve	Tag dimension	Labels
MIRFLICKR-25K	20,015	10,000	2,000	18,015	1,386	24
MS-COCO2014	122,218	10,000	5,000	117,218	2,026	80

Curves), and precision-recall curves (PR Curves) as evaluation metrics; for the detailed description of these evaluation metrics, refer to [22, 23].

**4.3. Baselines and Implementation Details.** We compare our proposed QDCMH method with eight state-of-the-art cross-modal hashing methods, including four handcrafted ones, i.e., cross-modal similarity sensitive hashing (CMSSH) method [7], semantics-preserving hashing (SePH) [9] method, semantic correlation maximization (SCM) method [8], and generalized semantic preserving hashing (GSPH) method [10] and four deep feature-based ones, i.e., deep cross-modal hashing (DCMH) method [13], pairwise relationship guided deep hashing (PRDH) method [14], self-supervised adversarial hashing (SSAH) method [15], and triplet-based deep hashing (TDH) method [16]. Most baseline methods are carefully implemented based on the codes provided by the authors. A few baseline methods are implemented by us following the suggestions and descriptions of the original papers.

All the experiments are executed by using the open source deep learning framework pytorch and running on an NVIDIA GTX Titan XP GPU server. In our experiments, we set  $n_{I \rightarrow T} = n_{T \rightarrow I} = 10000$ , `max_epoch` = 500, and  $\lambda = 10^{-5}$

and the learning rate is initialized to  $10^{-1.5}$  and gradually decreased to  $10^{-6}$  in 500 epochs. For those handcrafted feature-based baselines, each image in the two datasets is represented by a bag of words (BoW) histogram or feature vector having 512 dimensions. For the whole experiment, we use  $I \rightarrow T$  to denote using a querying image while returning text and  $T \rightarrow I$  to denote using a querying text while returning an image.

**4.4. Performance Evaluation and Discussion.** Firstly, we investigate the performance of QDCMH with different hyperparameters  $\beta$  and  $\gamma$ . To this goal, we experiment on MIRFLICKR-25K with the hash code length  $k = 64$  and record the corresponding MAPs under different values of  $\beta$  and  $\gamma$ , as shown in Figure 3. We find that high performance can be acquired when  $\beta = 1$  and  $\gamma = 0.2$ .

Secondly, to validate the performance of QDCMH, we perform the experiment to compare QDCMH with baseline methods in terms of MAP on datasets MIRFLICKR-25K and MS-COCO2014. Table 2 presents the MAPs of each method for different hash code lengths, i.e., 16, 32, and 64. DSePH represents the SePH method whose features of the original images are extracted by CNN-F. From Table 2, we can see that the following. (1) The MAPs of our proposed QDCMH



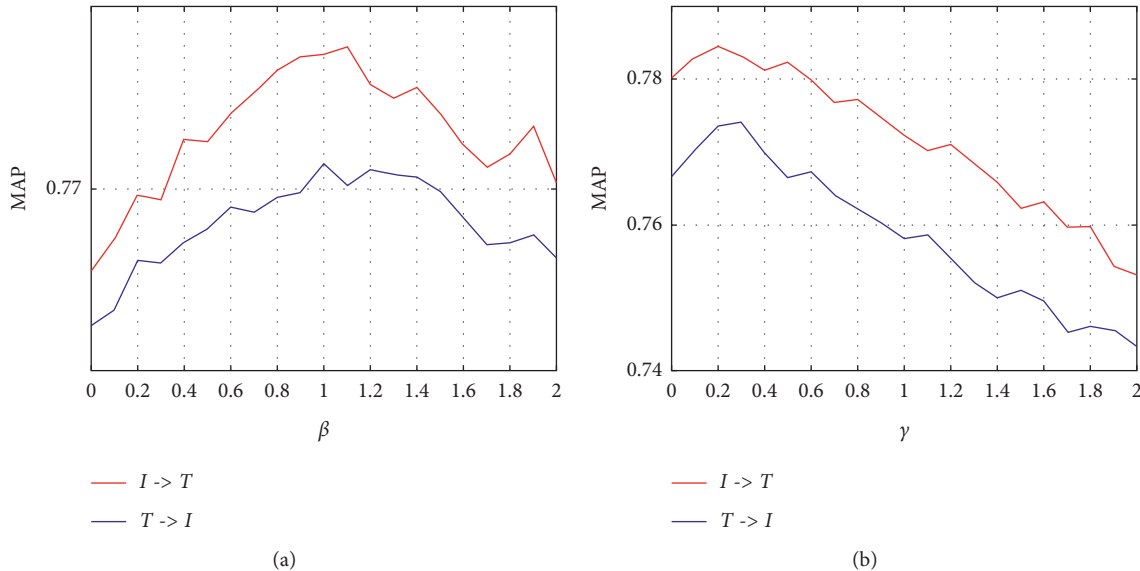


FIGURE 3: A sensitivity analysis of the hyperparameters. (a) Hyperparameter  $\beta$  on MIRFLICKR-25K dataset. (b) Hyperparameter  $\gamma$  on MIRFLICKR-25K dataset.

TABLE 2: Comparison to baselines in terms of MAP on two datasets: MIRFLICKR-25K, and Microsoft COCO2014, respectively. The best accuracy is shown in boldface.

Task	Methods	MIRFlickr-25K			MS-COCO			
		16bits	32bits	64bits	16bits	32bits	64bits	
$I \rightarrow T$	Handcrafted methods	CMSSH [7]	0.5600	0.5709	0.5836	0.5439	0.5450	0.5410
		SePH [9]	0.6740	0.6813	0.6803	0.4295	0.4353	0.4726
		SCM [8]	0.6354	0.6407	0.6556	0.4252	0.4344	0.4574
		GSPH [10]	0.6068	0.6191	0.6230	0.4427	0.4733	0.4840
	Deep methods	DCMH [13]	0.7316	0.7343	0.7446	0.5228	0.5438	0.5419
		PRDH [14]	0.6952	0.7072	0.7108	0.5238	<b>0.5521</b>	<b>0.5572</b>
		SSAH [15]	<b>0.7745</b>	<b>0.7882</b>	<b>0.7990</b>	0.5127	0.5256	0.5067
		TDH [16]	0.7423	0.7478	0.7512	0.5164	0.5222	0.5276
		DSePH [9]	0.7128	0.7285	0.7422	0.4621	0.4958	0.5112
		QDCMH	0.7635	0.7688	0.7713	<b>0.5286</b>	0.5313	0.5371
$T \rightarrow I$	Handcrafted methods	CMSSH [7]	0.5726	0.5776	0.5753	0.3793	0.3876	0.3899
		SePH [9]	0.7139	0.7258	0.7294	0.4348	0.4606	0.5195
		SCM [8]	0.6340	0.6458	0.6541	0.4118	0.4183	0.4345
		GSPH [10]	0.6282	0.6458	0.6503	0.5435	0.6039	0.6461
	Deep methods	DCMH [13]	0.7607	0.7737	0.7805	0.4883	0.4942	0.5145
		PRDH [14]	0.7626	0.7718	0.7755	0.5122	0.5190	0.5404
		SSAH [15]	<b>0.7860</b>	<b>0.7974</b>	<b>0.7910</b>	0.4832	0.4831	0.4922
		TDH [16]	0.7516	0.7577	0.7634	0.5198	0.5332	0.5399
		DSePH [9]	0.7422	0.7578	0.7760	0.4616	0.4882	0.5305
		QDCMH	0.7762	0.7725	0.7859	<b>0.5245</b>	<b>0.5398</b>	<b>0.5487</b>

are higher than the MAPs of most baseline methods in most cases, which demonstrates the superiority of QDCMH. We can also observe that SSAH outperforms than our proposed QDCMH in most cases, which is partly because SSAH takes self-supervised learning and generative adversarial networks into account during hash representation learning procedure. (2) The MAPs of QDCMH is always higher than the MAPs of TDH, which shows that quadruplet loss can better preserve semantic relevance than triplet loss in cross-modal hashing retrieval. (3) The MAPs of DSePH is always higher than the MAPs of SePH, which demonstrates that deep neural

networks have powerful features learning capacity. (4) Our proposed QDCMH can achieve better performance on MS-COCO 2014 dataset than on MIRFlickr-25K dataset, which is partly because the instances in MS-COCO 2014 dataset belong to 80 categories while the instances in MIRFlickr-25K dataset belong to 24 categories, and this makes the quadruplets generated from the MS-COCO 2014 dataset have better generalization ability than the quadruplets generated from the MIRFlickr-25K dataset.

Thirdly, to further investigate the performance of QDCMH, we plot the precision-recall curves and top  $N$ -precision curves

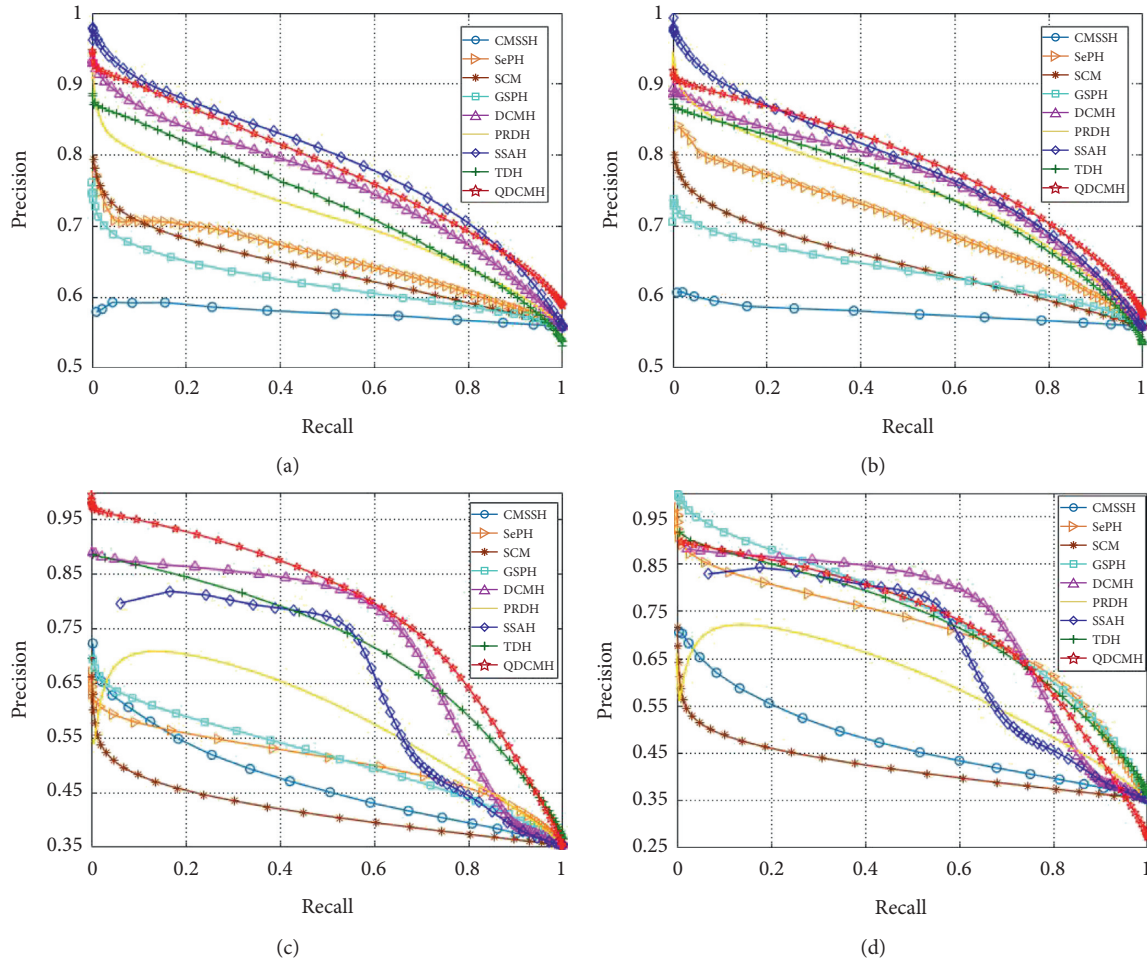


FIGURE 4: Precision-recall curves on datasets MIRFLICKR-25K and Microsoft COCO2014.

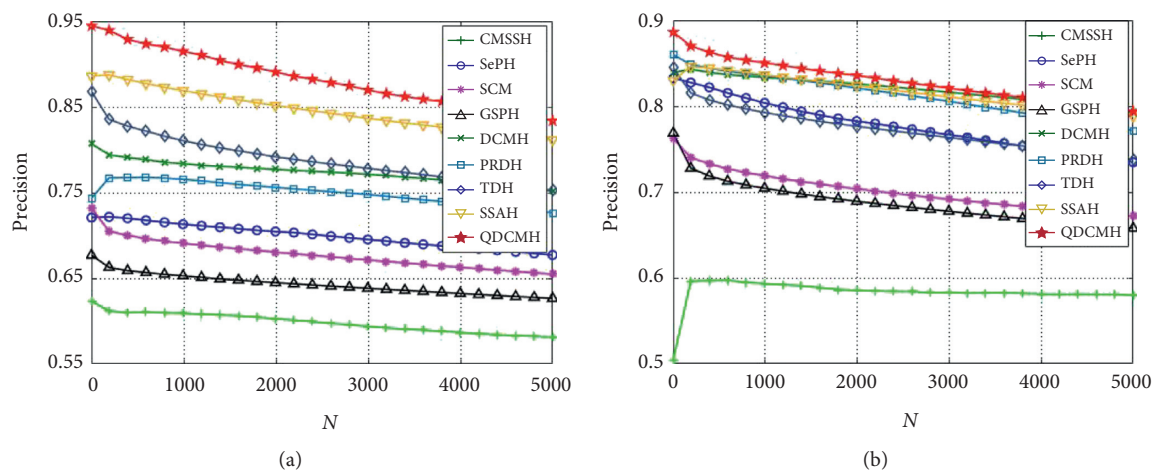


FIGURE 5: Continued.

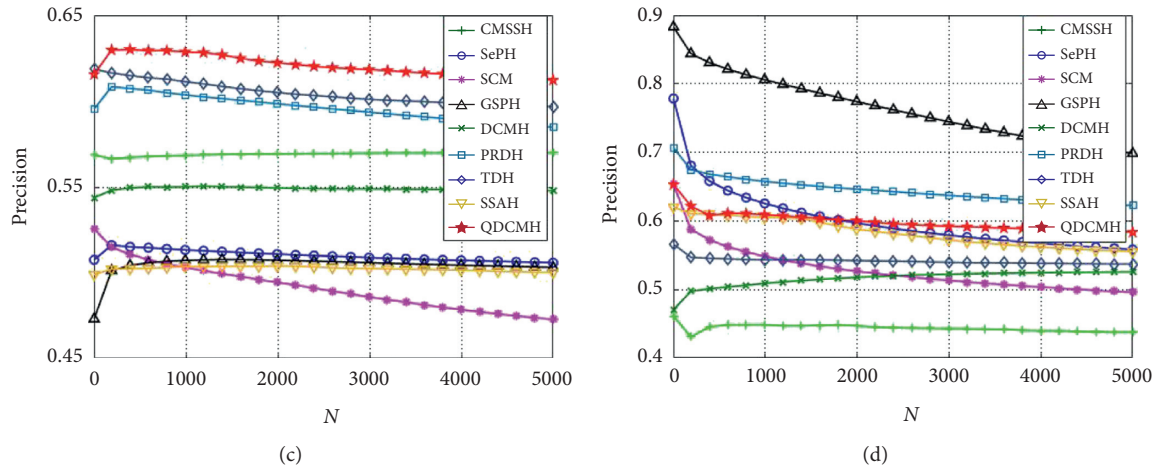


FIGURE 5: Top  $N$ -precision curves on datasets MIRFLICKR-25K and Microsoft COCO2014.

of QDCMH and baseline methods with hash code lengths 64 on datasets MIRFLICKR-25K, Microsoft COCO2014, respectively, as presented in Figures 4 and 5. From this figure, we can see that the precision-recall curves and top  $N$ -precision curves are nearly consistent with the MAPs in Table 2.

## 5. Conclusions

In this paper, we introduce a quadruplet loss into deep cross-modal hashing to fully preserve semantic relevance of original cross-modal quadruple instances and propose a quadruplet based deep cross-modal hashing method (QDCMH). QDCMH integrates quadruplet-based cross-modal semantic relevance preserving module, hash representation learning, and hash code generation into an end-to-end framework. Experiments on two benchmark cross-modal retrieval datasets demonstrate the efficiency of our proposed QDCMH.

## Data Availability

The experimental datasets and the related settings can be found in <https://github.com/SWU-CS-MediaLab/MLSPH>. The experimental codes used to support the findings of this study will be deposited in the github repository after the publication of this paper or can be provided by [xitaozou@sanxiau.edu.cn](mailto:xitaozou@sanxiau.edu.cn).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] Y. Peng, X. Huang, and Y. Zhao, "An overview of cross-media retrieval: concepts, methodologies, benchmarks, and challenges," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2372–2385, 2017.
- [2] K. Wang, Q. Yin, W. Wang, S. Wu, and L. Wang, "A comprehensive survey on cross-modal retrieval," *Multimedia*, <https://arxiv.org/abs/1607.06215>, 2016.
- [3] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2189–2201, 2019.
- [4] C. Deng, E. Yang, T. Liu, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Transaction on Image Processing*, vol. 28, Article ID 2903661, 2019.
- [5] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, "Shared predictive cross-modal deep quantization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5292–5303, 2018.
- [6] E. Yang, T. Liu, C. Deng, and D. Tao, "Adversarial examples for hamming space search," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1473–1484, 2018.
- [7] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3594–3601, San Francisco, CA, USA, June 2010.
- [8] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, pp. 2177–2183, Québec City, Québec, Canada, July 2014.
- [9] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3864–3872, Boston, MA, USA, June 2015.
- [10] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for n-label cross-modal retrieval," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4076–4084, Honolulu, HI, USA, July 2017.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 1097–1105, 2012.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

- [13] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3232–3240, Honolulu, HI, USA, July 2017.
- [14] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence 2017*, San Francisco, CA, USA, February 2017.
- [15] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4242–4251, Salt Lake City, UT, USA, June 2018.
- [16] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Transactions Image Processing*, vol. 27, no. 8, pp. 3893–3903, 2018.
- [17] J. Zhu, Z. Chen, L. Zhao, and S. Wu, "Quadruplet-based deep hashing for image retrieval," *Neurocomputing*, vol. 366, pp. 161–169, 2019.
- [18] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," *Computer Vision and Pattern Recognition*, pp. 403–412, 2017, <https://arxiv.org/abs/1704.01719>.
- [19] J. Deng, W. Dong, R. Socher et al., "A large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, June 2009.
- [20] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pp. 39–43, New York, NY, USA, October, 2008.
- [21] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *Proceedings of the European Conference on Computer Vision ECCV 2014*, pp. 740–755, Zurich, Switzerland, September 2014.
- [22] X. Wang, X. Zou, E. M. Bakker, and S. Wu, "Self-constraining and attention-based hashing network for bit-scalable cross-modal retrieval," *Neurocomputing*, vol. 400, pp. 255–271, 2020.
- [23] X. Zou, X. Wang, E. M. Bakker, and S. Wu, "Multi-label semantics preserving based deep cross-modal hashing," *Signal Processing Image Communication*, vol. 93, no. 9, Article ID 116131, 2021.

## Research Article

# Detecting COVID-19 in Chest X-Ray Images via MCFF-Net

Wei Wang <sup>1</sup>, Yutao Li <sup>1</sup>, Ji Li <sup>1</sup>, Peng Zhang <sup>2</sup>, and Xin Wang <sup>1</sup>

<sup>1</sup>School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

<sup>2</sup>School of Electronics and Communications Engineering, Sun Yat-Sen University, Shenzhen 518107, China

Correspondence should be addressed to Peng Zhang; zhangpeng5@mail.sysu.edu.cn and Xin Wang; wangxin@csust.edu.cn

Received 25 April 2021; Accepted 4 June 2021; Published 18 June 2021

Academic Editor: Nian Zhang

Copyright © 2021 Wei Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

COVID-19 is a respiratory disease caused by severe acute respiratory syndrome coronavirus (SARS-CoV-2). Due to the rapid spread of COVID-19 around the world, the number of COVID-19 cases continues to increase, and lots of countries are facing tremendous pressure on both public and medical resources. Although RT-PCR is the most widely used detection technology with COVID-19 detection, it still has some limitations, such as high cost, being time-consuming, and having low sensitivity. According to the characteristics of chest X-ray (CXR) images, we design the Parallel Channel Attention Feature Fusion Module (PCAF), as well as a new structure of convolutional neural network MCFF-Net proposed based on PCAF. In order to improve the recognition efficiency, the network adopts 3 classifiers: 1-FC, GAP-FC, and Conv1-GAP. The experimental results show that the overall accuracy of MCFF-Net66-Conv1-GAP model is 94.66% for 4-class classification. Simultaneously, the classification accuracy, precision, sensitivity, specificity, and F1-score of COVID-19 are 100%. MCFF-Net may not only assist clinicians in making appropriate decisions for COVID-19 diagnosis but also mitigate the lack of testing kits.

## 1. Introduction

Coronavirus disease 2019 (COVID-19) is a respiratory disease caused by severe acute respiratory syndrome coronavirus (SARS-CoV-2). Since its discovery in December 2019, the disease has spread rapidly around the world and is highly infectious. On March 11, 2020, the disease was declared a global pandemic by the World Health Organization (WHO) [1]. With the reopening of daily activities in countries around the world, the morbidity and mortality of COVID-19 have continued to increase, putting tremendous pressure on medical institutions and medical resources. Therefore, finding a quick and effective diagnosis method has become a top priority.

The current mainstream COVID-19 diagnosis technology is real-time reverse transcription polymerase chain reaction (RT-PCR) technology. However, the detection process is cumbersome and the diagnosis result has a high false-negative rate [2]. At the same time, chest imaging examinations, such as computed tomography (CT) and chest X-ray detection, also play a vital role in the early diagnosis of the disease [3]. Although the diagnostic

efficiency of COVID-19 is constantly improving, the current cost of testing and diagnosis is still at a relatively high level. By examining the patient's lung imaging images, the diagnosis efficiency of COVID-19 can be greatly accelerated, and the patient can be treated as soon as possible.

Some studies have shown that COVID-19 has obvious clinical imaging characteristics. The study of Zu et al. [2] showed that some patients with COVID-19 had lung opacity in chest CT images. Zhao et al. [4] proposed that most patients have ground glass opacity (GGO), and some patients have lung consolidation and vasodilatation in chest lesions. Li and Xia [5] proposed that the CT imaging lesions of COVID-19 patients showed signs of GGO, lung consolidation, thickened interlobular septa, and air bronchography. Compared with CT, chest X-Ray (CXR) diagnosis has the advantages of convenient detection process, low cost, and low ionizing radiation intensity [6], which is more patient-friendly and easy to promote in remote and underdeveloped areas. In addition, manual image reading is a time-consuming and error-prone task. In order to reduce the pressure of medical imaging physicians, it is necessary to propose an efficient and accurate COVID-19 detection method.

In recent years, deep learning has become one of the most popular research fields in artificial intelligence. Deep convolutional neural network (DCNN) has excellent performance in computer vision tasks such as image classification, image segmentation, and target detection. A wealth of research results has emerged in this field. For example, Wang et al. [7] proposed and improved a deep learning method for detecting colon polyp images and achieved good results. Wang et al. [8] introduced the dense connection idea of the DenseNet model in the MobileNet model and proposed a new type of image classification model Dense-MobileNet. On the basis of the original model, the accuracy of the image classification task is improved and the complexity of the model is reduced. Wang et al. [9] combined the dense connection idea with the full convolutional network FCN model, proposed a dense full convolutional network DFCN, and used this model to perform semantic segmentation tasks on the Chenzhou remote sensing image dataset, achieving good results. After the outbreak of the COVID-19 epidemic, the use of DCNN to detect COVID-19 has become a current hot research field. At the same time, many outstanding research results have emerged in this field. Based on the characteristics of CXR images, Wang et al. [10] designed the Channel Feature Weight Extraction module (CFWE) and proposed a new network structure CFW-Net on this basis, which has achieved a good classification effect. Wang et al. [11] designed a Multiattention Interaction Enhancement module (MAIE) and proposed a new convolutional neural network, MAI-Net. The overall accuracy and COVID-19 category accuracy were 96.42% and 100%, respectively, which were better than those of ResNet [12]. Based on the VGG19 [13] network model, Apostolopoulos and Mpesiana [14] conducted a three-category classification experiment on a dataset containing COVID-19 positive, common pneumonia, and normal CXR images, and the overall classification accuracy rate was 93.48%. Wang et al. [15] proposed a COVID-Net network model based on the PEPX structure and introduced the depthwise separable convolution [16] into the network. The accuracy of the 3-class classification was 93.3%, which reduced the amount of model parameters and had good classification performance. Khan et al. [17] proposed a CoroNet network model based on the structure of Xception [18] and conducted 2-class, 3-class, and 4-class classification experiments for CXR images. The classification accuracy rates were 99%, 95%, and 89.6%. On this basis, Hussain et al. [19] improved Khan's work and proposed the CoroDet network structure. The classification accuracy of 2-class, 3-class, and 4-class were 99.1%, 94.2%, and 91.2%, respectively.

Unlike conventional image classification tasks, CXR images have high interclass similarity and low intraclass variability. This kind of data characteristics can easily lead to model deviation and overfitting problems, reduce the generalization performance of the network, and increase the difficulty of image classification tasks. To solve these problems, the Parallel Channel Attention Feature Fusion Module (PCAF) is designed. Based on the PCAF module, a new convolutional neural network structure, MCFF-Net, is proposed. MCFF-Net is used to perform a 4-class

classification experiment on a dataset containing four types of image of COVID-19, normal, bacterial pneumonia, and viral pneumonia, with excellent performance. Compared with the deep learning methods in other documents, MCFF-Net has higher classification accuracy and stronger generalization ability.

## 2. CNNs

In recent years, deep convolutional neural networks have been widely used in the field of computer vision, and its basic structure is shown in Figure 1. In view of the brand-new techniques such as ReLU [20], LRN [20], and Dropout [21], AlexNet [22] designed by Hinton and AlexKrizhevsky won the championship in 2012 ImageNet Challenge, with excellent performance. At the same time, AlexNet reduces the problem of network overfitting and enhances the generalization ability of the model. In 2014, Simonyan and Zisserman proposed the visual geometry group network (VGGNet) [14], which increased the network depth to 19 layers by alternately using  $3 \times 3$  convolution kernels and  $2 \times 2$  maximum pooling layers, significantly improving the network performance. Christian Szegedy et al. [23] designed the Inception module and constructed the GoogLeNet network based on this module. By increasing the width and depth, GoogLeNet also improves the utilization of the internal resources of the network and alleviates the problem of overfitting to a certain extent.

Increasing the network depth can improve network performance, but it can also cause some problems such as overfitting, network degradation, gradient disappearance, and gradient explosion. In 2015, He et al. [12] proposed the residual network named ResNet, which solved the degradation problem of the network through skip connection and increased the network depth to 1000 layers for the first time, making the deep convolutional neural network reach an unprecedented depth. Inspired by the residual network, the dense network named DenseNet was proposed by Huang et al. [24] in 2017 based on the idea of dense connections. By directly introducing short connections in any two layers to realize the reuse of features, it greatly reduces the amount of network parameters and effectively alleviates the problem of gradient disappearance of deep network.

## 3. PCAF Module

In order to relieve the pressure of current medical staff and improve the diagnostic speed of COVID-19, we adopt a convolutional neural network that can adaptively learn the feature information exhaustively to identify and classify CXR images. CXR images have high interclass similarity and low intraclass variability. These problems will lead to model deviation and overfitting as well as reduce the recognition ability and generalization performance of the network. Hence, the PCAF module has been designed, whose structure is shown in Figure 2.  $C$  is the number of channels related to the input feature map.  $H$  and  $W$  represent the height and width of the feature map, respectively. "r"

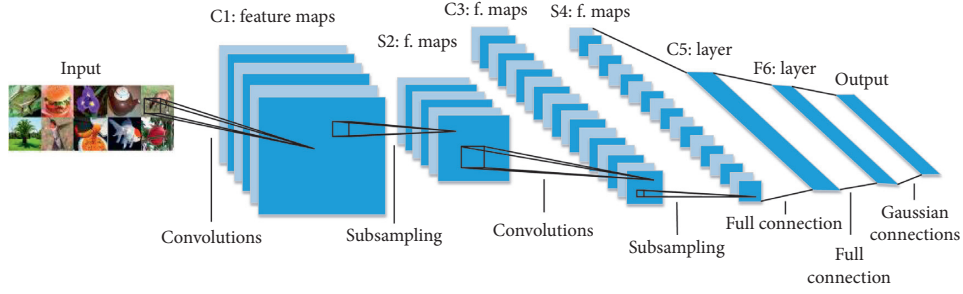


FIGURE 1: The basic structure of convolutional neural network [25].

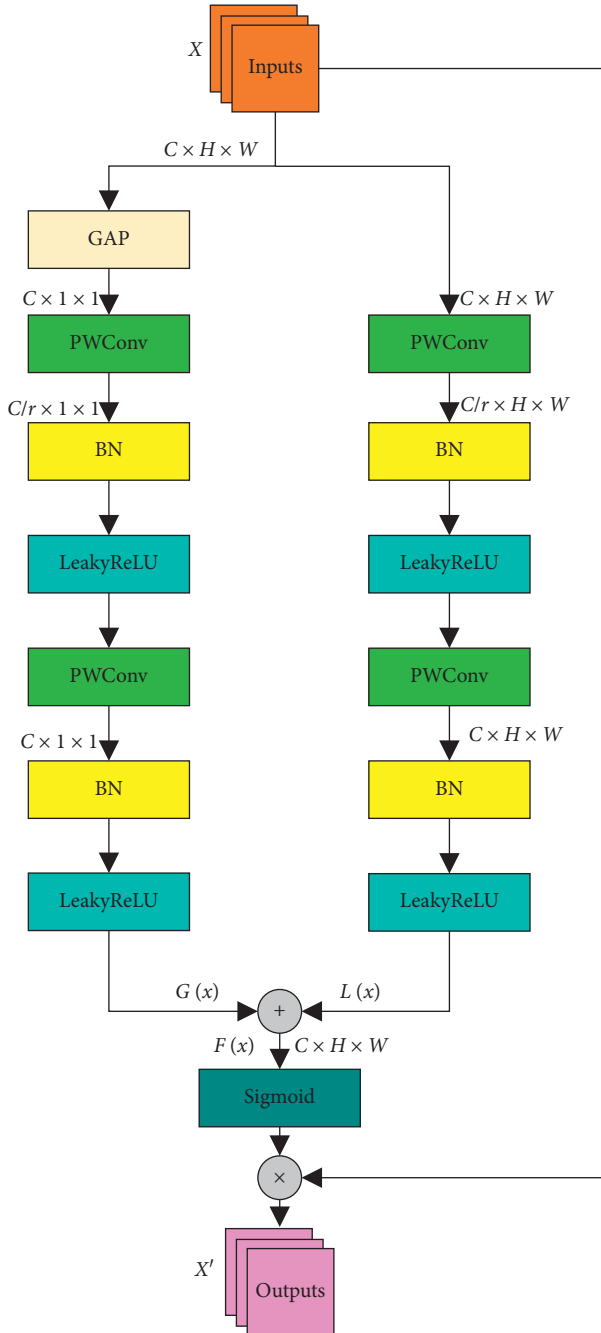


FIGURE 2: The structure of PCAF module.

represents the channel compression ratio. “GAP” [26] represents the global average pooling. “PWConv” represents the  $1 \times 1$  pointwise convolution. “BN” [27] is on behalf of batch normalization. “LeakyReLU” [28] and “Sigmoid” are activation functions. “ $\oplus$ ” represents the feature matrix bitwise addition operation. “ $\otimes$ ” represents the feature matrix bitwise multiplication operation.

The PCAF module is composed of two parallel branches, namely, the global feature extraction branch and the local feature extraction branch. The input feature map is imported into the two branches for feature extraction. Local feature extraction branch is composed of two PWConvs. The size of convolution kernel for the first PWConv is  $C/r \times 1 \times 1$ , compressing the channels of feature map to  $C/r$ , reducing the dimension of the feature map. The size of the second PWConv convolution kernel is  $C \times 1 \times 1$ , restoring the channels of the feature map to  $C$ , raising the dimension of the feature map.

Based on the above, the global feature extraction branch consists of one GAP layer and two PWConv layers. The GAP operation can compress the global information into a real number, which has the receptive field of global information to a certain extent.

Therefore, the global feature extraction branch focuses on extracting widely distributed global information in the feature map. The size of feature map in local feature extraction branch remains  $H \times W$ . It has not been compressed by the global average pooling from beginning to end. Consequently, more attention is paid to extract the local subtle information of the feature map.

The output features of the two branches can be expressed as

$$\begin{aligned} G(X) &= d\{\text{BN}[\text{PWConv}2[d[\text{BN}[\text{PWConv}1[\text{GAP}(X)]]]]]\}, \\ L(X) &= d\{\text{BN}[\text{PWConv}2[d[\text{BN}[\text{PWConv}1(X)]]]]\}, \end{aligned} \quad (1)$$

where BN represents batch normalization operation and  $d$  represents LeakyReLU activation function.

After output features  $L(X)$  and  $G(X)$  of the two branches are fused by matrix bitwise addition operation, the fusion feature  $F(X)$  is obtained by the sigmoid activation function, which can be described by the following formula:

$$F(X) = G(X) \oplus L(X). \quad (2)$$

The features of different scales are merged by  $F(X)$ . In this way, the weight of each channel in the feature map is recalibrated. The model can learn the weight coefficient of each channel in the global feature and the weight coefficient of each channel in the local feature, respectively.

Finally, the mask  $F(X)$  and the input feature map  $X$  are processed by matrix bitwise multiplication operation, and the output feature map  $X'$  is obtained. The formula is as follows:

$$X' = X \otimes \sigma(F(X)) = X \otimes \sigma[(L(X) \oplus G(X))], \quad (3)$$

where  $\sigma$  represents Sigmoid activation function.

After the input feature map is processed by the PCAF module, the network can learn more important information in a targeted manner, ignoring the secondary information.

#### 4. MCFF-NET

Based on the PCAF module, three convolutional neural networks with different depths are proposed: Multiscale Channel Feature Fusion Network (MCFF-Net), as shown in Table 1. When calculating the network depth in Table 1, a PCAF module is recorded as one layer, and the depth of the classifier in MCFF-Net is uniformly recorded as one layer. The ‘‘Conv’’ structure in Table 1 can be expressed as a composite structure including ‘‘convolution,’’ ‘‘batch normalization,’’ and ‘‘ReLU activation function.’’ The value after ‘‘Conv’’ represents the number of channels corresponding to the structure. The network diagram is shown in Figure 3.

In traditional convolutional neural networks such as AlexNet [22] and VGGNet [14], three fully connected layers (3 full connection layer, 3-FC) are used as classifiers. This can increase the nonlinear expression ability of network, accompanied by a large amount of memory occupation and high calculation overhead, which has caused a substantial increase in the amount of network parameters. In order to reduce the network parameters, our network uses a fully connected layer (1-FC) as the classifier to convert the computational overhead of the image recognition task to the convolutional layer, which reduces the burden of the fully connected layer.

Due to the extremely large number of features output by the convolutional layer, one fully connected layer as a classifier will cause excessive parameters. Therefore, we first reduce the output feature map size of the convolutional layer to  $1 \times 1$  through the GAP operation and then classify through the fully connected layer, which greatly reduces the amount of parameter of the network model. ‘‘GAP-FC’’ is used to represent this structure.

Besides,  $1 \times 1$  point convolution is considered to be inserted in front of the GAP structure, reducing the dimensionality of the output feature map at the end of the network. The classifier designed under this idea has nothing to do with the fully connected layer, thereby further reducing the amount of parameter. ‘‘Conv1-GAP’’ is used to represent this structure.

When using different depth networks and different classifiers to recognize CXR images, there are differences among the amount of parameter and calculation of the network. Take the 4-class classification task as an example, and suppose the output feature map size of the last

TABLE 1: MCFF-Net configuration.

MCFF-Net50		MCFF-Net66		MCFF-Net134	
		Conv7 $\times$ 7-64, stride 2			
		3 $\times$ 3 maxpooling, stride 2			
Conv3 $\times$ 3-64		Conv1 $\times$ 1-64		Conv1 $\times$ 1-64	
Conv3 $\times$ 3-64	$\times 3$	Conv3 $\times$ 3-64	$\times 3$	Conv3 $\times$ 3-64	$\times 3$
PCAF-64		Conv1 $\times$ 1-256		Conv1 $\times$ 1-256	
		PCAF-256		PCAF-256	
Conv3 $\times$ 3-128		Conv1 $\times$ 1-128		Conv1 $\times$ 1-128	
Conv3 $\times$ 3-128	$\times 4$	Conv3 $\times$ 3-128	$\times 4$	Conv3 $\times$ 3-128	$\times 4$
PCAF-128		Conv1 $\times$ 1-512		Conv1 $\times$ 1-512	
		PCAF-512		PCAF-512	
Conv3 $\times$ 3-256		Conv1 $\times$ 1-256		Conv1 $\times$ 1-256	
Conv3 $\times$ 3-256	$\times 6$	Conv3 $\times$ 3-256	$\times 6$	Conv3 $\times$ 3-256	$\times 23$
PCAF-256		Conv1 $\times$ 1-1024		Conv1 $\times$ 1-1024	
		PCAF-1024		PCAF-1024	
Conv3 $\times$ 3-512		Conv1 $\times$ 1-512		Conv1 $\times$ 1-512	
Conv3 $\times$ 3-512	$\times 3$	Conv3 $\times$ 3-512	$\times 3$	Conv3 $\times$ 3-512	$\times 3$
PCAF-512		Conv1 $\times$ 1-2048		Conv1 $\times$ 1-2048	
		PCAF-2048		PCAF-2048	
		Average pooling			
		Classifier, softmax			

convolutional layer in the network is  $H \times W \times D$ . When using a fully connected layer ‘‘1-FC’’ as the classifier, the parameter of the network is  $4 \times H \times W \times D + 4$ . When the ‘‘GAP-FC’’ structure is used as the classifier, the parameter of the network is  $D + D \times 4 + 4$ . When the ‘‘Conv1-GAP’’ structure is used as the classifier, the parameter of the network is  $H \times W \times 4 + D \times 4 + 4$ . When MCFF-Nets with different depths use different classifiers, the parameters are shown in Figure 4. Comparison of floating point of operations (FLOPs) is shown in Figure 5.

From Figure 4, the sort of classifier has great influence on the network parameters. In the case of the same network depth, the networks using the ‘‘1-FC’’ classifier are obviously larger than those using other classifiers. Therefore, using the ‘‘1-FC’’ classifier should be avoided as much as possible under the premise of ensuring the classification accuracy. In addition, the network depth also has a huge impact on the amount of network parameters. The parameters of MCFF-Net134-GAP-FC are 3.90 times that of MCFF-Net50-GAP-FC, and the parameters of MCFF-Net134-GAP-FC are 1.26 times that of MCFF-Net66-GAP-FC.

According to Figure 5, the computational cost is mainly determined by the depth of network. MCFF-Net134 is very computationally intensive. Compared with MCFF-Net66, MCFF-Net134 has a FLOPs increase of 94.87%. MCFF-Net66 has an increase of 53.18% compared to MCFF-Net50. Compared with MCFF-Net66, MCFF-Net134 has an increase of 194.87%, which is the largest increase in calculations. In conclusion, when there is no notable difference in recognition accuracy, in order to save computational cost, MCFF-Net66 has the highest cost performance.

## 5. Experiments and Results

5.1. Datasets. Since COVID-19 is a new type of disease, there is a lack of datasets suitable for this study. In this paper, we



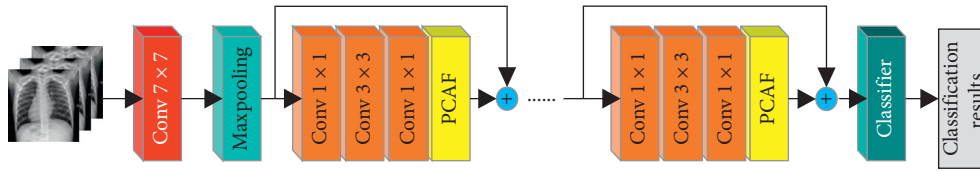


FIGURE 3: The network structure of MCFF-Net.

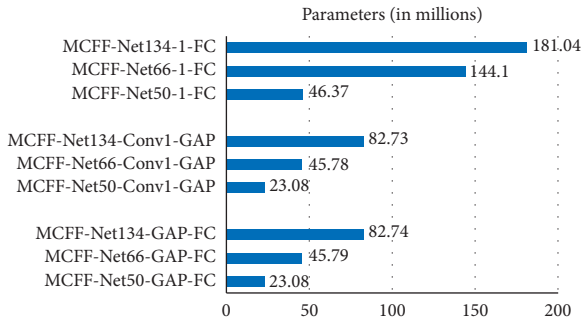


FIGURE 4: The parameter comparison of MCFF-Nets.

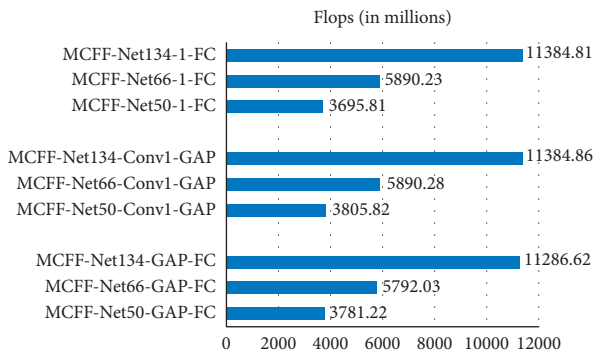


FIGURE 5: The comparison of floating points of operations (FLOPs).

have constructed a dataset by collecting CXR images from public image databases.

In order to further evaluate the generalization performance of the MCFF-Net, a 4-class dataset has been constructed. Dataset collects CXR images from five different public databases. These databases are (1) Actualmed-COVID-chestxray-dataset [29]; (2) COVID-19 Radiography Database [30]; (3) Figure 1-COVID-chestxray-dataset [31]; (4) Pneumonia Virus vs. Pneumonia Bacteria [32]; and (5) Chest X-ray Image [33]. Dataset contains four classes of CXR images, namely, COVID-19, normal, bacterial pneumonia, and viral pneumonia, totaling 5,985 images. There are 5300 images in training sets, including 800 COVID-19 patient images, 1300 normal images, 1600 viral pneumonia images, and 1600 bacterial pneumonia images. There are 741 images in the test sets, including 142 images of COVID-19 patients, 200 normal images, 202 bacterial pneumonia images, and 197 viral pneumonia images.

The eight sample images from the dataset that we have established are shown in Figure 6.

**5.2. Experimental Setup.** The experiments are carried out on the same platform and environment to ensure the credibility of the comparison results between different network models. Table 2 shows the software and hardware configuration information of the experimental platform. The batch size of the training set and the test set is both 16.

The learning rate annealing algorithm is introduced in the training process, and a larger learning rate is used in the initial stage of training. As the number of iterations increases, the learning rate is gradually reduced. This algorithm can avoid large fluctuations of classification accuracy in the later stage of training, so as to get closer to the optimal solution. After repeated experiments, we finally adjusted the parameter settings as follows: the initial learning rate is set to 0.001. Since the first 50 epochs, the learning rate decays twice as much as before and then decreases by 2 times every 50 epochs. A total of 300 epochs are used for training. In order to evaluate the performance of the model more objectively, we take the recognition accuracy of the last 10 epochs on test set to calculate the average value, which is used as the final classification accuracy.

**5.3. Evaluation Criteria.** In this section, we will explain the evaluation indicators used to quantify the classification performance of the network: accuracy, precision, sensitivity, specificity, and F1-score. In order to represent the above indicators, we also need to count the four numbers in the confusion matrix: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

**5.4. Experimental Results and Discussion.** In order to further verify the generalization ability of MCFF-Net66-Conv1-GAP in the CXR image recognition task, we increase the difficulty of the classification task and use the model to conduct experiments on dataset with four classes of CXR images. The training period is 300 epochs, which is divided into 6 stages, each with 50 epochs. We take test set recognition accuracy of the last 5 epochs in each stage and calculate the average value as the experimental result of the corresponding stage. Figure 7 shows the 4-class confusion matrix of MCFF-Net66-Conv1-GAP. Figure 8 shows the overall accuracy of MCFF-Net66-Conv1-GAP in each stage of the 4-classification experiment.

According to Figures 7 and 8, the overall accuracy of the four classification tasks of MCFF-Net66-Conv1-GAP reaches 94.6%, showing that the MCFF-Net has excellent classification performance in CXR image recognition tasks. In the discussion of Introduction, we briefly described a variety of COVID-19 detection methods proposed by

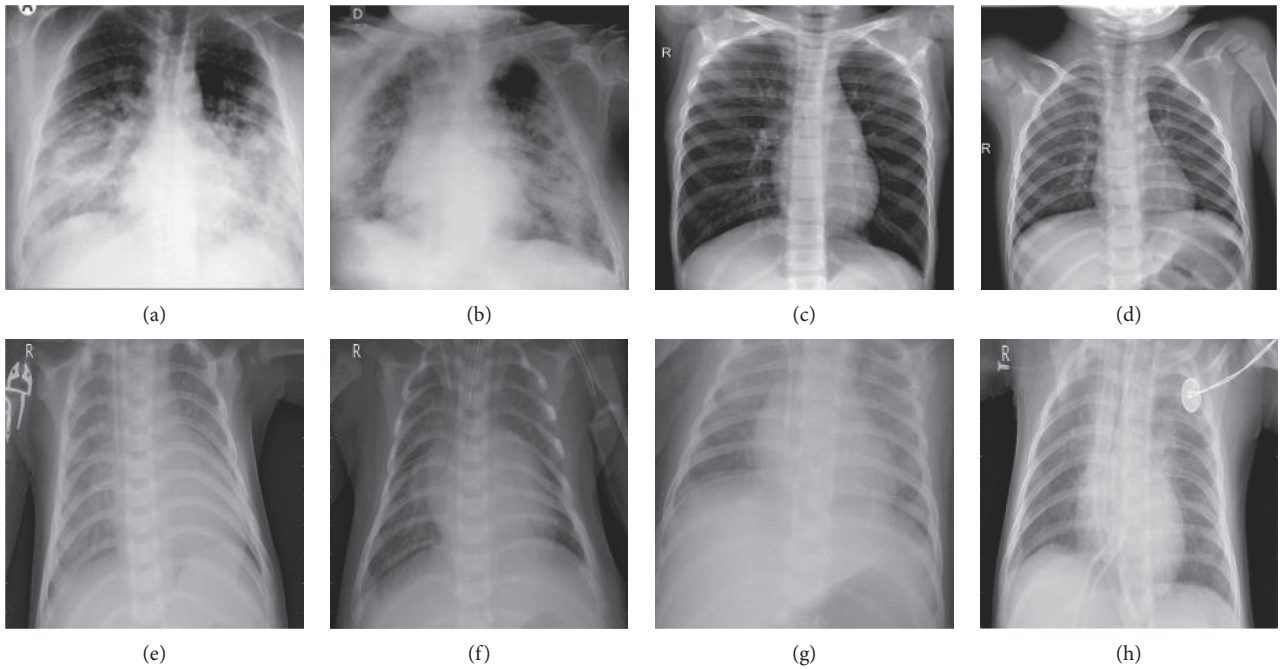


FIGURE 6: Chest X-ray images. (a) COVID-19, (b) COVID-19, (c) normal, (d) normal, (e) pneumonia—bacteria, (f) pneumonia—bacteria, (g) pneumonia—viral and (h) pneumonia—viral.

TABLE 2: Experimental platform configuration.

Attributes	Configuration information
Operating system	Ubuntu 18.04.5 LTS
CPU	Intel (R) Xeon (R) silver 4214 CPU @ 2.20 GHz
GPU	GeForce RTX 2080
CUDNN	CUDNN 7.5.0
CUDA	CUDA 10.0.130
Frame	Fastai
IDE	PyCharm
Language	Python

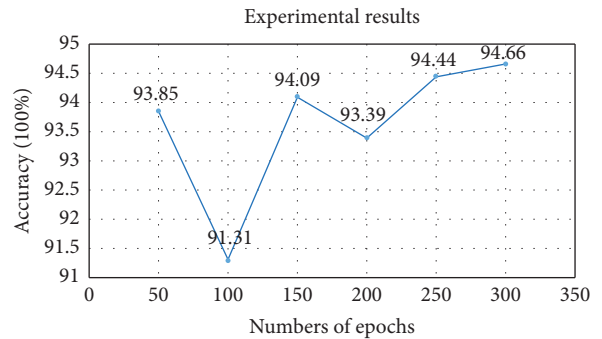


FIGURE 8: The 4-class accuracy rate of MCFF-Net66-Conv1-GAP.

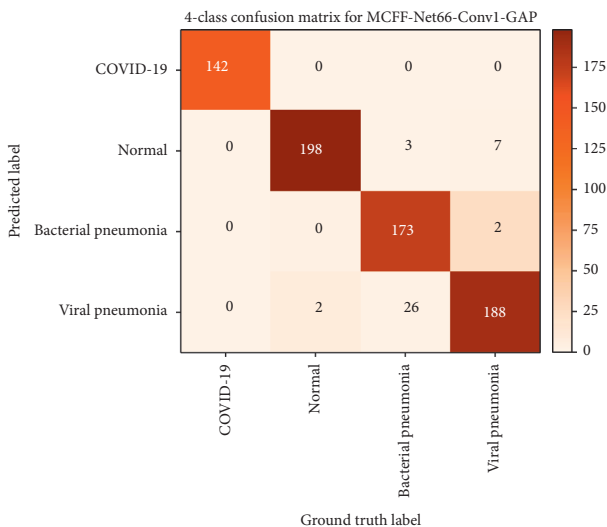


FIGURE 7: The 4-class confusion matrix of the MCFF-Net66-Conv1-GAP.

researchers from various regions of the world. Some models are suitable for 2-class classification, and some models are suitable for multiclass classification. Hence, the model MCFF-Net66-Conv1-GAP is compared with the methods of Khan [17], Hussain [19], Mangal [34], and Joshi [35]. The comparison results are shown in Table 3.

According to Table 4, our proposed network model MCFF-Net66-Conv1-GAP can efficiently help classify CXR images of COVID-19-positive patients, normal, and ordinary pneumonia patients. What is more, the overall accuracy, sensitivity, specificity, and *F1*-score of COVID-19 images have reached 100%.

The various methods in Table 3 use different numbers of CXR images from different data sources for training. The number of images used for training is shown in the fourth column. When there are four values in the number of images column, then the first value indicates the number of COVID-19 images, the second value indicates the number of viral pneumonia images, the third value indicates the

TABLE 3: Accuracy comparison of our proposed method with other existing deep learning methods.

Study	Architecture	Accuracy 4-class (%)	Number of images	# of parameters (in millions)
Khan et al. [17]	CoroNet	89.60	284, 327, 330, 310	33
Hussain et al. [19]	CoroDet	91.20	500, 400, 400, 800	N/A
Mangal et al. [34]	CovidAID	87.20	115, 1337, 2530, 1341	N/A
Joshi et al. [35]	DarkNet-53	76.46	659, 1493, 2772, 1660	N/A
Proposed method	MCFF-Net	94.66	800, 1600, 1600, 1300	45.78

TABLE 4: Average class-wise accuracy, precision, recall, specificity, F1-score of 4-class MCFF-Net66-Conv1-GAP (%).

Class	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)
COVID-19	100	100	100	100	100
Normal	98.38	99	95.19	99.63	97.06
Viral pneumonia	95.82	85.64	98.86	94.88	91.78
Bacterial pneumonia	95.01	95.43	87.04	98.29	91.04
Average	97.3	95.02	95.27	98.2	94.79

number of bacterial pneumonia images, and the fourth value indicates the number of normal images. “N/A” indicates an item of information that is not disclosed in the above-mentioned documents.

Compared with other methods, we have used the largest number of COVID-19 images to train our MCFF-Net model and have got 94.66% classification accuracy in the 4-class recognition task, which are higher than other methods in Table 3. This shows that the MCFF-Net has better performance in CXR image classification tasks.

**5.5. Experimental Analysis.** According to the experimental results in Section 5.4, we can find that, in the 4-class classification experiments, MCFF-Net66-Conv1-GAP has been chosen to conduct a 4-class classification experiment. The experimental results are compared with other existing methods. The overall accuracy of the 4-class classification experiment is 94.66%. In conclusion, the overall performance is better than other existing methods.

Through experimental analysis, it can be seen that, in the CXR image classification task of COVID-19, the network depth should be kept moderate. If the network is too shallow, it is hard to fully extract the feature information. If the network is too deep, while greatly increasing the amounts of parameters and calculations, it is also likely to overfitting and gradient explosion problems.

Because CXR images have high similarity between classes and low intraclass variability, it is easy to cause model deviation and overfitting, which increases the difficulty of image classification tasks. Therefore, this paper designs a PCAF module, which is composed of two parallel branches, and includes “GAP” and “PWConv” structures. After the input feature map is processed by the PCAF module, the output feature map will both have global and local information in the image, which improves the feature extraction capability of the network.

## 6. Conclusions

In this paper, a Parallel Channel Attention Feature Fusion (PCAF) module is designed according to the characteristics

of CXR images. And based on this module, a new convolutional neural network structure MCFF-Net is proposed to classify CXR images in order to diagnose and detect COVID-19 cases. Through the analysis and comparison of the experimental results, we believe that MCFF-Net66-Conv1-GAP has the highest application value. The overall accuracy of the 4-class classification experiment and the COVID-19 image recognition accuracy have reached 94.66% and 100%, respectively. Despite the fact that good results have been achieved, MCFF-Net still needs clinical research and testing. We will overcome the limitations of hardware conditions and train the MCFF-Net with a larger dataset to further improve its classification accuracy.

## Data Availability

All datasets in this article are public datasets and can be found respectively in references [29–33].

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This research was funded by the National Defense Science and Technology Innovation Special Zone Project (2019XXX00701), the Natural Science Foundation of Hunan Province, China (2019JJ80105), the Changsha Science and Technology Project (kq2004071), the Hunan Graduate Student Innovation Project (CX20200882), and the Shenzhen Science and Technology Project (KQTD20190929172704911).

## References

- [1] World Health Organization, *WHO Updates on COVID-19*, World Health Organization, Geneva, Switzerland, 2020, <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen>.
- [2] Z. Y. Zu, M. D. Jiang, P. P. Xu et al., “Coronavirus disease 2019 (COVID-19): a perspective from China,” *Radiology*, vol. 296, no. 2, p. E15, 2020.

- [3] J. P. Kanne, B. P. Little, J. H. Chung et al., “Essentials for radiologists on COVID-19: an update—radiology scientific expert panel,” *Radiology*, vol. 296, no. 2, Article ID 200527, 2020.
- [4] W. Zhao, Z. Zhong, X. Xie, Q. Yu, and J. Liu, “Relation between chest CT findings and clinical conditions of coronavirus disease (COVID-19) pneumonia: a multicenter study,” *AJR. American Journal of Roentgenology*, vol. 214, no. 5, pp. 1072–1077, 2020.
- [5] Y. Li and L. Xia, “Coronavirus disease 2019 (COVID-19): role of chest CT in diagnosis and management,” *American Journal of Roentgenology*, vol. 214, no. 6, pp. 1280–1286, 2020.
- [6] World Health Organization, *Use of Chest Imaging in COVID-19: A Rapid Advice Guide, 11 June 2020*, World Health Organization, Geneva, Switzerland, 2020.
- [7] W. Wang, J. Tian, C. Zhang, Y. Luo, X. Wang, and J. Li, “An improved deep learning approach and its applications on colonic polyp images detection,” *BMC Medical Imaging*, vol. 20, no. 1, p. 83, 2020.
- [8] W. Wang, Y. Li, T. Zou et al., “A novel image classification approach via dense-mobileNet models,” *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.
- [9] W. Wang, Y. Yang, J. Li, Y. Hu, Y. Luo, and X. Wang, “Woodland labeling in Chenzhou, China, via deep learning approach,” *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 1393–1403, 2020.
- [10] W. Wang, H. Liu, J. Li et al., “Using CFW-Net deep learning models for X-ray images to detect COVID-19 patients,” *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 199–207, 2021.
- [11] W. Wang, X. Huang, J. Li, P. Zhang, and X. Wang, “Detecting COVID-19 patients in X-ray images based on MAI-nets,” *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1607–1616, 2021.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings Of the IEEE Conference On Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings Of the International Conference On Learning Representations*, Banff, Canada, April 2014.
- [14] I. D. Apostolopoulos and T. A. Mpesiana, “Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks,” *Physical and Engineering Sciences in Medicine*, vol. 43, 2020.
- [15] L. Wang, Z. Q. Lin, and A. Wong, “Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images,” *Scientific Reports*, vol. 10, no. 1, pp. 19549–19612, 2020.
- [16] A. G. Howard, M. Zhu, B. Chen et al., “Mobilenets: efficient convolutional neural networks for mobile vision applications,” 2017, <https://arxiv.org/abs/1704.04861>.
- [17] A. I. Khan, J. L. Shah, M. M. Bhat et al., “Coronet: a deep neural network for detection and diagnosis of COVID-19 from chest x-ray images,” *Computer Methods and Programs in Biomedicine*, vol. 196, Article ID 105581, 2020.
- [18] F. Chollet, “Xception: deep learning with depthwise separable convolutions,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, Honolulu, HI, USA, July 2017.
- [19] E. Hussain, M. Hasan, M. A. Rahman et al., “CoroDet: a deep learning based classification for COVID-19 detection using chest X-ray images,” *Chaos, Solitons & Fractals*, vol. 142, Article ID 110495, 2020.
- [20] Y. Lecun, L. Bottou, Y. Bengio et al., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [21] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proceedings Of the International Conference On Machine Learning*, pp. 807–814, Haifa, Israel, January 2010.
- [22] G. E. Hinton, N. Srivastava, A. Krizhevsky et al., “Improving neural networks by preventing co-adaptation of feature detectors,” 2012, <https://arxiv.org/abs/1207.0580>.
- [23] A. Krizhevsky, I. Sutskever, G. E. Hinton et al., “ImageNet classification with deep convolutional neural networks,” in *Proceedings of the International Conference on Neural Information Processing Systems*, vol. 1, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.
- [24] C. Szegedy, L. Wei, J. Yangqing et al., “Going deeper with convolutions,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Chapel Hill, CA, USA, June 2015.
- [25] G. Huang, Z. Liu, V. D. M. Laurens et al., “Densely connected convolutional networks,” in *Proceedings Of the IEEE Conference Computer Vision and Pattern Recognition*, pp. 2261–2269, Honolulu, HI, USA, July 2017.
- [26] M. Lin, Q. Chen, S. Yan et al., “Network in network,” in *Proceedings Of the International Conference On Learning Representations*, Banff, Canada, April 2014.
- [27] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, July 2015.
- [28] B. Xu, N. Wang, T. Chen, and L. Mu, “Empirical evaluation of rectified activations in convolutional network,” 2015, <https://arxiv.org/pdf/1505.00853.pdf>.
- [29] Agchung, “Actualmed-COVID19-chestxray-dataset,” <https://github.com/agchung/Actualmed-COVID-chestxray-dataset>.
- [30] M. Chowdhury, T. Rahman, A. Khandakar et al., “Can AI help in screening viral and COVID-19 pneumonia?” *IEEE Access*, vol. 8, pp. 132665–132676, 2020, <https://www.kaggle.com/tawfifurrahman/covid19-radiography-database>.
- [31] Agchung, “Figure1-COVID-chestxray-dataset,” <https://www.kaggle.com/prottoysaha99/figure1covidchestxraydataset>.
- [32] M. Masdar, “Pneumonia virus vs. pneumonia Bacteria,” <https://www.kaggle.com/muhammadmasdar/pneumonia-virus-vs-pneumonia-bacteria>.
- [33] T. Dincer, “Chest X-ray images,” <https://www.kaggle.com/tolgadincer/abeled-chest-xray-images>.
- [34] A. Mangal, S. Kalia, H. Rajgopal et al., “CovidAID: COVID-19 detection using chest X-ray,” 2020, <https://arxiv.org/abs/2004.09803>.
- [35] R. C. Joshi, S. Yadav, V. K. Pathak et al., “A deep learning-based COVID-19 automatic diagnostic framework using chest X-ray images,” *Biocybernetics and Biomedical Engineering*, vol. 41, no. 1, pp. 239–254, 2021.

## Research Article

# A Multipulse Radar Signal Recognition Approach via HRF-Net Deep Learning Models

Ji Li,<sup>1</sup> Huiqiang Zhang,<sup>1</sup> Jianping Ou ,<sup>2</sup> and Wei Wang <sup>1</sup>

<sup>1</sup>School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

<sup>2</sup>ATR Key Lab, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Jianping Ou; [oujianping@nudt.edu.cn](mailto:oujianping@nudt.edu.cn)

Received 24 March 2021; Accepted 26 May 2021; Published 3 June 2021

Academic Editor: Nian Zhang

Copyright © 2021 Ji Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the field of electronic countermeasure, the recognition of radar signals is extremely important. This paper uses GNU Radio and Universal Software Radio Peripherals to generate 10 classes of close-to-real multipulse radar signals, namely, Barker, Chaotic, EQFM, Frank, FSK, LFM, LOFM, OFDM, P1, and P2. In order to obtain the time-frequency image (TFI) of the multipulse radar signal, the signal is Choi-Williams distribution (CWD) transformed. Aiming at the features of the multipulse radar signal TFI, we designed a distinguishing feature fusion extraction module (DFFE) and proposed a new HRF-Net deep learning model based on this module. The model has relatively few parameters and calculations. The experiments were carried out at the signal-to-noise ratio (SNR) of  $-14 \sim 4$  dB. In the case of  $-6$  dB, the recognition result of HRF-Net reached 99.583% and the recognition result of the network still reached 97.500% under  $-14$  dB. Compared with other methods, HRF-Nets have relatively better generalization and robustness.

## 1. Introduction

The external electromagnetic environment is becoming more and more complex, which brings severe challenges to electronic reconnaissance and electronic countermeasures systems. In the process of electronic countermeasures, the rapid and accurate identification of intercepted signals can give priority to the right to control information. However, the intercepted enemy signal is not only a single pulse signal, but also a multipulse signal, so the identification of multipulse is also extremely important.

Traditional radar signal recognition technology usually uses pulse description words (PDW) to match conventional parameters and designs feature extraction algorithms and classifiers for recognition. Wenqiang Zhang et al. [1] designed a TPOT-LIME algorithm, which can recognize radar signals from multiple aspects. Krzysztof Konopko et al. [2] used Wigner-Ville distribution to perform time-frequency analysis on the signal, then used a probability density function estimator to extract feature vectors, and finally used a statistical classifier to recognize radar signals. The

recognition accuracy is better, but the recognized signal classes are less. Jian Guo et al. [3] designed an FCBF-AdaBoost algorithm to identify radar signals and achieved good results. Qiang Guo et al. [4] designed a method that combines the main ridge slice and cloud model and constructed a feature vector for radar signal recognition, which has a high recognition rate. Jingchao Li and Ying [5] designed an entropy feature algorithm. The algorithm described the distribution features of different classes of radar signals by extracting odd-spectrum Shannon entropy and odd-spectrum exponential entropy features and had a higher recognition rate under low SNR.

However, with the increasingly serious external interference, the signal features are easily submerged by external interference. The traditional radar signal recognition method also needs to carry out complex feature design, which is difficult to achieve high recognition results. With the development of deep learning, Convolutional Neural Networks (CNNs) have been widely used. The network is widely used in image classification, semantic segmentation, target detection, and other directions. Muqing Zhang et al.

[6] designed an algorithm based on stacked autoencoders and support vector machines (SVM). This method obtained the time-frequency diagram of radar signals through Choi-Williams distribution, then used stacked autoencoders to automatically extract features, and finally completed signal recognition through SVM. Shunjun Wei et al. [7] designed a new type of network combining shallow CNN, LSTM, and deep DNN, which has a good recognition effect on a variety of radar signals. Li ji et al. [8] proposed an IIF-Net deep learning model, which achieved good recognition effect under low SNR. Guo, Limin et al. [9] designed an improved AlexNet network, and through time-frequency analysis of the signal, the overall recognition rate is higher under low SNR. Yihan Xiao et al. [10] designed a feature fusion algorithm, combined with an improved CNN, and got better recognition results.

In this paper, GNU Radio, USRP N210, and USRP-LW N210 are used to generate close-to-real radar signals with high reliability. 10 classes of multipulse radar signals are generated between  $-14$  and  $4$  dB in SNR, namely, Barker, Chaotic, EQFM, Frank, FSK, LFM, LOFM, OFDM, P1, and P2. Various signals through the CWD are used to generate two-dimensional TFIs. Different radar signals TFI have larger repetitive similar regions, while the distinctive feature regions are relatively small. In order to solve the above difficult, this paper designs a distinguishing feature fusion extraction module (DFFE) and proposes a new high-resolution feature fusion extraction network (HRF-Net) based on this module.

## 2. DFFE Module and HRF-Nets

**2.1. CNNs.** CNNs can extract target features adaptively. Figure 1 shows its network architecture [11]. Wang Wei et al. [12] gave a detailed analysis and introduction to CNN. In semantic segmentation, CNN can extract image features and achieve image pixel-level classification [13]. In order to improve the recognition effect, the image can be pre-processed, such as superresolution reconstruction [14], and attention mechanism can also be introduced to improve the performance of the network [15]. AlexNet [16] applies ReLU, LRN [17], and Dropout [18] at the same time. Simonyan and Zisserman [19] proposed the  $3 \times 3$  small convolution filter in visual geometry group networks (VGGNets), and the network reached 19 layers. But when the network has been trained enough, the performance of the network will decrease instead. Residual net (ResNet) [20] used skip connections to solve this problem and continued to increase the depth of the network.

Affected by the ResNet, Huang, Gao et al. [21] designed a dense connection mechanism that can connect all layers to each other and achieved reuse feature. Wang et al. combined DenseNet and MobileNet [22] to design a Dense-MobileNet [23], which achieved higher recognition rate and reduced the amount of network parameters and calculations.

**2.2. DFFE Module.** The TFIs of different radar signals have larger repeating similar regions, and the distinguishing

feature regions are smaller. Therefore, this paper designs a distinguishing feature fusion extraction module (DFFE). Figure 2 shows its structure.

First, in the spatial dimension, MaxPool and AvgPool are simultaneously performed on the input feature map. The feature map is compressed in the spatial dimension to obtain two one-dimensional vectors, which, respectively, represent the channel weight coefficients of the two feature maps. Then, through the ReLU activation function, the two channel weight coefficients are added together. A comprehensive analysis is performed to highlight the highly correlated channels, suppress the irrelevant channels, and focus on which input channels are more distinguishable. Then, we use the Sigmoid activation function. Finally, the one-dimensional channel weight vector is multiplied by the input feature map, keeping the input size unchanged, so the channel weight feature map Out1 is obtained.

In the channel dimension, MaxPool and AvgPool are performed on Out1. The channel dimension is compressed to obtain two two-dimensional spatial weight feature matrices, which are stitched together according to the channel dimension. A feature map with 2 channels is obtained. After that, Conv7 is used for convolution operation. Then, Sigmoid function is used for activation to obtain a comprehensive two-dimensional spatial weight feature matrix. It emphasizes the spatial position of high correlation and weakens the spatial position of less correlation, focusing on which areas of the input image are more distinguishable. Finally, the obtained two-dimensional feature matrix is multiplied with Out1, keeping the input size unchanged, so the feature map Out2, which is more distinguishable in space and channel, is obtained.

As the network deepens, problems such as image information loss and network degradation will occur. Adding skip connections can solve these problems. Therefore, we add the obtained feature map Out2 to the original input feature map "Base." And after the activation function ReLU, we obtain the feature map Out3 with complete information and more spatial and channel resolution. Next, we perform multiscale feature extraction on Out3 by using "Conv7," "Conv5," "Conv3," and "Conv1," respectively. Larger convolution kernels have larger receptive fields and strong semantic information representation ability. Smaller convolution kernels have smaller receptive fields, strong geometric detail information representation ability, and high resolution. Therefore, a variety of sizes of convolution kernels are used to perform feature fusion extraction on Out3, so the high-resolution features are obtained.

**2.3. HRF-Nets.** Based on the DFFE module, we propose three deep convolutional neural network structures, namely, high-resolution feature fusion extraction networks (HRF-Nets), which are HRF-Net157, HRF-Net187, and HRF-Net217. Among them, C-[MaxPool, AvgPool] represents the compression of the image spatial dimensions to obtain a feature map with channel weight coefficients. S-[MaxPool, AvgPool] means compressing the image channel dimensions

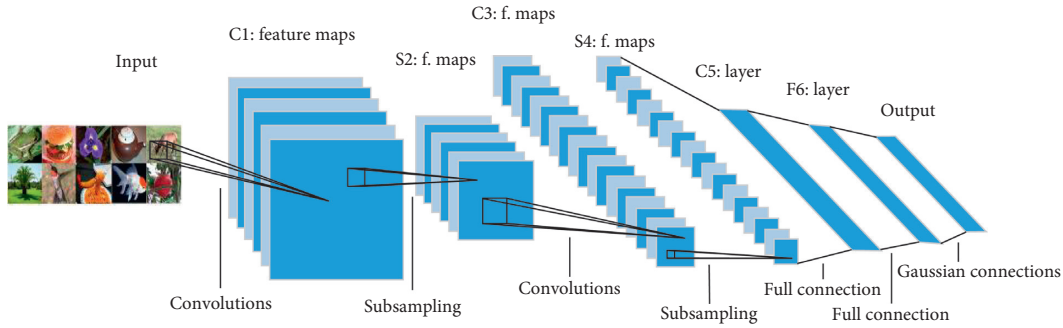


FIGURE 1: The basic architecture of CNN.

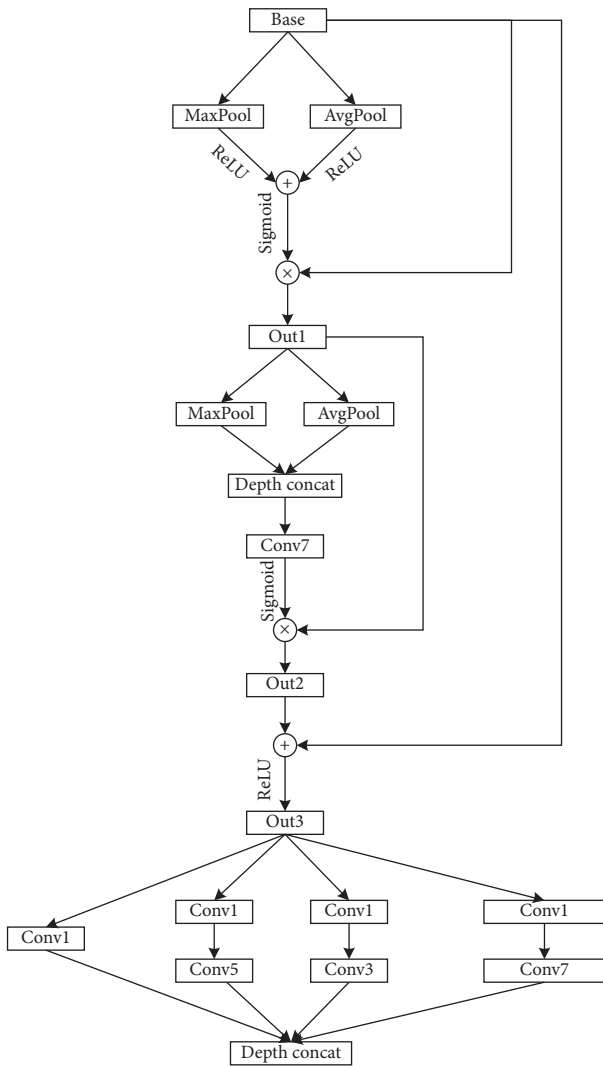


FIGURE 2: Structure of DFFE.

to obtain a feature map with spatial weight coefficients. Table 1 is the network structure.

The field of radar electronic countermeasures has high requirements for time delay, so the network needs smaller computational costs. Many classic CNNs use three-layer fully connected layers such as AlexNet and VGGNets, which have a high computational cost and take a long

time. Therefore, this paper first adopts the Global Average Pooling (GAP) [24] and then uses single-layer full connection, which obviously reduces the calculation cost and makes the network have relatively high real-time performance.

**2.4. Network Complexity.** The classifier occupies a large part of the calculation and parameter amount in the network, and the difference of the classifier greatly affects the performance and calculation cost of the network. In this paper, 10 classes of multipulse radar signals are recognized. Assuming that the output of the last layer of the network is  $H \times W \times D$ . The parameter amount of using three-layer full connection as the classifier is  $16,818,184 + 4096 \times H \times W \times D$ , the parameter amount of using single-layer full connection is  $H \times W \times D \times 10 + 10$ , and the parameter amount of using GAP is  $D + D \times 10 + 10$ .

Figure 3 shows the amount of parameters for different networks, and Figure 4 shows the amount of calculation.

According to Figure 3, it can be seen that when the network depth gradually increases, the amount of network parameters of the same type is gradually increasing, indicating that the network depth affects the size of the parameter amount to a certain extent. The three-layer fully connected layer is the classifier of VGGNets. The HRF-Nets classifier uses GAP plus a single-layer fully connected. Although the VGG13 network has only 13 layers, its parameter quantity is 4.18 times that of HRF-Net157, 3.64 times that of HRF-Net187, and 3.16 times that of HRF-Net217. Therefore, the classifier is a key factor affecting the amount of network parameters. The parameters of SKNet152, SEnet152, and ResNet152 are larger than those of HRF-Net157. The parameter of ResNet152 is 1.89 times that of HRF-Net157, which has more parameters of about 28.37 million.

According to Figure 4, it can be seen that the VGGNets have a huge amount of calculation. The 13-layer VGG network has a floating point calculation amount of 11.321 billion, which is 1.56 times that of the 157-layer HRF-Net. Compared with the ResNet152, HRF-Net157 has a deeper depth. But the calculation of ResNet152 is 1.59 times that of HRF-Net157, which has an increase of 4.309 billion. This is because ResNet152 uses a large number of convolutional layers. Without considering the bias, the calculation amount of the convolutional layer is  $(2 * C_{int} * K^2 - 1) * C_{out} * H_{out} * W_{out}$ . HRF-Nets have many pooling layers, and the calculation amount of the pooling layer

TABLE 1: HRF-Nets configurations.

HRF-Net157		HRF-Net187		HRF-Net217	
		Conv7-64, stride:2 3×3Maxpool, stride:2			
Conv1-64		Conv1-64		Conv1-64	
Conv3-64		Conv3-64		Conv3-64	
C-[MaxPool, AvgPool]	×3	C-[MaxPool, AvgPool]	×3	C-[MaxPool, AvgPool]	×3
S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]	
Conv7-64		Conv7-64		Conv7-64	
Conv1-256		Conv1-256		Conv1-256	
<hr/>					
<i>DFFE-256</i>					
Conv1-128		Conv1-128		Conv1-128	
Conv3-128		Conv3-128		Conv3-128	
C-[MaxPool, AvgPool]	×7	C-[MaxPool, AvgPool]	×8	C-[MaxPool, AvgPool]	×8
S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]	
Conv7-128		Conv7-128		Conv7-128	
Conv1-512		Conv1-512		Conv1-512	
<hr/>					
<i>DFFE-512</i>					
Conv1-256		Conv1-256		Conv1-256	
Conv3-256		Conv3-256		Conv3-256	
C-[MaxPool, AvgPool]	×10	C-[MaxPool, AvgPool]	×14	C-[MaxPool, AvgPool]	×19
S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]	
Conv7-256		Conv7-256		Conv7-256	
Conv1-1024		Conv1-1024		Conv1-1024	
<hr/>					
<i>DFFE-1024</i>					
Conv1-512		Conv1-512		Conv1-512	
Conv3-512		Conv3-512		Conv3-512	
C-[MaxPool, AvgPool]	×3	C-[MaxPool, AvgPool]	×3	C-[MaxPool, AvgPool]	×3
S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]		S-[MaxPool, AvgPool]	
Conv7-512		Conv7-512		Conv7-512	
Conv1-2048		Conv1-2048		Conv1-2048	
Classifier, Softmax					

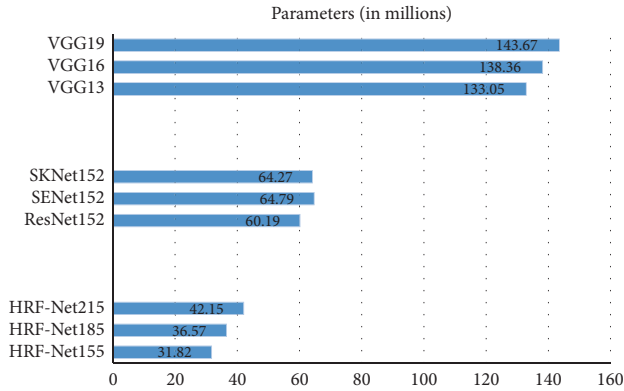


FIGURE 3: Parameters for different networks.

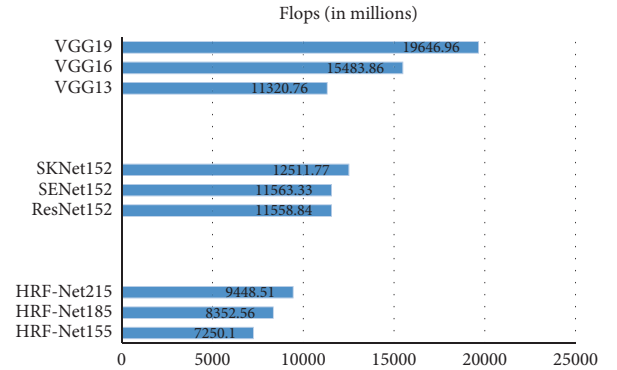


FIGURE 4: FLOPs.

is  $K^2 * C_{out} * H_{out} * W_{out}$ . Therefore, HRF-Net157 has a smaller amount of calculation than ResNet152. The calculation amount of HRF-Net217 is 30.32% more than that of HRF-Net157, and the calculation amount of HRF-Net187 is 15.21% more than that of HRF-Net157. Radar electronic countermeasure system requires low delay, especially in small devices such as missiles. The memory is insufficient, and the hardware conditions do not support too many parameters and calculations. The HRF-Net157 is relatively small in terms of parameters and calculations. Therefore, when the recognition result of the

signal differs slightly, HRF-Net157 has the highest cost performance and is a better choice.

### 3. Experimental Results

**3.1. Dataset.** In this paper, the multipulse radar signal is generated by GNU Radio USRP N210, and USRP-LW N210. When we intercept the enemy signal, our interception should be a multipulse signal. Therefore, this paper generates a multipulse radar signal with 4 pulses. Because of the different distribution of noise, the signals between radar



pulses are not exactly the same. In order to obtain the TFI of the radar signal, the signal is CWD transformed. Different from SAR image [25] and high-resolution radar target image [26], TFI has high definition, which is good for signal recognition.

Different time-frequency analysis algorithms have different characteristics. Among them, Gabor transform localizes time and frequency at the same time, which can better describe the transient structure in the signal. Its time-frequency resolution is completely determined by the Gaussian window. The Wegener–Wiley distribution (WVD) is to distribute the energy of the signal in the time-frequency plane. It has a good time-frequency focus. Affected by the interference of the cross term, its various smoothing improvement methods can reduce the cross-term interference but reduces the time-frequency focus.

In order to improve the recognition rate of the signal, we need high-resolution images. The CWD has the characteristics of minimal cross-term interference and has high definition and resolution for different signals. The radar signal data set in this paper contains 10 classes of signals. There are 2880 TFIs for each class of signal. We add Gaussian white noise to the signal. In the SNR of  $-14 \sim 4$  dB, there are a total of 28800 samples, including 21,600 in the training set and 7,200 in the test set. Figure 5 is the TFI after the signal passes through the CWD.

According to Figure 5, it can be seen that different radar signals TFI have a large number of repeated similar regions, while the regions of distinguishing features are relatively small. Therefore, this paper designs a DFFE module, which can focus on extracting regional features with strong resolution and achieve the purpose of improving the signal recognition rate.

**3.2. Preprocessing and Experiment Setup.** In the process of data preprocessing, we downsample the image, and the resolution is fixed to  $224 \times 224$ . Then, the data is expanded, and the image is randomly flipped horizontally, vertically flipped randomly, and rotated  $90^\circ$  randomly. Data expansion increases the complexity of the image and improves the performance of the network. USRP N210 and USRP-LW N210 are hardware devices for signal generation. Among them, its ADC Sampling Rate is 100MS/s, DAC Sampling Rate is 400MS/s, and LO accuracy is 2.5 ppm.

In the experiment, we set some parameters. Among them, the bandwidth of the signal is 4MHZ, the batch size is 16, the initial learning rate is 0.001, the weight decay is  $5e-4$ , and the momentum is 0.9. The experiment is conducted for a total of 60 cycles, and the final results are the average of the last 10 cycles. The training and testing process of the network is implemented on the server. Among them, we use the PyTorch framework. Its operating system is Ubuntu 14.04.5 LTS, GPU is GeForce GTX TITAN X, and CUDA is CUDA 8.0.61.

**3.3. Experimental Results.** In this paper, we add noise to the signal to keep the SNR at  $-14 \sim 4$  dB. Then, we generate more realistic multipulse radar signals through GNU Radio

and USRP N210, USRP-LW N210. We use HRF-Net with different depths to identify multipulse radar signals. Figure 6 shows the experimental results.

According to Figure 6, when HRF-Nets have a SNR of  $-8$  dB, the network recognition results are all over 99%. When HRF-Nets are at a SNR of  $-14$  dB, noise interference has already had a great impact on the signal. However, the recognition result of the network still exceeds 97%, which shows that the HRF-Nets network has good robustness. The recognition rate of HRF-Net157 is slightly lower (within 1%) than that of the other two networks. It indicates that as the network deepens, its signal feature extraction ability has approached saturation, and the signal recognition rate has not improved significantly. HRF-Net217 and HRF-Net187 have more parameters by 32.46% and 14.93% than HRF-Net157, and more calculations by 30.32% and 15.21% than HRF-Net157. It can be seen from Figure 6 that the recognition result of HRF-Net217 is the best. Compared with HRF-Net157, the recognition rate is improved by no more than 1%, but the computational cost of the network has increased obviously. Through comprehensive analysis, we believe that HRF-Net157 has the highest cost performance. We also compare HRF-Net157 with other CNN networks, and Table 2 shows the recognition effect.

According to Table 2, the recognition performance of HRF-Nets is relatively high when the SNR is between  $-14$  and  $4$  dB. As the electromagnetic environment of the modern battlefield is becoming more and more complex, the signal interference is increasing. The recognition of radar signals under low SNR is of greater significance, and its recognition is more difficult. In the case of  $-14$  dB, the recognition result of HRF-Net157 is about 7% higher than that of VGGNets, and the calculation cost of VGGNets is too high, and the delay is long. Therefore, the VGGNets cannot be applied to the field of low delay radar electronic countermeasures.

In the recognition results of 10 multipulse radars, HRF-Net157 is about 2% higher than ResNet152, SNet152, and SKNet152. It is 2.418% higher than ResNet152, while the calculation and parameter amount of HRF-Net157 are relatively small. Although ResNet152 uses skip connections, it maintains information integrity. However, the TFI distinguishing feature area of the multipulse radar signal is small, and the repetitive area is large. When ResNet152 extracts image features, it performs the same processing on the image globally and has no specificity to the distinctive feature area. The DFFE module focuses on extracting high-resolution regional features of the image, improves the recognition effect of the network, and enhances generalization.

Table 3 shows the comparison result of HRF-Net157 with other methods, and it can be seen that the CLDNN network has a better recognition result at a SNR of over  $-8$  dB, reaching more than 90%, but when SNR is within the range of  $-14 \sim -8$  dB, its recognition rate is relatively poor. HRF-Net157 has a better recognition rate at  $-14$  dB. The comprehensive recognition rate of HRF-Net157 is still as high as 97.5%, indicating that HRF-Net157 can still fully extract image features under low SNR, so it has strong anti-

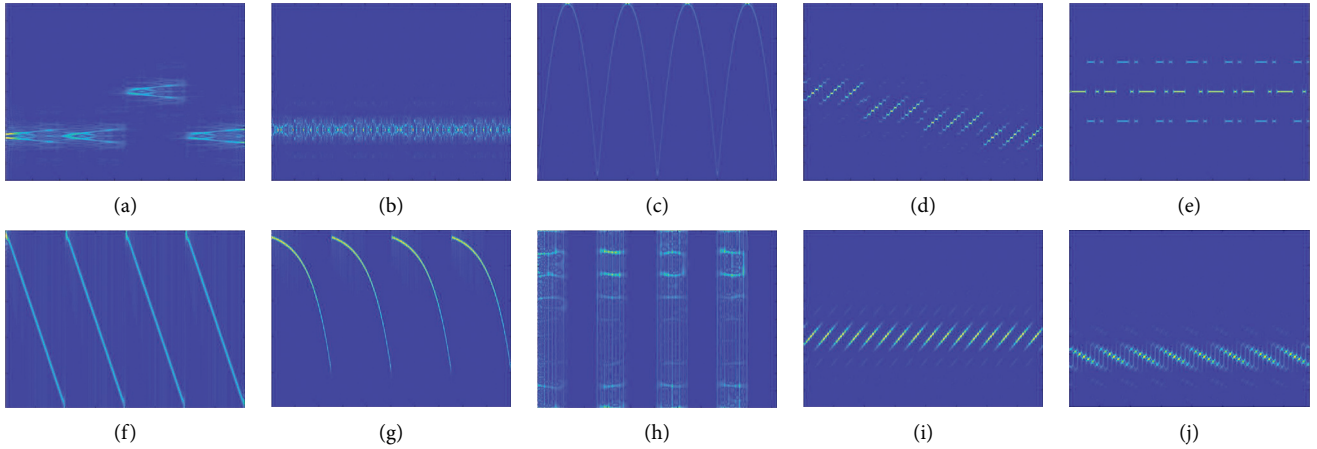


FIGURE 5: TFIs of 10 multipulse radar signals. (a) Barker, (b) Chaotic, (c) EQFM, (d) Frank, (e) FSK, (f) LFM, (g) LOFM, (h) OFDM, (i) P1, and (j) P2.

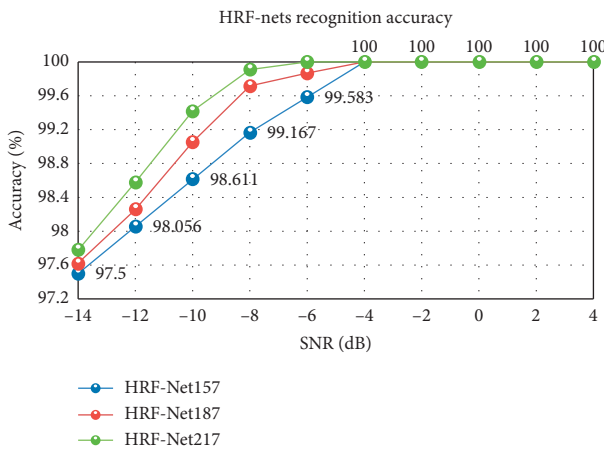


FIGURE 6: HRF-Nets recognition accuracy at different depths.

interference ability and good robustness. FCBF-AdaBoost adopts traditional feature selection and classifier design, which has a good recognition rate under the condition of less interference. But it is mostly for a certain class of image recognition. In a multitask and low SNR environment, its recognition rate is relatively poor. CNN-KCRDP, AlexNet, and I-CNN all combine deep learning to recognize images to a certain extent and can adaptively extract image features. Their recognition rates are not much different from HRF-Net157 when SNR is above  $-6$  dB. But, in the case of more serious interference, the signal features are submerged by noise. HRF-Net157 can extract more distinguishable features to a greater extent through the DFFE module. Therefore, when the interference is large, a better recognition effect can still be achieved.

Table 4 shows the recognition results of HRF-Nets for different signals. In the case of  $-14$  dB, the difference between the recognition results of the same class of radar signal for the three depths of HRF-Net is only about 2%. Deepening the network increases a lot of parameters and calculations, but the recognition effect is not significantly improved. Among the 10 multipulse radar signals, the recognition

results of Barker, Chaotic, Frank, OFDM, FSK, LFM, EQFM, and LOFM in HRF-Nets all reached more than 94%. The recognition effects of P1 and P2 are relatively poor, around 90%, and the fluctuation is large. We choose HRF-Net157 with the best cost performance and generate a confusion matrix at  $-14$  dB for further analysis.

According to Figure 7, it can be seen that the classes of errors identified in P1 are all P2. Among the 6 errors of P2, 5 are P1. According to Figure 5, it can be seen that the TFI of P1 and P2 has a certain degree of similarity. In the case of  $-14$  dB, the interference of noise has largely covered the features of the image, making the similarity of P1 and P2 increase, and further improved the difficulty of identification. However, the recognition rate of P1 and P2 still reaches about 90%. The HRF-Nets proposed in this paper can focus on extracting high-resolution image features for images with small distinguishing regions and obtain better recognition results. The comprehensive recognition rate of HRF-Net157 under  $-14$  dB reached 97.500%.

**3.4. Experiment Analysis.** In this paper, three depths of HRF-Net are proposed, namely, HRF-Net157, HRF-Net187, and HRF-Net217. According to the experimental results, the recognition rate of the signal is more than 99% when SNR is above  $-6$  dB. In the case of  $-14$  dB, the recognition results of the network also reached 97.500%. With the increase of network depth, the difference in recognition rate of HRF-Net157, HRF-Net187, and HRF-Net217 is only within 1%, but the computational cost has increased obviously. Taking into account comprehensive considerations, we believe that HRF-Net157 is the most cost-effective. In the process of comparing with other CNNs, it is found that the recognition rate of HRF-Net157 between  $-14$  dB and  $-6$  dB is higher than other CNNs, which is more obvious in the case of low SNR. When comparing with other methods, it is found that the recognition results of HRF-Net157 are better than other methods under the condition of  $-14$  dB. When compared with other methods, it is found that the signal recognition rate of HRF-Net157 is higher than other methods under the

TABLE 2: Recognition results of different networks (%).

SNR (dB)	ResNet152	SENet152	SKNet152	VGG13	VGG16	VGG19	HRF-Net157
-14	95.082	95.253	95.535	89.268	90.366	90.851	97.500
-12	96.374	96.862	97.134	91.423	93.514	93.735	98.056
-10	97.746	98.254	98.481	93.526	94.316	95.242	98.611
-8	98.356	98.426	98.768	95.628	96.211	97.522	99.167
-6	99.161	99.287	99.442	98.254	98.856	99.082	99.583
-4	100	100	100	99.142	99.627	99.855	100
-2	100	100	100	100	100	100	100
0	100	100	100	100	100	100	100
2	100	100	100	100	100	100	100
4	100	100	100	100	100	100	100

TABLE 3: Recognition results of other methods (%).

Methods	-14	-12	-10	-8	-6	-4	-2	0	2	4
CLDNN [7]	46	66	83	92	97	98	99	100	100	100
CNN-KCRDP [27]	—	—	88	94	97	98	100	100	100	100
AlexNet [9]	—	—	82	89	92	93	96	99	100	100
I-CNN [28]	—	—	55	80	96.10	—	100	100	100	100
FCBF-AdaBoost [3]	—	—	—	—	—	—	—	94.46	96.86	98.75
HRF-Net157	97.500	98.056	98.611	99.167	99.583	100	100	100	100	100

TABLE 4: HRF-Nets recognition results of different signals (-14 dB) (%).

Signal	HRF-Net157	HRF-Net187	HRF-Net217
Barker	98.611	100	100
Chaotic	100	100	100
EQFM	97.222	100	98.241
Frank	100	97.536	100
FSK	100	100	100
LFM	100	100	100
LOFM	94.444	96.538	96.524
OFDM	100	98.564	100
P1	87.500	89.536	89.422
P2	93.056	92.467	91.362

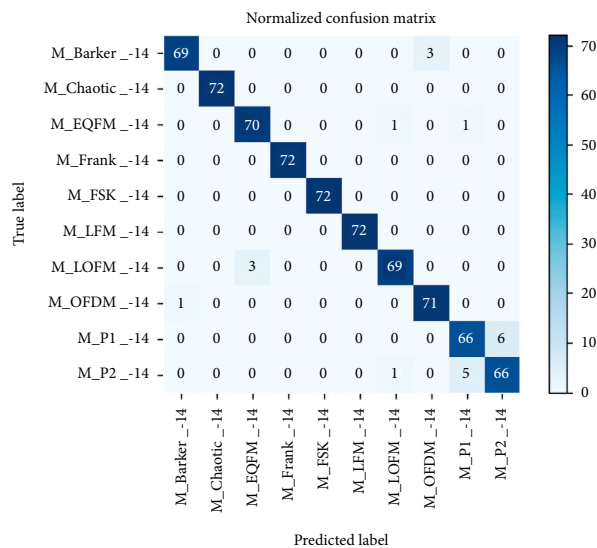


FIGURE 7: Confusion matrix of HRF-Net157(-14 dB).

condition of  $-14$  dB. It has better robustness. In the case of  $-14$  dB, HRF-Nets also have a good recognition effect on different classes of radar signals.

According to the TFI of the multipulse radar signal, we can see that the similarity area between different images is larger, and the distinguishing area is smaller. Therefore, when extracting image features, the importance of different areas of the image should be considered, focusing on extracting more distinguishable regional features. The network depth should be kept moderate. The image features cannot be fully extracted if the network is too shallow. The recognition rates of networks have not changed significantly when the network is too deep. There may also be network degradation problems, and the amount of parameters and calculations will increase significantly. The use of skip connections can maintain the integrity of image information. The classifier uses GAP followed by a single-layer full connection, which can greatly reduce the computational cost of the network. The DFFE module designed in this paper can perform distinguishing feature fusion extraction of images. First, Maxpool and Avgpool are used to compress the spatial dimensions to obtain feature maps with channel weights. Then, Maxpool and Avgpool are used to compress the channel dimensions, and the spatial weight feature map is obtained. Finally, the features of images are extracted by multiscale fusion through Conv1, Conv3, Conv5, and Conv7 to obtain high-resolution features and improve the signal recognition rate.

#### 4. Conclusions

In this paper, we use GNU Radio, USRP N210, and USRP-LW N210 to generate close-to-real multipulse radar signals and then perform CWD transformation on the echo signal to get TFI. Aiming at the features of the multipulse radar signal TFI, a DFFE module is designed, which can perform distinguishing fusion extraction of image features. Based on the DFFE module, we proposed three deep CNN structures, that is, HRF-Net157, HRF-Net187, and HRF-Net217. The nets can identify 10 classes of radar signals and have good generalization. Through comprehensive comparison, we believe that HRF-Net157 is the most cost-effective. In the case of a SNR of  $-14$  dB, there is still a recognition rate of 97.500%, with better robustness and lower computational cost. In radar systems that require low delay, HRF-Nets have certain advantages and can be further studied in the areas of radar interference recognition and radar radiation source recognition.

#### Data Availability

The data source is available from the author at hqzhang9013@163.com.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### Acknowledgments

This research was supported by the Natural Science Foundation of Hunan Province under Grant 2019JJ80105, Changsha Municipal Natural Science Foundation under Grant kq2014111, Changsha Science and Technology Project under Grant kq2004071, Hunan Graduate Scientific Research Innovation Project under Grant CX20200882, Shenzhen Science and Technology Program under Grant KQTD20190929172704911, and Scientific Research Project of Hunan Provincial Department of Education under Grant 20C1249.

#### References

- [1] W. Zhang, P. Ge, W. Jin, and J. Guo, "Radar signal recognition based on TPOT and LIME," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 4158–4163, Wuhan, China, July 2018.
- [2] K. Konopko, Y. P. Grishin, and D. Janczak, "Radar signal recognition based on time-frequency representations and multidimensional probability density function estimator," in *Proceedings of the 2015 Signal Processing Symposium (SPSymo)*, pp. 1–6, Debe, Poland, September 2015.
- [3] J. Guo, P. Ge, W. Jin, and W. Zhang, "Radar signal recognition based on FCBF and Adaboost algorithm," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 4185–4190, Wuhan, China, July 2018.
- [4] Q. Guo, P. Nan, and J. Wan, "Radar signal recognition based on ambiguity function features and cloud model similarity," in *Proceedings of the 2016 8th International Conference on Ultrawideband and Ultrashort Impulse Signals (UWBUSIS)*, pp. 128–134, Odessa, Ukraine, September 2016.
- [5] J. Li and Y. Ying, "Radar signal recognition algorithm based on entropy theory," in *Proceedings of the The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, pp. 718–723, Shanghai, China, November 2014.
- [6] M. Zhang, H. Wang, K. Zhou, and P. Cao, "Low probability of intercept radar signal recognition by staked autoencoder and SVM," in *Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Hangzhou, China, August 2018.
- [7] S. Wei, Q. Qu, H. Su, M. Wang, J. Shi, and X. Hao, "Intra-pulse modulation radar signal recognition based on CLDN network," *IET Radar, Sonar & Navigation*, vol. 14, no. 6, pp. 803–810, 2020.
- [8] Ji Li, H. Zhang, J. Ou, and W. Wang, "A radar signal recognition approach via IIF-net deep learning models," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8858588, 8 pages, 2020.
- [9] G. Limin, X. Chen, and C. Tao, "Radar signal modulation type recognition based on AlexNet model," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 49, no. 3, pp. 1000–1008, 2019, In Chinese.
- [10] Y. Xiao, W. Liu, and L. Gao, "Radar signal recognition based on transfer learning and feature fusion," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1563–1571, 2020.
- [11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [12] W. Wang, Y. Yang, X. Wang, W. Wang, and L. I. Ji, "The development of convolution neural network and its

- application in image classification: a survey,” *Optical Engineering*, vol. 58, no. 4, Article ID 040901, 2019.
- [13] W. Wang, Y. Yang, Ji Li, Y. Hu, Y. Luo, and X. Wang, “Woodland labeling in chenzhou, China, via deep learning approach,” *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 1393–1403, 2020.
  - [14] W. Wang, Y. Jiang, Y. Luo, Li Ji, X. Wang, and T. Zhang, “An advanced deep residual dense network (DRDN) approach for image super-resolution,” *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592–1601, 2019.
  - [15] W. Wang, H. Liu, Ji Li, H. Nie, and X. Wang, “Using CFW-Net deep learning models for X-ray images to detect COVID-19 patients,” *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 199–207, 2021.
  - [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1097–1105, Doha, Qatar, November 2012.
  - [17] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proceedings of the International Conference on Machine Learning*, pp. 807–814, Haifa, Israel, June 2010.
  - [18] G. E. Hinton, N. Srivastava, A. Krizhevsky et al., “Improving neural networks by preventing co-adaptation of feature detectors,” 2012, <http://arxiv.org/abs/1207.0580>.
  - [19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.
  - [20] K. He, X. Zhang, S. Ren et al., “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
  - [21] G. Huang, Z. Liu, L. Van Der Maaten et al., “densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Seattle, WA, USA, June 2017.
  - [22] A. G. Howard, M. Zhu, B. Chen et al., “Mobilenets: efficient convolutional neural networks for mobile vision applications,” 2017, <http://arxiv.org/abs/1704.04861>.
  - [23] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, “A novel image classification approach via dense-MobileNet models,” *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.
  - [24] M. Lin, Q. Chen, S. Yan et al., “Network in network,” in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.
  - [25] W. Wang, C. Zhang, J. Tian et al., “A SAR image targets recognition approach via novel SSF-net models,” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8859172, 9 pages, 2020.
  - [26] W. Wang, C. Zhang, J. Tian et al., “High resolution radar targets recognition via inception-based VGG (IVGG) networks,” *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8893419, 11 pages, 2020.
  - [27] D. Li, R. Yang, X. Li, and S. Zhu, “Radar signal modulation recognition based on deep joint learning,” *IEEE Access*, vol. 8, pp. 48515–48528, 2020.
  - [28] Z. Qu, X. Mao, and Z. Deng, “Radar signal intra-pulse modulation recognition based on convolutional neural network,” *IEEE Access*, vol. 6, pp. 43874–43884, 2018.

## Research Article

# Digital Twin-Enabled Online Battlefield Learning with Random Finite Sets

Peng Wang , Mei Yang, Jiancheng Zhu , Yong Peng, and Ge Li

*College of Systems Engineering, National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Jiancheng Zhu; [zhujiancheng@nudt.edu.cn](mailto:zhujiancheng@nudt.edu.cn)

Received 5 February 2021; Accepted 2 May 2021; Published 13 May 2021

Academic Editor: Qingshan Liu

Copyright © 2021 Peng Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The digital twin is becoming the most promising emerging technology in the field of unmanned combat and has the potential to innovate future combat styles. Online battlefield learning is one of the key technologies for supporting the successful application of digital twin in unmanned combat. Since there is an urgent need for effective algorithms for online learning the battlefield states in real time, a new random finite set- (RFS-) based algorithm is proposed in the presence of detection uncertainty including clutters, missed detection, and noises. The system architecture and operational mode for implementing the digital twin-enabled online battlefield learning are provided. The unmanned ground vehicle (UGV) is employed as the experimental subject for systematically describing the proposed algorithm. The system architecture for implementing the digital twin-enabled online battlefield learning is firstly given, and its operational mode is also described in detail. The RFS-based digital twin models including the battlefield state model, UGV motion model, and sensor model are designed. The Bayesian inference is adopted, and the probability hypothesis density (PHD) filter is modified to implement the online learning process. At last, a group of experiments are conducted to verify the performance and effectiveness of the proposed algorithm. The research work in this paper will provide a good demonstration of the application of digital twin in unmanned combat.

## 1. Introduction

The adoption of unmanned vehicles brings both great autonomy and new technical challenges to modern warfare. Unmanned vehicles such as unmanned ground vehicles (UGVs) hold great promise for future combat operations and have already been used in several recent military conflicts in Syria and Afghanistan [1–3]. UGVs are the vehicles that operate while in contact with the ground and without a human presence on board. How to feed back the effective information collected from the real battlefield to the simulation space and how to enable the benefits of future paradigms, such as the Cyber-Physical Systems (CPSs) and digital twin, are big challenges for unmanned combat [4–7]. In this paper, we employ the UGV as the experimental subject to specify our contributions in implementing digital twins in unmanned combat.

Due to the data separation between the real battlefield and its models, it is difficult to achieve the automatic flow of

information in a closed loop. Digital twin provides a new and effective way to solve this problem. It can enable the real-time bidirectional interoperability between the real world and virtual simulation space and is also an effective way to enable efficient real-time data sharing throughout the entire operational process including intelligent monitoring, prediction, digital representation, evaluation, decision support, and battlefield learning [8–10].

Battlefield refers to the environment constituted by all the objective factors in the battlespace except the combatants and weapons. All kinds of combat operations are inseparable from the specific battlefield. Battlefield has an important influence on the course and outcomes of combat operations. Combat entities can receive inputs from and provide outputs to the battlefield. The combat intention of the combat entity is realized through its interaction with the battlefield.

Battlefield learning means sensing the entities on the battlefield rapidly, understanding the current situation comprehensively, and predicting future status accurately

before decision-making [11]. Battlefield learning is important for predicting future situations and evaluating the operational effectiveness of different actions. Battlefield learning helps to improve the commander's understanding of the situation as a whole and form a basis for decision-making. It is also very important for the commanders' real-time monitoring and perception of the dynamic situation [12].

Based on the classical definition of battlefield learning, online battlefield learning is the process of perceiving an existing battlefield and anticipating how it may evolve in the future. It is useful for obtaining knowledge of the previously unknown battlefield while the real combat process is proceeding simultaneously [13]. Online battlefield learning is also extremely important for generating plans and online decision support for security patrol [14].

In the military simulation, computer-generated force (CGF) is the virtual combat force object which is created by a computer and can control or guide all or part of its action and behavior [15]. The core task of constructing CGF is to model the behavior of combat entities on the battlefield. Online battlefield learning is one of the key technologies of CGF and has a broad application prospect. CGF depends on the online learning battlefield to fuse the data generated by the sensors in the battlefield and generate the real-time battlefield states online.

In recent years, the digital twin has become a hot topic, as well as the representative intelligence in all fields from military to people's livelihood [16–19]. Digital twin emphasizes that the virtual object evolves in real time by receiving data from the physical object, thereby keeping consistent with the physical object throughout its entire simulation cycle [20, 21]. In a broad sense, the digital twin is a system composed of physical objects, simulation models, and the real-time dynamic interaction between them. It requires building the simulation models for real entities and simulating their behaviors [22]. It is regarded as the core link between the real and virtual spaces. With the help of various high-performance sensors and high-speed communication technologies, the digital twin can present and predict the actual situation of physical entities in near real time by integrating the data of physical entities. It enhances the ability of analysis and simulation and controls the physical entities through the virtual-real interactive interfaces and data fusion algorithms [23]. Key to enable digital twin in unmanned combat is understanding the evolving situations in the battlefield accurately and timely.

In this paper, we focus on learning the battlefield states that consist of significant environmental cues and the UGV states. In order to explore how to implement the digital twin-enabled online battlefield learning, we propose a random finite set- (RFS-) based algorithm which can support real-time interaction, as well as the deep integration and mutually beneficial symbiosis between the virtual and real battlefield. It is the necessary foundation for the successful application of the digital twin in unmanned combat. Our main contribution is designing and implementing a new online battlefield learning algorithm by using the RFS-based Bayesian theory and modifying the probability hypothesis density

(PHD) filter [24]. The most important value of the proposed algorithm is to break through the data boundary between the real and virtual battlefield and enable the application of digital twin in unmanned combat. This algorithm can eliminate information islands and realize the tight integration and equal interaction of real and virtual battlefield.

The rest of the paper is structured as follows. A literature review on the recent digital twin and random finite set (RFS) is given in Section 2. We present the system architecture and operational mode of the proposed online battlefield learning algorithm in Section 3. The RFS-based battlefield model, UGV motion model, and sensor model are introduced in Section 4. The design and implementation of the learning process are given in Section 5. Experimental results are detailed in Section 6, and conclusions are given in Section 7.

## 2. Related Works

The digital twin has important research and application value in every stage of online battlefield learning. In the design and demonstration stage, the digital twin can help to improve the evaluation capability of system performance by enabling the equal two-way interaction between the simulation system and the real system. Through the semiphysical simulation, digital twin enhances the ability to quickly locate the design defect, optimize system design, and test the practicability of an online battlefield learning algorithm in execution.

In order to apply the digital twin-enabled online battlefield learning in the operation stage, it is important to realize the bidirectional interaction between the simulation space and the real space. Tao gives the five-dimensional structure models of digital twin and presents six application principles [25, 26]. The digital twin is the best way to realize the interactive integration of real space and simulation space and is highly concerned by many academics and enterprises. Its most important breakthrough is that it is not only a mirror image of the physical world but also accepts real-time data from the physical world and in turn acts on the physical world in real time [22, 27]. Digital twin brings new development opportunities to the combat simulation area, because it can allow commanders to have a complete digital footprint of the battlefield from beginning to end [28, 29]. The real-time dynamic interaction between the virtual world and the physical world is the foundation of the digital twin, as well as the main challenge of modeling and simulation. Some researchers present a digital twin-driven manufacturing cyber-physical system for parallel controlling of the smart workshop. By using the decentralized digital twin models, they successfully connect cyberspace and physical space.

Online battlefield learning needs autonomy in the operation stage. A decentralized multiagent system is also a new approach for implementing online battlefield learning, such as blockchain and CGF. Some researchers have discussed how to use blockchain to overcome the cybersecurity barriers for achieving intelligence in Industry 4.0 and introduced eight cybersecurity issues in manufacturing systems. Some researchers have surveyed the ability of blockchain for overcoming the barriers and examined the

literature on the manufacturing system perspective and the product lifecycle management perspective. Ali et al. provided a survey of all aspects of multiagent systems, starting from definitions, features, applications, challenges, and communications to evaluation. They also gave a classification on multiagent system applications and challenges along with references for further studies [30].

RFS provides a novel unified probabilistic way for fusing real-time battlefield data [31]. The conventional battlefield learning algorithms usually depend on the vector-based data representation and fail to support the digital twin in real time. The vector-based representation requires the dimension and elements' order in each vector to be equal and fixed. It also needs necessary operations outside of the Bayesian recursion to ensure the consistency of the vectors. The determination of newly observed measurements and missed measurements is implementing through vector augmentation and truncation which are very computationally intensive and irreversible. In this paper, we propose employing the random set theory to overcome these disadvantages. The proposed RFS-based algorithm can overcome the limitations of conventional algorithms very well, because it takes into account a more realistic situation where the randomly varying number of targets and measurements, detection uncertainty, false alarms, and association uncertainty are all taken into consideration.

### 3. System Architecture

With the rapid development of emerging information technologies, such as artificial intelligence (AI), cloud computing, edge computing, digital twin, and Internet of Things (IoT), the combat style has also been undergoing profound changes. New information technologies have facilitated the birth, development, and application of unmanned combat. Just as it is shown in Figure 1, new information technologies provide more diverse data sources, more powerful computing power, and more efficient computing methods for the key activities of unmanned combat including description, diagnosis, prediction, and decision.

The operational mode of the digital twin-enhanced online battlefield learning consists of five elements, i.e., computing services, physical entities, simulation models, connected data, and the connection between them. As shown in Figure 2, digital twin enables the bidirectional real-time mapping and interaction between real battlefield and its simulation model. Simulation models of the real combat entities are employed to reflect and predict their behaviors in real space. On the other hand, through the RFS-based battlefield states generated by the online battlefield learning algorithm, the combat simulation systems could guide the military commanders to respond to situation changes and choose the optimal courses of action (COA). Digital twin realizes the closed-loop optimization in the entire process from observing, orienting, and deciding to act. The simulation aspect of digital twin means building digital models of weapons, soldiers, or battlefield and executing all the models in an integrated way. The RFS-based simulation models are

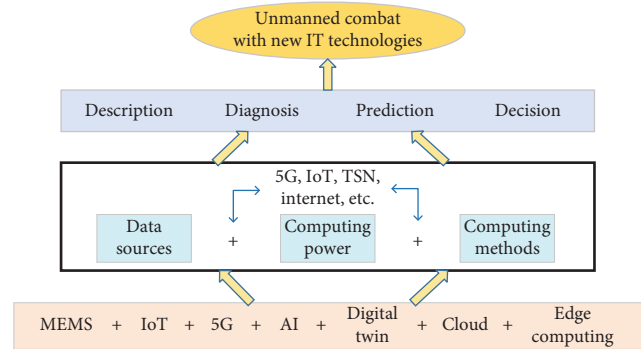


FIGURE 1: The relationship between new information technologies and unmanned combat.

executed in parallel with the real battlefield and provide useful knowledge to the commanders.

The battlefield considered in this paper consists of all the significant environmental cues and the states of UGVs. Since GPS and topographic map in actual combat are most likely be disabled, location and mapping for unmanned vehicles can only be obtained with the help of the equipped sensors. The RFS-based online battlefield learning algorithm plays a central role in the virtual space. It provides simulated battlefield information to the decision support system to train the deep learning network system. It can also generate real-time battlefield information to the unmanned combat simulation system and helps to evaluate the possible outputs of available COAs.

For combat simulation, the battlefield provides spatial-temporal constraints for all participating actors. The simulated combat objects are deployed and controlled in the virtual space. They learn the battlefield that consists of other combat objects and significant environmental cues by using the proposed algorithm. The combat simulation system in the virtual space is used as a decision-making aid tool that assists the commanders to evaluate all the available COAs. It is in charge of choosing the optimal COA. The proposed online battlefield learning algorithm aims at analyzing and understanding operational activity in the real space at a given time. It can help to make the right decision and predict the future situation. It is the key technology for enabling and implementing digital twin-enabled online battlefield learning in unmanned combat.

Corresponding to the operational mode, the system architecture of digital twin-enabled online battlefield learning in unmanned combat is shown in Figure 3. The runtime infrastructure (RTI) is adopted to provide the simulation services to support the interconnection and interoperation for the entities in the real space and the simulation models in the virtual space. This system architecture employs digital twin and RTI to support real-time interaction between the virtual and real battlefield. By this means, it can realize the deep integration and mutually beneficial symbiosis between the virtual and real battlefield. The proposed algorithm can synchronously learn the number and position of the significant environmental cues (or landmarks) in the battlefield that exist in the sensor's



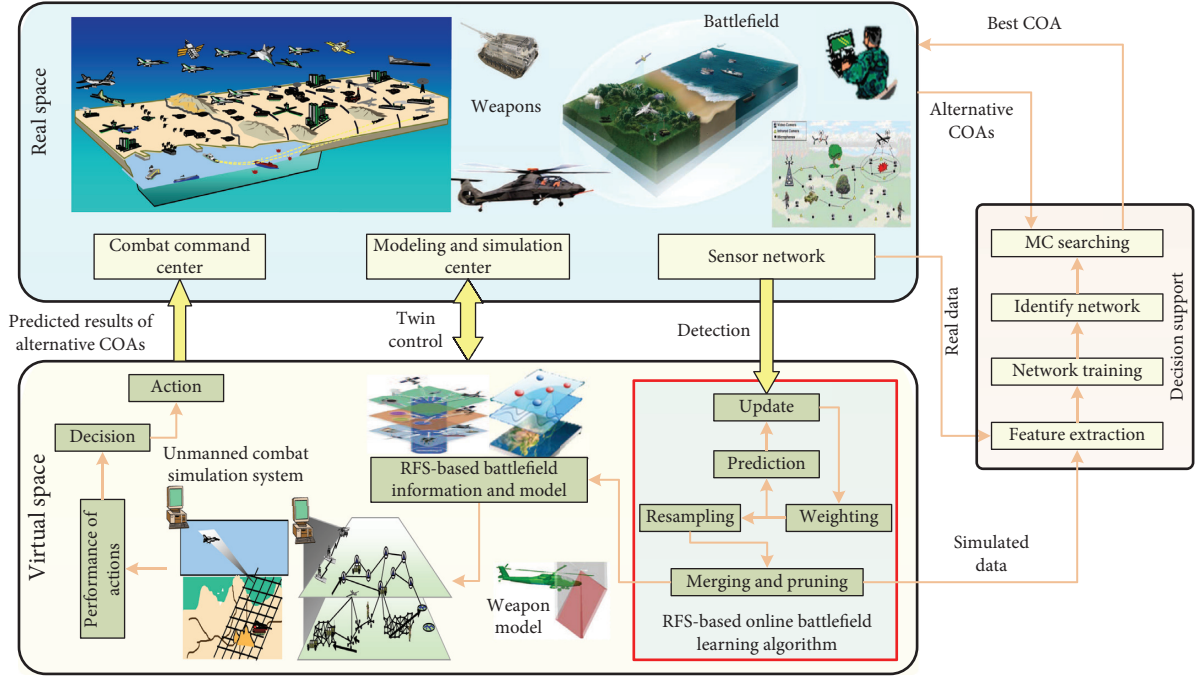


FIGURE 2: The operational mode of digital twin-enabled online battlefield learning in unmanned combat.

field of view (FoV). It also has the advantages of precise mapping, virtual-real interaction, stereoperception, intelligent intervention, and other characteristics.

#### 4. RFS-Based Simulation Models

The digital twin-enabled battlefield modeling consists of three aspects. The first one is modeling the battlefield states including cues (or landmarks). The second one is modeling the UGV movement. The third one is modeling the sensors equipped on the UGV. In order to overcome the data association uncertainty problem under high clutters and measurement noises, the RFS-based modeling method is employed to fully integrate data association uncertainty into battlefield learning. The key of the proposed algorithm is to represent the battlefield states by using RFS. The derivation of the simulation models depends on RFS. RFS is the theory proposed by Mahler for implementing RFS in engineering applications [32]. The RFS-based models are the twinning models that are executed in parallel with the real entities and provide new knowledge about the real battlefield [8, 27].

The vector-based representation of the battlefield has been demonstrated to have some mathematical consequences, such as the ordering of significant environmental cues, data association problems, and element management problems. In addition, for the dynamic random scene, how to quantify the errors of the learned results generated by the vector-based Bayesian inference is also a great challenge. The abovementioned problems are usually solved by augmenting or truncating vectors outside of the Bayesian inference process. This will lead to the problem that the Bayesian optimality can only be achieved on the subset of the battlefield that is defined in advance. In this section, we give the

RFS-based models which can solve these problems systematically.

The difficulty of RFS-based Bayesian inference is its computational complexity. To solve this problem, Mahler proposed the PHD (probability hypothesis density) filter. The PHD of the posterior probability density  $f_{k|k}(\mathbf{X}_k|\mathbf{Z}_k)$  is denoted by  $v_{k|k}(\mathbf{x}|\mathbf{Z}_k)$  and is a density function defined on the single object state  $\mathbf{x} \in \mathbf{X}_0$  as follows:

$$v_{k|k}(\mathbf{x}|\mathbf{Z}_k) = \int f_{k|k}(\{\mathbf{x}\} \cup \mathbf{X}_k|\mathbf{Z}_k) \delta \mathbf{X}_k. \quad (1)$$

Here,  $\mathbf{Z}_k$  denotes the RFS of detection received at time  $k$ , and  $\mathbf{X}_k$  denotes the RFS of states at time  $k$ . We use the abbreviation  $v_{k|k}(\mathbf{x}) = v_{k|k}(\mathbf{x}|\mathbf{Z}_k)$ . In point theory,  $v_{k|k}(\mathbf{x})$  is defined as the intensity density. It is not a probability density and represents the density of the expected number of points at  $\mathbf{x}$ . Given any subspace  $S$  of single object state space  $\mathbf{X}_0$ , the integral  $\int_S v_{k|k}(\mathbf{x}) d\mathbf{x}$  is the expected number of objects in  $S$ .

*4.1. RFS-Based Battlefield Representation.* We adopt the RFS-based battlefield representation; here,  $S$  denotes the RFS that represents the entire unknown battlefield. In addition, in order to assist in operational decision-making, we also relate the battlefield  $S$  to the UGV state  $X$ . RFS  $S_{k-1}$ , which is based on the UGV state  $X_{0:k-1} = [X_0, X_1, \dots, X_{k-1}]$  at time  $k-1$ , is used to denote the battlefield that has been explored.  $S_{k-1}$  is the RFS of the battlefield states which consists of significant environmental cues and is the intersection of the union of all FoVs and the entire battlefield state. Thus,  $S_{k-1}$  can be represented as follows:

$$S_{k-1} = S \cap \text{FoV}(X_{0:k-1}). \quad (2)$$

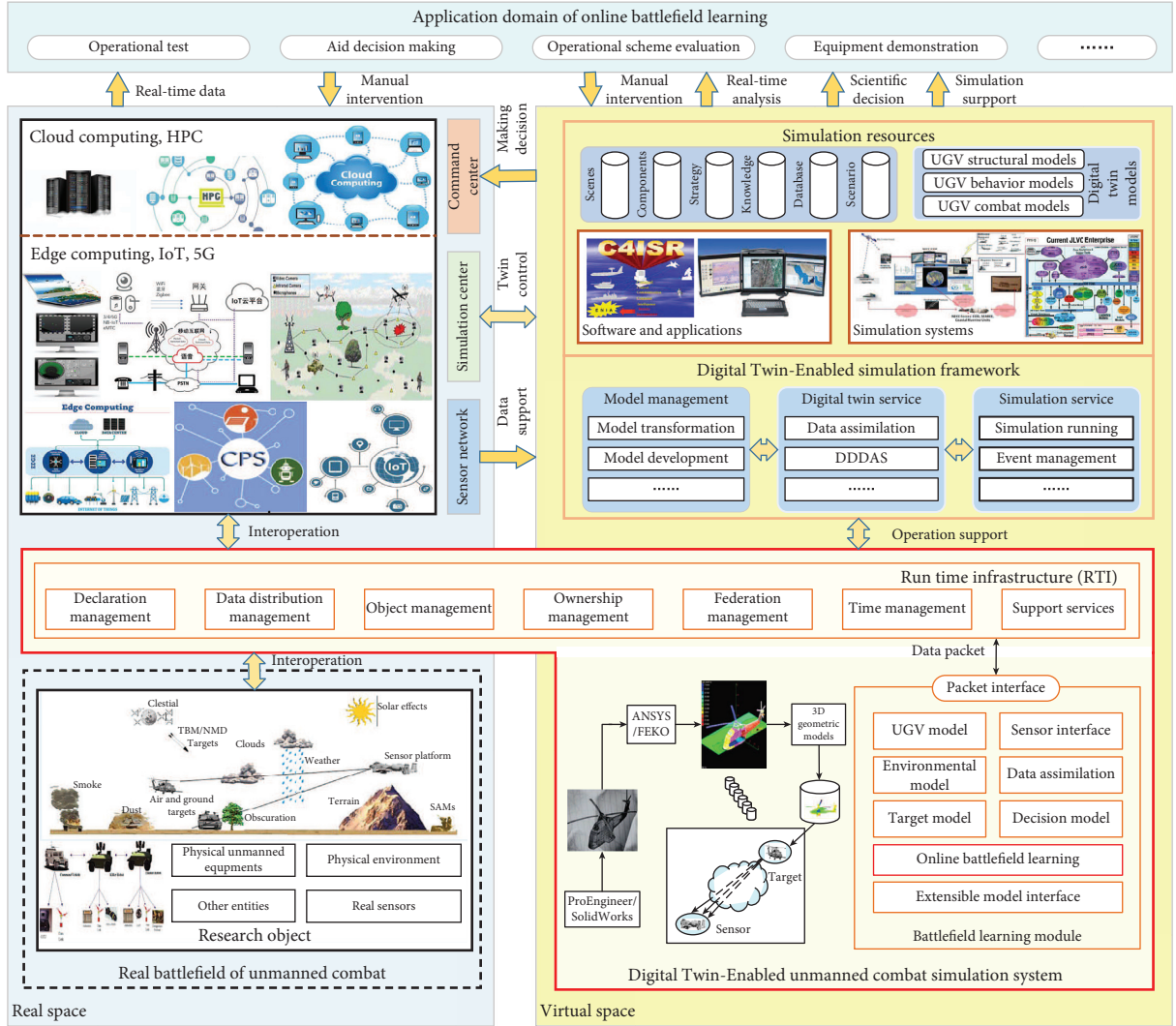


FIGURE 3: The system architecture of digital twin-enabled unmanned combat simulation.

Here,  $\text{FoV}(X_{0:k-1}) = \text{FoV}(X_0) \cup \text{FoV}(X_1) \cup \dots \cup \text{FoV}(X_{k-1})$ . FoV depends on the UGV states at time  $k-1$ . The learned battlefield at time  $k$  can be obtained based on  $S_{k-1}$  in the following way:

$$S_k = S_{k-1} \cup (\text{FoV}(X_k) \cap \bar{S}_{k-1}), \quad (3)$$

where  $\bar{S}_{k-1} = S - S_{k-1}$  represents the unexplored battlefield namely, the set of significant environmental cues that are not in  $S_{k-1}$ . RFS  $B_k(X_k)$  denotes the learned battlefield which has appeared in the FoV for the first time. Therefore, the battlefield transition process can be modeled as

$$f_S(S_k | S_{k-1}, X_k) = \sum_{W \in S_k} f_S(W | S_{k-1}) f_B(S_k - W | X_k), \quad (4)$$

where  $f_S(W | S_{k-1})$  denotes the state transition density of battlefield from time  $k-1$  to time  $k$ , and  $f_B(S_k - W | X_k)$  denotes the density of the RFS  $B(X_k)$ .

**4.2. RFS-Based UGV Motion Model.** The location of UGV can be represented by the state vector  $X = [x, y, \theta]^T$ . The UGV motion model characterizes the state transition between  $X_{k-1} = [x_{k-1}, y_{k-1}, \theta_{k-1}]^T$  and  $X_k = [x_k, y_k, \theta_k]^T$  after inputting the control command  $\mathbf{u}_{k-1}$ . In this paper, we adopt the following two-dimensional motion model with translational and rotational displacement:

$$\begin{aligned} X_k &= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k \\ &= g(X_{k-1}, \mathbf{u}_{k-1}) + \delta \\ &= g\left(\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k, \begin{bmatrix} \delta x \\ \delta y \\ \delta \theta \end{bmatrix}_k\right) + \omega_k, \quad \omega_k \sim (0, \mathbf{Q}). \end{aligned} \quad (5)$$

In this paper, the specific mathematical expression of  $g$  is employed as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \frac{t_{k-1}}{\gamma_{k-1}} (\cos(\theta_{k-1} + \gamma_{k-1}) - \cos(\theta_{k-1})) \\ y_{k-1} + \frac{t_{k-1}}{\gamma_{k-1}} (\sin(\theta_{k-1} + \gamma_{k-1}) - \sin(\theta_{k-1})) \\ \theta_{k-1} + \gamma_{k-1} \end{bmatrix} + \boldsymbol{\omega}_k. \quad (6)$$

Here,  $\boldsymbol{\omega}_k$  is used to represent the uncertainty and noise, and  $\mathbf{u}_{k-1} = [t_{k-1}, \gamma_{k-1}]^T$  is the control command that UGV received at time  $k-1$ .

**4.3. RFS-Based Sensor Model.** Given the current UGV state RFS  $X_k$  and the battlefield RFS  $S_k$ , the detection RFS can be described as follows:

$$Z_k = \bigcup_{s \in S_k} D_k(s, X_k) \cup C_k(X_k). \quad (7)$$

Here,  $D_k(s, X_k)$  denotes the detection RFS related to the significant environmental cue with state  $s$ , and  $C_k(X_k)$  denotes the clutters RFS, which is related to the UGV state  $X_k$ . Due to the uncertainty and randomness in the detection process, the number of elements in  $Z_k$  is random and may be different from the number of states in  $S_k$ .

The detection RFS  $D_k(s, X_k)$  generated by battlefield state  $s$  is modeled by Bernoulli RFS. Therefore, there are two forms of  $D_k(s, X_k)$ . The first one is  $D_k(s, X_k) = \emptyset$  and the probability is  $1 - p_D(s, X_k)$ . The other one is  $D_k(s, X_k) = z$  and the probability is  $p_D(s, X_k)g_k(z|s, X_k)$ .  $X_k$  denotes the UGV state at time  $k$ , and  $p_D(s, X_k)$  denotes the probability of generating detection from  $s$ .  $g_k(z|s, X_k)$  models the likelihood that  $s$  generates detection  $z$ . In this paper,  $p_D(s, X_k) = p_D$  if the significant environmental cue exists in the sensor's FoV, and  $p_D(s, X_k) = 0$ , otherwise.

Depending on  $X_k$  and  $S_k$ , the sensor's likelihood function for generating  $Z_k$  is represented as follows:

$$g_k(Z_k|X_k, S_k) = \sum_{W \subseteq Z_k} g_D(W|S_k, X_k)g_C(Z_k - W). \quad (8)$$

Here,  $g_D(W|S_k, X_k)$  is the likelihood function of generating detection RFS  $D_k$  for RFS  $S_k$ , and  $g_C(Z_k - W)$  is the probability density of the clutter RFS  $C_k$ .

In this paper, the range and bearing sensor is used. The detection generated by the two-dimensional environmental cues at location  $s$  can be modeled as follows:

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} r_k \\ b_k \end{bmatrix} \\ &= h(\mathbf{x}_k, s) + \mathbf{e}_k \\ &= \begin{bmatrix} \sqrt{(x_s - x)^2 + (y_s - y)^2} \\ \arctan\left(\frac{y_s - y}{x_s - x}\right) - \theta \end{bmatrix} + \mathbf{e}_k, \quad \mathbf{e}_k \sim (\mathbf{0}, \mathbf{R}). \end{aligned} \quad (9)$$

Here,  $\mathbf{z}_k = [r_k \ b_k]^T$  is the range and bearing detection,  $s = [x_s \ y_s]^T$  is the cue's position, and  $\mathbf{e}_k$  is the noise with covariance  $\mathbf{R}$ .

## 5. Learning Process and Its Implementation

In this section, we give the basic principles, design, and implementation of the proposed algorithm. The process of the proposed algorithm relies on sequentially propagating the joint posterior probability density of the RFS-based battlefield and the UGV state as detection arrives.

**5.1. RFS-Based Learning Process.** With the RFS-based battlefield modeling, the RFS-based Bayesian inference is used to jointly learn the environmental cues' locations and UGV state at every time step. The battlefield RFS can be characterized as follows:

$$p_{k|k}(S = \{s^1, s^2, \dots, s^{\hat{m}_k}\} | Z_{0:k}, X_{0:k}). \quad (10)$$

In this paper, we use  $p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0)$  to denote the predicted distribution of the battlefield state and  $p_{k|k}(X_{0:k}, S_k | Z_{0:k}, U_{0:k-1}, X_0)$  to denote the a posteriori distribution of the battlefield state. The knowledge of the battlefield can be propagated by the following prediction and update process:

- (i) Predict the battlefield state by using the previous battlefield states and input parameters:

$$\begin{aligned} &p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0) \\ &= \int f(X_{0:k}, S_k | X_{0:k-1}, S_{k-1}, U_{k-1}) \\ &\times p_{k-1|k-1}(X_{0:k-1}, S_{k-1} | Z_{0:k-1}, U_{0:k-2}, X_0) dX_{k-1}. \end{aligned} \quad (11)$$

- (ii) Update the battlefield state depending on the received detection RFS  $Z_k$ :

$$\begin{aligned}
& p_{k|k}(X_{0:k}, S_k | Z_{0:k}, U_{0:k-1}, X_0) \\
&= \frac{g_k(Z_k | S_k, X_k) p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0)}{\iint g_k(Z_k | S_k, X_k) p_{k|k-1}(X_{0:k}, S_k | Z_{0:k-1}, U_{0:k-1}, X_0) dX_k \delta S_k}.
\end{aligned} \quad (12)$$

Here,  $\delta$  implies set integration.

In this paper, the PHD filter is employed to implement the RFS-based Bayesian recursion [24, 33–35]. We modify and extend the Gaussian mixture-based PHD filter with a particle filter. The Gaussian mixture PHD filter is applied to learn the number and locations of the environmental cues, and the particle filter is applied to learn the UGV state at the same time. The computing process of online battlefield learning by modifying PHD filter is shown in Figure 4. The Bayesian recursion encapsulates the inherent uncertainty of the number of significant environmental cues that may be caused by detection uncertainty, clutters, UGV maneuvers, and the uncertainty related to detection noises.

The main challenge of online battlefield learning is how to learn the number and location of environmental cues while estimating the UGV state at the same time. In this paper, we partition the battlefield state into two kinds:  $s$  for environmental cues and  $X_k^{(i)}$  for UGV movement. We can analytically integrate out  $s$  provided that we know  $X_{0:k}^{(i)}$ . This means that even though we only have the sample sets  $X_{0:k}^{(i)}$ , we can also represent  $p(s|X_{0:k}^{(i)})$  successfully. Thus, each particle  $X_k^{(i)}$  represents a value for  $s$ . The advantage of this approach is that we can reduce the dimensionality of state space in which we are sampling and reduce the error of the learned battlefield.

Here, the Gaussian mixture PHD filter is applied to propagate each PHD that depends on the UGV state. The location of environmental cues in the battlefield is characterized by the Gaussian components of the mixture, and the number of cues in the battlefield is characterized by masses of all the Gaussian components. In this paper, the PHD at time  $k-1$  is characterized by the following  $N$  particles:

$$\{w_{k-1|k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1|k-1}^{(i)}(s|X_{0:k-1}^{(i)})\}_{i=1}^N, \quad (13)$$

where  $X_{0:k-1}^{(i)} = [X_0^{(i)}, X_1^{(i)}, X_2^{(i)}, \dots, X_{k-1}^{(i)}]$  is the  $i$ th hypothesized UGV state set,  $w_{k-1|k-1}^{(i)}$  denotes the weight, and  $v_{k-1|k-1}^{(i)}(s|X_{0:k-1}^{(i)})$  is the related PHD. The posterior distribution is approximated by the following set of weighted particles:

$$\{w_{k|k}^{(i)}, X_{0:k}^{(i)}, v_{k|k}^{(i)}(s|X_{0:k}^{(i)})\}_{i=1}^N. \quad (14)$$

In this paper,  $v_{k-1|k-1}^{(i)}(s|X_{k-1}^{(i)})$  is the prior PHD of the battlefield states for the  $i$ th particle related to the  $i$ th UGV trajectory.  $v_{k-1|k-1}^{(i)}(s|X_{k-1}^{(i)})$  can be represented by the following Gaussian mixture:

$$v_{k-1|k-1}^{(i)}(s|X_{k-1}^{(i)}) = \sum_{j=1}^{J_{k-1|k-1}^{(i)}} \eta_{k-1|k-1}^{(i,j)} N(s; \mu_{k-1|k-1}^{(i,j)}, P_{k-1|k-1}^{(i,j)}), \quad (15)$$

which consists of  $J_{k-1|k-1}^{(i)}$  Gaussian components. For  $j$ th Gaussian component,  $\eta_{k-1|k-1}^{(i,j)}$  is predicted weight,  $\mu_{k-1|k-1}^{(i,j)}$  is mean, and  $P_{k-1|k-1}^{(i,j)}$  is covariance. The PHD of the new environmental cue for the sampled state  $X_k^{(i)}$  at time  $k$  is represented by  $b(s|Z_{k-1}, X_k^{(i)})$ .  $b(s|Z_{k-1}, X_k^{(i)})$  is also a Gaussian mixture and can be represented as follows:

$$b(s|Z_{k-1}, X_k^{(i)}) = \sum_{j=1}^{J_{b,k}^{(i)}} \eta_{b,k}^{(i,j)} N(s; \mu_{b,k}^{(i,j)}, P_{b,k}^{(i,j)}), \quad (16)$$

where  $J_{b,k}^{(i)}$  is the number of the Gaussian components of the new PHD at time  $k$ , and  $\eta_{b,k}^{(i,j)}$ ,  $\mu_{b,k}^{(i,j)}$ , and  $P_{b,k}^{(i,j)}$  are the corresponding Gaussian parameters. The predicted PHD is therefore also a Gaussian mixture and can be represented as follows:

$$v_{k|k-1}(s|X_k^{(i)}) = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)} N(s; \mu_{k|k-1}^{(i,j)}, P_{k|k-1}^{(i,j)}). \quad (17)$$

Here,  $v_{k|k-1}(s|X_k^{(i)})$  is composed of  $J_{k|k-1}^{(i)} = J_{k-1|k-1}^{(i)} + J_{b,k}^{(i)}$  Gaussian components that represent the union of the prior PHD  $v_{k-1|k-1}(s|X_{k-1}^{(i)})$  and the PHD of new environmental cues. Since the detection function can also be represented by a Gaussian mixture, the posterior PHD  $v_{k|k}(s|X_k^{(i)})$  can be represented by a Gaussian mixture as follows:

$$\begin{aligned}
v_{k|k}(s|X_k^{(i)}) &= v_{k|k-1}(s|X_k^{(i)}) \\
&\cdot \left[ 1 - p_D(s|X_k^{(i)}) + \sum_{z \in Z_k} \sum_{j=1}^{J_{k|k-1}^{(i)}} v_{G,k}^{(i,j)}(z, s|X_k^{(i)}) \right].
\end{aligned} \quad (18)$$

The components of equation (18) are given as follows:

$$v_{G,k}^{(i,j)}(z, s|X_k^{(i)}) = \eta_{k|k}^{(i,j)}(z|X_k^{(i)}) N(s; \mu_{k|k}^{(i,j)}, P_{k|k}^{(i,j)}),$$

$$\eta_{k|k}^{(i,j)}(z|X_k^{(i)}) = \frac{p_D(s|X_k^{(i)}) \eta_{k|k-1}^{(i,j)} q^{(i,j)}(z, X_k^{(i)})}{c(z) + \sum_{l=1}^{J_{k|k-1}^{(i)}} p_D(s|X_k^{(i)}) \eta_{k|k-1}^{(i,l)} q^{(i,l)}(z, X_k^{(i)})}. \quad (19)$$

Here,  $q^{(i,j)}(z, X_k^{(i)}) = N(z; H_k \mu_{k|k}^{(i,j)}, S_{k|k}^{(i,j)})$ . The terms  $\mu_{k|k}^{(i,j)}$ ,  $P_{k|k}^{(i,j)}$ , and  $S_{k|k}^{(i,j)}$  can be got through standard Kalman filters; here, we adopt the unscented Kalman filter [36].

We assume that the number of clutters in  $C_k$  complies with the Poisson distribution, and the elements comply with uniform distribution over the battlefield state space. Then, the clutter PHD can be represented by  $c(z) = \lambda_c U(z)$ ; here,  $\lambda_c$  denotes the averaged number of clutters, and  $U(z)$  complies with a uniform distribution. In order to reduce the amount of calculation, we use pruning and merging methods to reduce the number of Gaussian components of the updated distribution [37].

The posterior UGV state  $p_k(X_{1:k})$  is sampled by  $\{\bar{w}_{k|k}^{(i)}, \bar{X}_k^{(i)}\}_{i=1}^N$  with

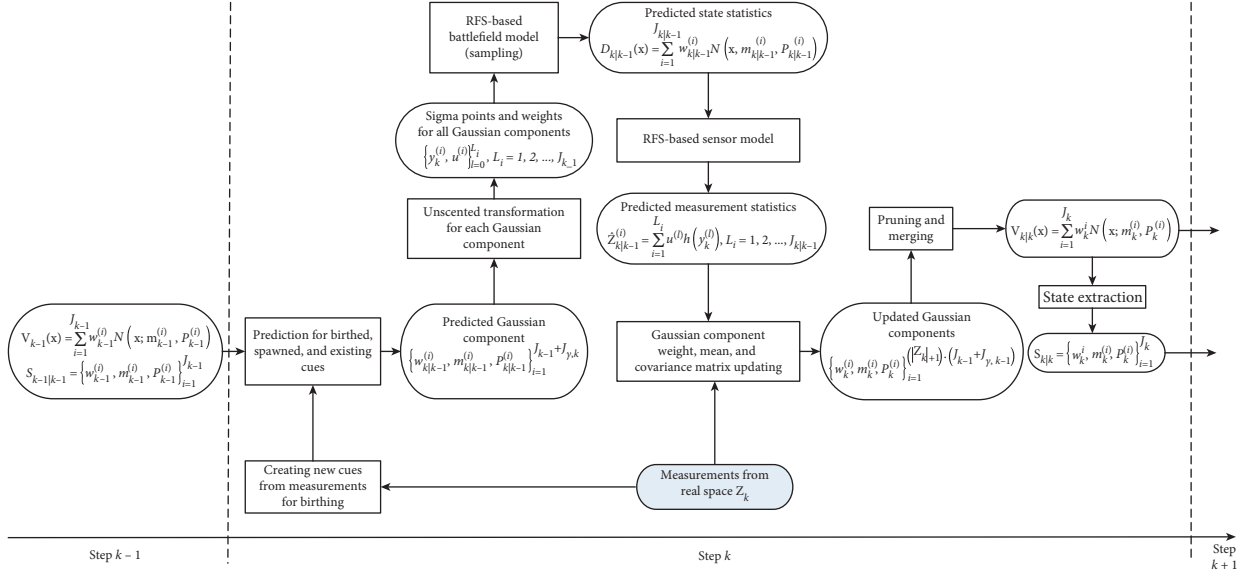


FIGURE 4: The online battlefield learning process based on Gaussian mixture-based PHD filter.

$$\tilde{X}_k^{(i)} \sim f\left(\tilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1}\right),$$

$$\tilde{w}_k^{(i)} = \frac{g_k\left(Z_k | Z_{0:k-1}, \tilde{X}_{0:k}^{(i)}\right) f\left(\tilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1}\right)}{f\left(\tilde{X}_k^{(i)} | X_{0:k-1}^{(i)}, U_{k-1}\right)} w_{k-1}^{(i)}. \quad (20)$$

The weights should be normalized as  $\sum_{i=1}^N \tilde{w}_k^{(i)} = 1$ . With the resampling step [24], we can get the resampled particles

$\{\tilde{w}_k^{(i)}, \tilde{X}_k^{(i)}\}_{i=1}^N$ . By choosing the UGV transition density as the proposal density, we get the weight as follows:

$$\tilde{w}_k^{(i)} = g_k\left(Z_k | Z_{0:k-1}, \tilde{X}_{0:k}^{(i)}\right) w_{k-1}^{(i)}. \quad (21)$$

By assuming that there is only one environmental cue  $\bar{s}$  in the battlefield, then, we can get

$$g_k\left(Z_k | Z_{0:k-1}, X_{0:k}\right) \approx \frac{1}{\Gamma} \left[ \left( (1 - p_D(\bar{s} | X_k)) \kappa_k^{Z_k} + p_D(\bar{s} | X_k) \sum_{z \in Z_k} \kappa_k^{Z_k \setminus \{z\}} g_k(z | \bar{s}, X_k) \right) v_{k|k-1}(\bar{s} | X_{0:k}) \right], \quad (22)$$

with

$$\Gamma = \exp\left(\hat{m}_{k|k-1} - \hat{m}_{k|k} + \int c_k(z) dz\right) v_{k|k}(\bar{s} | X_{0:k}). \quad (23)$$

$$\text{Here, } \hat{m}_{k|k-1} = \sum_{j=1}^{J_{k|k-1}} \eta_{k|k-1}^{(i,j)} \text{ and } \hat{m}_{k|k} = \sum_{j=1}^{J_{k|k}} \eta_{k|k}^{(i,j)}.$$

**5.2. Implementation.** According to the learning process given above, we give the concrete realization method of the proposed algorithm in this section. We use C++ to write the experimental program for this algorithm. The C++ library dependencies such as Eigen (version 3.0.0), Boost (version 1.5.3), and gtest are also used. In order to detail the implementation of the proposed algorithm, the flow diagram of the proposed algorithm is presented in Figure 5. The concrete steps are described in Algorithm 1.

The computational complexity of the proposed algorithm is  $O(m_k \cdot |Z_k| \cdot N)$  and is linear in the number of landmarks (in the FoV), as well as the number of detections and number of particles for the UGV state.

## 6. Experiments

In this section, a group of experiments are conducted to quantitatively verify the effectiveness and analyze the performance of the proposed algorithm. The virtual machine we used to run our experiments has 4 G of RAM and 6 3.40 GHz Intel CPUs and runs on Ubuntu 14.04 OS. The experimental data used to support the findings of this study are included within the article. The parameters used in this experiment are given in Section 6.1, and the models used in this experiment are given in Section 4.

**6.1. Experimental Setup.** As shown in Figure 6, the UGV patrols in a simulated two-dimensional space. The known ground truth (including the UGV states and locations of landmarks) is generated by the simulation models. The black dots represent the real locations of landmarks, and the black dashed line represents the real UGV states. The number of clutters complies with the Poisson distribution, and the

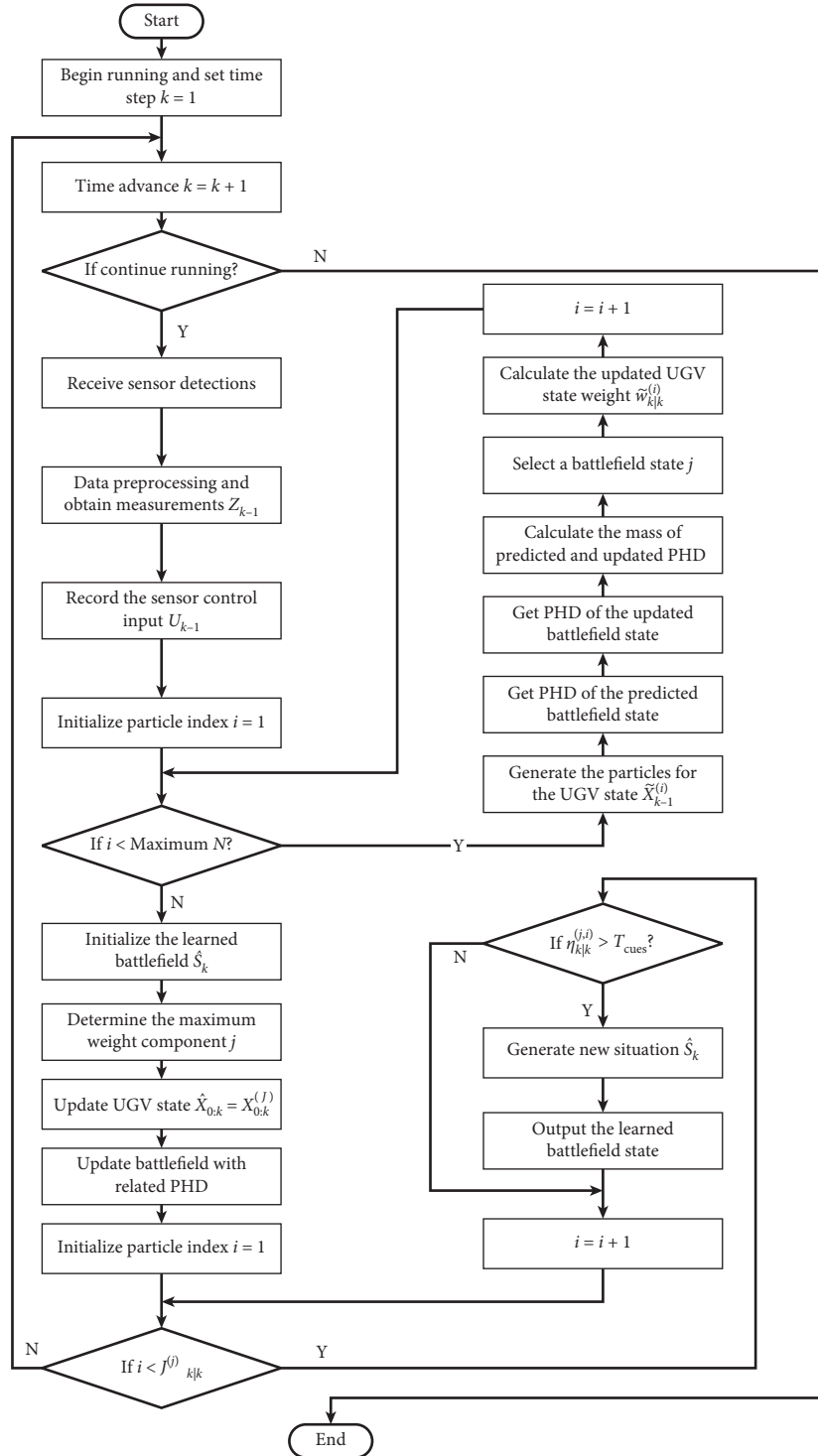


FIGURE 5: The flow diagram of the RFS-based online battlefield learning algorithm.

clutter PHD is uniformly distributed. Table 1 shows some important parameters for the simulation models to generate the ground truth.

The sensor used in this experiment is the range-bearing sensor that can detect landmarks with distances of 5 m to 30 m in any direction. The range measurement standard deviation (std) is 1 m and the bearing measurement std is

2 deg. The maximum FoV of the sensor used by the UGV is 10 m and 360 deg.

6.2. *Results and Analysis.* The experimental results are shown in Figure 6, the red dashed line represents the learned UGV states, and the red points represent the learned

**Input:**  $(\{w_{k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1}^{(i)}(s|Z_{k-1}, X_{k-1}^{(i)})\}_{i=1}^N, Z_{k-1}, U_{k-1})$   
**Output:**  $(\hat{X}_{0:k}, \hat{S}_k)$

- (1) for  $i = 1$  to  $N$  do
- (2) Generate the particles for the UGV state,  $\tilde{X}_{k-1}^{(i)} \sim f(\tilde{X}_{k-1}^{(i)}|X_{k-1}^{(i)}, U_{k-1})$ ;
- (3) Get predicted battlefield PHD through the predict step of PHD filter;
- (4) Get updated battlefield PHD through the update step of PHD filter;
- (5) Get predicted PHD mass  $\hat{m}_{k|k-1} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)}$ ;
- (6) Get updated PHD mass  $\hat{m}_{k|k} = \sum_{j=1}^{J_{k|k}^{(i)}} \eta_{k|k}^{(i,j)}$ ;
- (7) Select a given battlefield state  $j = \{i = 1, \dots, J_k^{(i)} | m^{(i,j)} = \bar{m}\}$ ;
- (8)  $a = (1 - p_D)c(z)^{|Z_k|} + p_D \eta_{k|k-1}^{(i,j)} \times \sum_{z \in Z_k} (c(z)^{|Z_k|-1})N(z; z_{k|k-1}, S_k^{(i,j)})$ ;
- (9)  $b = \exp^{\hat{m}_{k|k-1} - \hat{m}_{k|k} + \lambda_c} \eta_{k|k}^{(i,j)}$ ;
- (10) Get updated UGV state weight  $\tilde{w}_{k|k}^{(i)} = (a/b)\tilde{w}_{k|k-1}^{(i)}$ ;
- (11) end for
- (12) Initialize the learned battlefield state  $\hat{S}_k = \emptyset, I = \{1, \dots, N\}$ ;
- (13) Determine the maximum weight component  $j = \operatorname{argmax}_{j \in I} w_{k|k}^{(i)}$ ;
- (14) Update UGV state with  $\hat{X}_{0:k} = X_{0:k}^{(j)}$ ;
- (15) Update battlefield state according to the related PHD:
- (16) for  $i = 1$  to  $J_{k|k}^{(j)}$  do
- (17) if  $\eta_{k|k}^{(j,i)} > T_{\text{cues}}$ , here  $T_{\text{cues}}$  is the landmark existence threshold
- (18) Generate new battlefield state by  $\hat{S}_k = [\hat{S}_k \mu_{k|k}^{(j,i)}]$ ;
- (19) end if
- (20) end for
- (21) Return  $(\hat{X}_{0:k}, \hat{S}_k)$ .

ALGORITHM 1: RFS-based online battlefield learning algorithm.

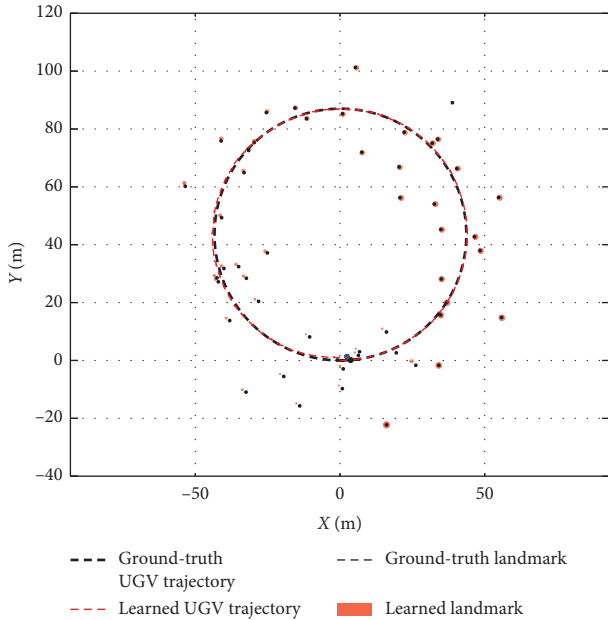


FIGURE 6: The real and learned battlefield states.

locations of landmarks. The collection of red dashed lines and points represents the battlefield states. The results successfully confirm that the proposed algorithm can learn the battlefield states by using sensor detection at runtime.

In order to quantitatively evaluate the performance of the proposed algorithm, we give the errors of the learned battlefield states in Figures 7 and 8. Figures 7(a) and 7(b) give the errors of the learned number and locations of landmarks. The errors of

locations are represented by optimal subpattern assignment (OSPA) distance [38]. We can find out that the performance of the proposed algorithm can satisfy the requirements of simulation and evaluation in unmanned combat.

Consider two sets  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$  and  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ , where  $m, n \in \mathbb{N}_0 = \{0, 1, \dots\}$ . Vectors  $\mathbf{x} \in \mathbf{X}$  and  $\mathbf{y} \in \mathbf{Y}$  are taking values from the battlefield state space. The OSPA metric is defined as a distance between sets  $\mathbf{X}$  and  $\mathbf{Y}$ . The OSPA distance of order  $1 \leq p \leq \infty$ , with the cut-off parameter  $c$ , is defined for  $m \leq n$  as follows:

$$d_{p,c}(\mathbf{X}, \mathbf{Y}) = \left[ \frac{1}{n} \left( \min_{\pi \in \Pi_n} \sum_{i=1}^m (d_c(\mathbf{x}_i, \mathbf{y}_{\pi(i)}))^p + (n-m) \cdot c^p \right) \right]^{(1/p)}, \quad (24)$$

where  $\Pi_n$  represents the permutations set of length  $m$  with elements taken from  $\{1, 2, \dots, n\}$ .

The errors of learned UGV states are shown in Figures 8(a) and 8(b). We can find out that the proposed algorithm can generate the learned UGV states with acceptable accuracy. But the errors are increased as time advance. This is caused by the cumulative errors of the UGV states.

In order to analyze how detection parameters affect the proposed algorithm, the averaged errors of the UGV states and landmarks are generated with different probabilities of detection  $p_D$  from 0.1 to 0.99 and clutter intensity  $\lambda_c$  from 0.0001 to 1. For each pair of parameters, 10 simulation runs were carried out. Here, cardinalized optimal linear assignment (COLA) is used to evaluate the errors for the learned landmarks. From Figures 9(a) and 9(b), we can find out that the errors of the learned UGV states increase as  $p_D$

TABLE 1: Partial experimental parameters.

Parameter	Velocity input std. (m/s)	Steering input std. (deg)	Detection probability $p_D$	Clutter rate $\lambda_c$	Particle number $N$	Landmark existence threshold $T_{cue}$
Value	2	2	0.90	0.0001	100	0.5

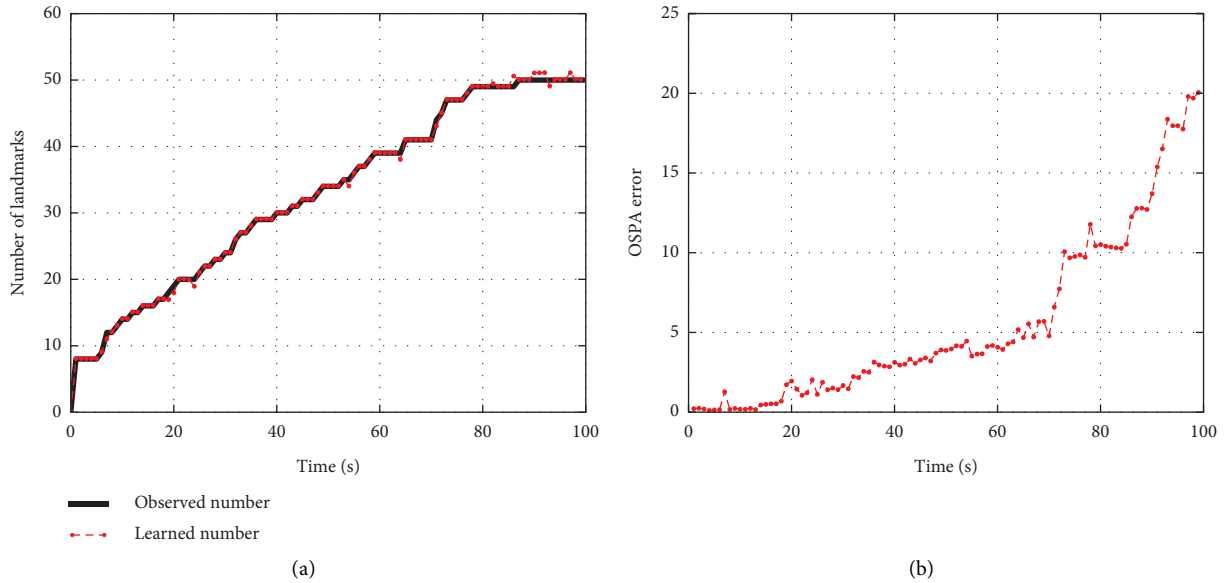


FIGURE 7: The errors of the learned number and locations of landmarks. (a) The learned and observed number of landmarks. (b) The OSPA errors of the learned landmarks.

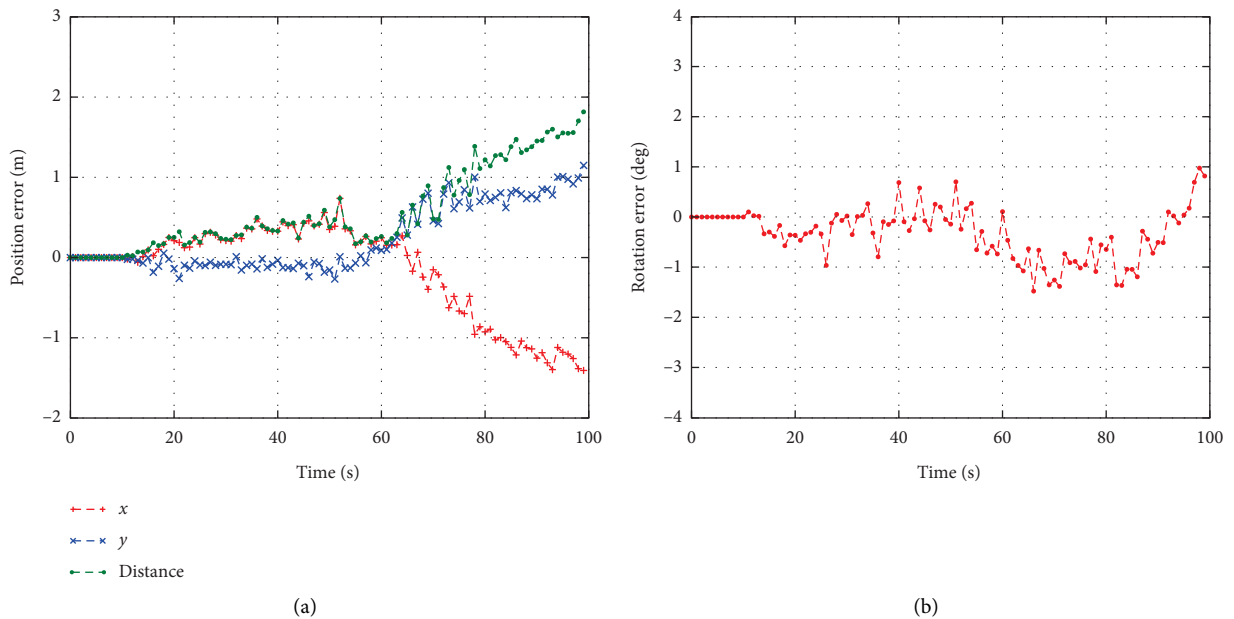
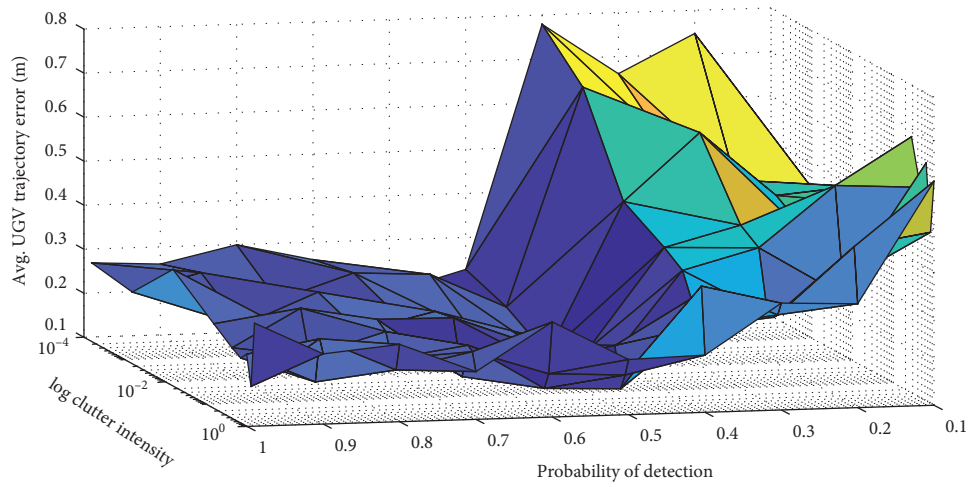


FIGURE 8: The errors of the learned UGV states. (a) Euclidean errors of the learned UGV states. (b) Orientation errors of the learned UGV states.

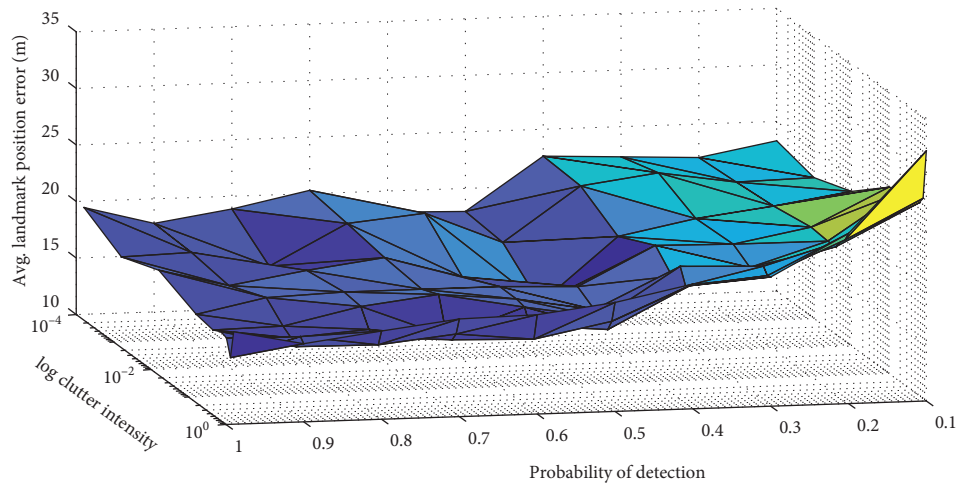
decreases. The errors of the learned landmarks only increase slightly as  $P_D$  decreases. The increase of  $\lambda_c$  will increase the errors of the learned locations of landmarks, but the effect on the errors of learned UGV states is quite small.

In order to apply the proposed algorithm in real unmanned combat applications, the time cost should be fully evaluated. As shown in Figure 10, we record 10 simulation runs for each pair of detection probability and clutter



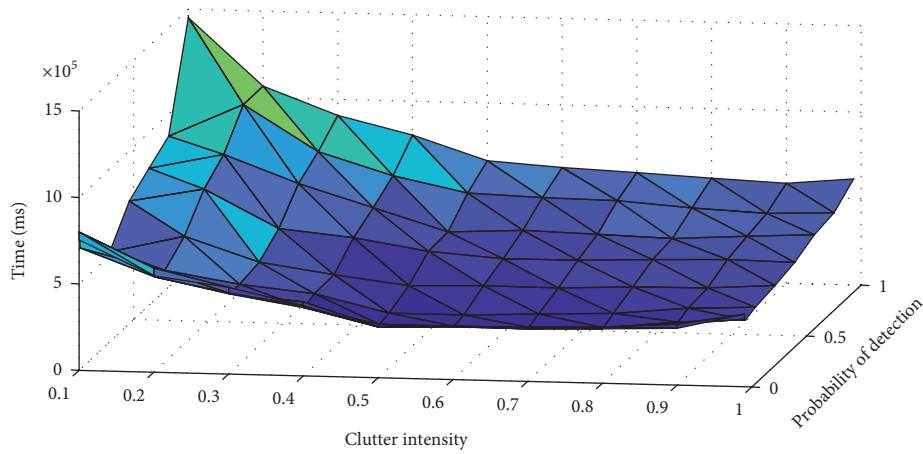


(a)



(b)

FIGURE 9: The performance for varying values of probability of detection and clutter intensity. (a) Averaged errors for the learned UGV states. (b) Averaged COLA errors for learned landmarks.



(a)

FIGURE 10: Continued.

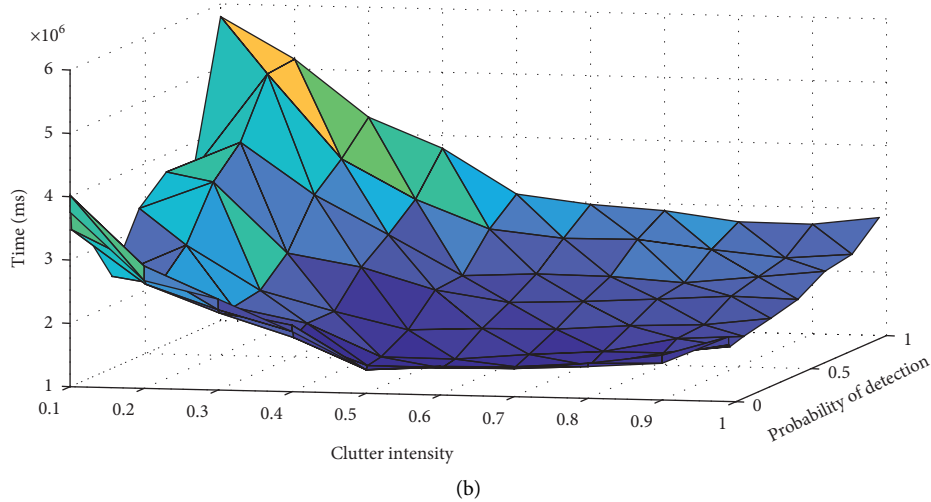


FIGURE 10: Time cost for varying values of detection probability and clutter intensity. (a) Averaged time cost of the algorithm. (b) Averaged time cost of CPU while the algorithm running.

intensity, and each simulation run consists of 1000 time steps. The averaged time costs of the proposed algorithm and CPU are shown in Figures 10(a) and 10(b). We can find that the increase of detection probability will increase the time cost, and the decrease of clutter intensity will also increase the time cost. The average time cost for each time step is about 500 ms, and it can satisfy many unmanned combat applications very well.

## 7. Conclusions

Digital twin technology enables real-time dynamic interaction between the real battlefield and the simulation system. Our main contribution is proposing a new online battlefield learning algorithm based on RFS to enable the application of the digital twin in unmanned combat. The digital twin has a broad application prospect in unmanned combat and greatly promotes the innovation of unmanned combat mode. Since the implementation of the digital twin in unmanned combat depends on battlefield understanding, an effective battlefield learning algorithm is quite important. By adopting the RFS-based representation of the battlefield, the proposed algorithm can overcome the limitations of the traditional vector-based representation. The performance of the proposed algorithm is verified by using two groups of experiments. This paper is the first attempt for applying the digital twin to the unmanned combat area and has practical significance for implementing the digital twin in many other areas.

## Abbreviations

CGF: Computer-generated force  
 COA: Courses of action  
 COLA: Cardinalized optimal linear assignment  
 EKF: Extended Kalman filter  
 FoV: Field of view  
 IoT: Internet of things  
 OSPA: Optimal subpattern assignment

PHD: Probability hypothesis density  
 RFS: Random finite set  
 UGV: Unmanned ground vehicle.

## Data Availability

The experimental data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

Peng Wang conceived, designed, and performed the simulations and wrote the manuscript. Jiancheng Zhu and Yong Peng provided the basic ideas and analyzed the experimental results. Mei Yang helped to perform the experiments. Ge Li and Yong Peng reviewed the manuscript.

## Acknowledgments

The authors would like to acknowledge the support of the Young Elite Scientists Sponsorship Program of China Association of Science and Technology.

## References

- [1] E. P. Blasch, R. Breton, P. Valin, and E. Bosse, "User information fusion decision making analysis with the C-OODA model," in *Proceedings of 14th International Conference on Information Fusion*, pp. 1–8, Chicago, IL, USA, July 2011.
- [2] N. Wang, X. Jin, and M. J. Er, "A multilayer path planner for a USV under complex marine environments," *Ocean Engineering*, vol. 184, no. 15, pp. 1–10, 2019.
- [3] J. Lee, S. Shin, M. Park, and C. Kim, "Agent-based simulation and its application to analyze combat effectiveness in network-centric warfare considering communication failure

- environments,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 2730671, 9 pages, 2018.
- [4] J. Luo, W. Zhang, W. Gao, Z. Liao, X. Ji, and X. Gu, “Opponent-aware planning with admissible privacy preserving for UGV security patrol under contested environment,” *Electronics*, vol. 9, no. 1, p. 5, 2020.
- [5] A. Bonci, M. Pirani, and S. Longhi, “Tiny cyber-physical systems for performance improvement in the factory of the future,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1598–1608, 2019.
- [6] C. Thule, K. Lausdahl, C. Gomes, G. Meisl, and P. G. Larsen, “Maestro: the INTO-CPS co-simulation framework,” *Simulation Modelling Practice and Theory*, vol. 92, pp. 45–61, 2019.
- [7] Z. B. Rivera, M. C. De Simone, and D. Guida, “Unmanned ground vehicle modelling in gazebo/ROS-based environments,” *Machines*, vol. 7, no. 2, p. 42, 2019.
- [8] A. Francisco, N. Mohammadi, and J. E. Taylor, “Smart city digital twin-enabled energy management: toward real-time urban building energy benchmarking,” *Journal of Management in Engineering*, vol. 36, no. 2, Article ID 04019045, 2020.
- [9] M. Abramovici, J. C. Göbel, and H. B. Dang, “Semantic data management for the development and continuous reconfiguration of smart products and systems,” *CIRP Annals*, vol. 65, no. 1, pp. 185–188, 2016.
- [10] M. Schlus and J. Rossmann, “From simulation to experimentable digital twins—simulation based development and operation of complex technical systems,” vol. 2016, pp. 273–278, in *Proceedings of 2nd IEEE International Symposium on Systems Engineering*, vol. 2016, , IEEE, Edinburgh, UK, October 2016.
- [11] H. R. Faghihi, M. A. Fazli, and J. Habibi, “Hybrid-learning approach toward situation recognition and handling,” 2019, <https://arxiv.org/abs/1906.09816>.
- [12] C. Mundutéguy and I. Ragot-Court, “A contribution to situation awareness analysis,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53, no. 6, pp. 687–702, 2011.
- [13] B. Handaoui, M. Alkalbani, T. Znati, and R. Ammar, “Unleashing the power of participatory iot with block chains for increased safety and situation awareness of smart cities,” *IEEE Network*, vol. 34, no. 2, pp. 202–209, 2020.
- [14] R. A. Bell, “Unmanned ground vehicles and EO-IR sensors for border patrol,” *Optics and Photonics in Global Homeland Security III*, International Society for Optics and Photonics, vol. 6540Bellingham, DC, USA, Article ID 65400B, 2007.
- [15] X. Xu, M. Yang, and G. Li, “Adaptive CGF commander behavior modeling through HTN guided Monte Carlo tree search,” *Journal of Systems Science and Systems Engineering*, vol. 27, no. 2, pp. 231–249, 2018.
- [16] S. Boschert and R. Rosen, “Digital twin—the simulation aspect,” in *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*, pp. 59–74, Springer International Publishing, Berlin, Germany, 2016.
- [17] J. Cheng, H. Zhang, F. Tao, and C.-F. Juang, “DT-II: digital twin enhanced industrial internet reference framework towards smart manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 62, no. 4, Article ID 101881, 2020.
- [18] J. Hochhalter, “Coupling damage-sensing particles to the digital twin concept,” 2018, <https://ntrs.nasa.gov/search.jsp?R=20140006408>.
- [19] S. Boschert, “Next generation digital twin,” in *Proceedings of TMCE 2018*, Las Palmas de Gran Canaria, Spain, May 2018.
- [20] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, “Digital twin-driven smart manufacturing: connotation, reference model, applications and research issues,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, Article ID 101837, 2020.
- [21] P. Wang, M. Yang, Y. Peng, J. Zhu, R. Ju, and Q. Yin, “Sensor control in anti-submarine warfare—a digital twin and random finite sets based approach,” *Entropy*, vol. 21, no. 8, p. 767, 2019.
- [22] P. M. Karve, Y. Guo, B. Kapusuzoglu, S. Mahadevan, and M. A. Haile, “Digital twin approach for damage-tolerant mission planning under uncertainty,” *Engineering Fracture Mechanics*, vol. 225, Article ID 106766, 2020.
- [23] L. Y. Yang, S. Y. Chen, X. Wang, J. Zhang, and C. H. Wang, “Digital twins and parallel systems: state of the art, comparisons and prospect,” *Acta Automatica Sinica*, vol. 45, no. 11, pp. 2001–2031, 2019.
- [24] R. P. S. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [25] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, “Digital twin-driven product design, manufacturing and service with big data,” *The International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 3563–3576, 2018.
- [26] F. Tao and Q. Qi, “Make more digital twins,” *Nature*, vol. 573, no. 7775, pp. 490–491, 2019.
- [27] Y. Zheng, S. Yang, and H. Cheng, “An application framework of digital twin and its case study,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 3, pp. 1141–1153, 2019.
- [28] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: state-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [29] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke, “Future modeling and simulation of CPS-based factories: an example from the automotive industry,” *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 97–102, 2016.
- [30] D. Ali, S. S. Kanhere, and K. Raja, “Multi-agent systems: a survey,” *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
- [31] D. A. Kai, L. I. Tiancheng, Y. Zhu, H. Fan, and Q. Fu, “Recent advances in multisensor multitarget tracking using random finite set,” *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 1, pp. 5–24, 2021.
- [32] P. Wang, G. Li, Y. Peng, and R. Ju, “Random finite set based parameter estimation algorithm for identifying stochastic systems,” *Entropy*, vol. 20, no. 8, 569 pages, 2018.
- [33] H. Sidenbladh, “Multi-target particle filtering for the probability hypothesis density,” in *Proceedings of the 6th International Conference of Information Fusion 2003*, p. 806, Cairns, Australia, July 2003.
- [34] T. Zajic and R. P. S. Mahler, “Particle-systems implementation of the PHD multitarget-tracking filter,” *Signal Processing, Sensor Fusion, and Target Recognition XII*, vol. 5096, no. 1, pp. 291–299, 2003.

- [35] C. Musso, N. Oudjane, and F. LeGland, "Improving regularised particle filters," *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, NY, USA, 2001.
- [36] P. Wang, G. Li, R. Ju, and Y. Peng, "Random finite set based data assimilation for dynamic data driven simulation of maritime pirate activity," *Mathematical Problems in Engineering*, vol. 2017, Article ID 5307219, 18 pages, 2017.
- [37] R. Mahler, "PHD filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.
- [38] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

## Research Article

# Hybrid Pyramid Convolutional Network for Multiscale Face Detection

Shaoqi Hou <sup>1</sup>, Dongdong Fang <sup>1</sup>, Yixi Pan <sup>2</sup>, Ye Li <sup>1</sup> and Guangqiang Yin <sup>3</sup>

<sup>1</sup>School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

<sup>2</sup>Glasgow College, University of Electronic Science and Technology of China, Chengdu, China

<sup>3</sup>School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China

Correspondence should be addressed to Guangqiang Yin; [yingq@uestc.edu.cn](mailto:yingq@uestc.edu.cn)

Received 14 March 2021; Revised 30 March 2021; Accepted 13 April 2021; Published 5 May 2021

Academic Editor: Nian Zhang

Copyright © 2021 Shaoqi Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Face detection remains a challenging problem due to the high variability of scale and occlusion despite the strong representational power of deep convolutional neural networks and their implicit robustness. To handle hard face detection under extreme circumstances especially tiny faces detection, in this paper, we proposed a multiscale Hybrid Pyramid Convolutional Network (HPCNet), which is a one-stage fully convolutional network. Our HPCNet consists of three newly presented modules: firstly, we designed the Hybrid Dilated Convolution (HDC) module to replace the fully connected layers in VGG16, which enlarges receptive field and reduces its loss of local information; secondly, we constructed the Hybrid Feature Pyramid (HFP) to combine semantic information from higher layers together with details from lower layers; and thirdly, to deal with the problem of occlusion and blurring effectively, we introduced Context Information Extractor (CIE) in HPCNet. In addition, we presented an improved Online Hard Example Mining (OHEM) strategy, which can enhance the average precision of face detection by balancing the number of positive and negative samples. Our method has achieved an accuracy of 0.933, 0.924, and 0.848 on the Easy, Medium, and Hard subset of WIDER FACE, respectively, which surpasses most of the advanced algorithms.

## 1. Introduction

The face is a key biometric characteristic of humans, thus making face detection the most widely used technology in the field of object detection, recognition, and tracking. The objective of face detection is to detect the existence of a face from a given image and return its size and location, and in practice, many face recognition and pedestrian matching systems have a higher demand for the speed and accuracy of detection.

Because images are taken under a variety of conditions, there is high variability in the scale, occlusion, lighting condition, and viewing angle between faces. To address these problems, the development of face detection techniques underwent three stages: template matching, AdaBoost, and deep learning.

In the early period, most of the face detection algorithms used template matching technology, that is, using a face

template image and comparing it with all regions in a given image to judge whether this region contains faces. One representative method was proposed by Rowley et al. [1, 2], who built a multilayer perceptron model using  $20 \times 20$  face and nonface images. Their methods handled the detection of images taken from not only the front [1] but also various angles [2]. Though this model performed well in precision, its detection speed was too slow due to a relatively complex design of classifier and dense sliding-window sampling. After that, machine learning algorithms were used in matching, including neural networks and quorum mode [3], support vector machine based on polemic kernel [4], Bayes classifier [5], and statistics model based on Hidden Markov Models (HMM) [6]. Despite a quite slow speed of detection, these algorithms did not overcome the disadvantages of naïve features.

In 2001, P. Viola and M. Jones published “Rapid Object Detection Using a Boosted Cascade of Simple Features” on

CVPR, which represented the coming of AdaBoost period [7]. This Viola–Jones method and Discriminatively Trained Part-Based Models (DPM) [8] were the most commonly known ones in that period. The principle of these algorithms is to build multiplied simple weak classifiers with the help of Haar [9], ACF [10], HOG [11], and other manual features, and then use them to construct a strong classifier possessing a high precision. However, because manual features were a few in number, poor in self-adaption, and stability, these algorithms generally failed to deal with complex conditions like different occlusion, lighting condition, or viewing angles and were usually slow in detection speed.

After that, benefiting from the fast development of deep learning, the above problems were effectively handled. Convolutional neural networks (ConvNet) [12] have a strong expression ability in learning nonlinear features. After its success in image classification, ConvNet was soon applied to face detection and showed a significantly higher precision compared with the previous AdaBoost framework [13]. Cascade CNN [14] can be recognized as a representative of the combination between the traditional method and deep learning. Similar to the algorithms in the Adaboost period, it also adopts a cascade structure, only with ConvNets as its cascade classifiers. Starting from Cascade CNN, a series of deep learning-based object detection algorithms were proposed. The most representative ones include one-stage algorithms with a fast speed, such as YOLO series [15–17], SSD series [18–20], and two-stage algorithms with high precision, such as Fast R-CNN series [21, 22], MTCNN [23], and R-FCN series [24]. These general detection models were also applied to face detection and performs well.

However, though most deep learning algorithms achieve success under different lighting conditions and viewing angles, their performances are still disappointing when confronted with complex circumstances like multiscale and occlusion. By comparing these methods, we found that one common drawback of them is to use a single or simple composite feature map rather than combine semantic information from higher layers together with details from lower layers effectively. For example, most two-stage ConvNets use several single feature maps and ignore information from higher or lower layers, while ScaleFace [25] combines features from only the lower layers. We assumed that this was the main reason for these methods to fail under extreme conditions.

In this paper, we proposed a multiscale face detection algorithm based on HFP structure and design a new face detection framework HPCNet. The main contributions of this paper are concluded as follows:

Targeting large-scale detection, we designed a HDC module, which can enlarge receptive field (RF) rapidly to acquire feature maps with a higher resolution. This mechanism is introduced from object segmentation to face detection for the first time.

Targeting small-scale detection, we presented a HFP structure as the core of our model, which combines semantic information from higher layers together with details from lower layers. Compared with Feature

Pyramid Network (FPN) [26], HFP processes features more carefully, with more convolution operations before feature fusion.

Targeting face occlusion and blurring, we introduced a CIE module here, which reduces the amount of computation and avoid feature confusion.

In addition, in the training stage, we presented an improved OHEM strategy in face of the imbalance between the number of positive and negative samples and introduced multiscale training to enhance the robustness of the model further. After running on the authoritative WIDER FACE [27], we found that our model showed a high precision of 0.933, 0.924, and 0.848 on three subsets Easy, Medium, and Hard, respectively. When running on GTX 1080Ti, the inference speed can achieve 44 Frames Per Second (FPS) with a higher resolution. After a series of comparative experiments, we proved our method to be reasonable.

The rest of the paper is organized as follows: Section 2 introduces some related works. Section 3 illustrates the proposed methods from point to total. Section 4 provides the experiments and Section 5 concludes the paper.

## 2. Related Works

*2.1. Dilated Convolution.* SSD [18], SFD [28], DSFD [29], and other algorithms add several convolution layers at the end of VGGNet [30] to address with large-scale objective or face. These added convolution layers help to process the information further, reduce the size of the feature map, and enlarge RF. Dilated Convolution has similar effects, only with the size of the feature map unchanged.

Specifically, dilated convolution is to make convolution kernel dilate. Assuming that the size of the kernel is  $f \times f$  and the dilation factor is  $d$ , then the size of the kernel after dilation  $f_d$  is

$$f_d = 1 + (f - 1)d. \quad (1)$$

The number of inserted pixels  $p_d$  is

$$p_d = \left\lfloor \frac{f_d}{2} \right\rfloor. \quad (2)$$

The dilation process of the kernel is shown in Figure 1, where the blank space left after dilation is filled by 0. Dilation convolution can enlarge kernel and RF rapidly without changing the size of the feature map, thus generating a feature map of a higher resolution. Dilated convolution is also commonly used in extracting structured and context information.

*2.2. Feature Pyramid.* To use feature maps of different scales for object detection is an effective method to handle the scale problem. There are mainly two ways to realize it: one is the featured image pyramid [31] and another one is using multiscale feature maps at the end of the network (as shown in Figure 2(a)). The former one has a large amount of computation due to repetitive calculation and has difficulty training network in an end-to-end way, while the latter one

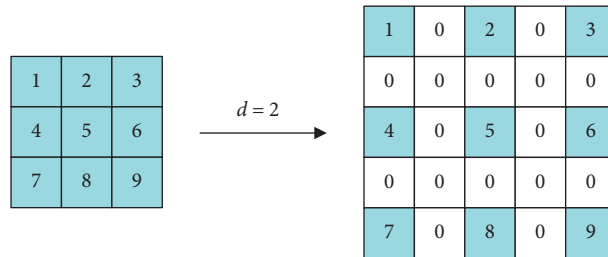


FIGURE 1: The dilation process of the convolution kernel.

avoids this successfully. Nevertheless, neither of these two methods takes advantages of feature maps from higher layers. As feature maps in lower layers contain no semantic information, its absence brings challenges to detection.

Our objective is to take full advantage of the pyramidal feature hierarchy embedded in ConvNets, which contains information from lower to higher layers and construct a feature pyramid combining information from lower to higher layers together.

FPN [26] offers a rather simple way to use feature maps. Its principle is to build a top-down architecture by introducing higher-level information to the current layer: first, the feature map from higher pyramid levels is upsampled by a factor of 2 (using nearest neighbor upsampling for simplicity); then, it undergoes a  $1 \times 1$  convolutional layer to reduce channel dimensions; finally, the upsampled map is merged with the current map (which undergoes a  $1 \times 1$  convolutional layer) by elementwise addition. The detailed process of merging is shown in Figure 3.

**2.3. Context Information.** When humans search for faces, they take not only faces but also hats, clothes, surroundings, and other information. Context information is exactly a simulation of this behavior. When it is difficult to judge whether the candidate proposal contains faces, we can use the information around the proposal as a supplement, which is an effective way to handle occlusion and blurring.

Based on the experience, CMS-RCNN [32] combined face and body information together for face detection. The spatial relationship between face and body is described as follows:

$$\begin{aligned}
 t_x &= \frac{x_b - x_f}{w_f}, \\
 t_y &= \frac{y_b - y_f}{h_f}, \\
 t_w &= \log \frac{w_b}{w_f}, \\
 t_h &= \log \frac{h_b}{h_f},
 \end{aligned} \tag{3}$$

where  $f$  and  $b$  represent face and body, respectively;  $t$  is a fixed value;  $x$ ,  $y$ ,  $w$ , and  $h$  represent the center coordinate, width, and height of candidate proposal. CMS-RCNN

substitutes the coordinate of extracted face candidate into equation (3) to acquire body candidate and then maps face and body candidate onto feature maps. After undergoing pooling layer, convolutional layer, and two fully connected layers, they are joint together for bias regression and classification of coordinates. The way that CMS-RCNN acquires context information can be easily combined with the two-stage objective detection algorithm, which is credited to the RoI pooling layer. Though the CIE in CMS-RCNN has positive effects on the detection result, the assumption it contains is too strong to be accurate and it is difficult to combine with one-stage detection algorithms.

**2.4. OHEM.** OHEM [33] is a completely online hard sample mining algorithm, which samples according to the non-uniform and nonstationary distribution depending on sample classification loss, and makes simple changes to the stochastic gradient descent. For each detection task, OHEM chooses  $N$  samples with a higher loss from thousands of proposals or anchors in one or two images. Though only using a part of proposals or anchors, its backward propagation is still effective and robust. The reason why OHEM does not use all the samples is that simple samples contribute little to the loss. In addition, when there are too many negative samples, the dataset is filled with simple samples, which altogether have a huge effect on loss, with no help to classification. The same with SVM, it is the hard samples that contribute truly to classification. Compared with hard sample mining, OHEM does not need to construct a dataset or train model; while compared with stochastic gradient descent, OHEM takes advantage of hard samples that make a contribution to classification loss and thus avoid useless computation.

### 3. Method

In this section, we will introduce each proposed module and give a comprehensive description of the overall framework of HPCNet.

#### 3.1. Components in HPCNet

**3.1.1. HDC Module.** Instead of adding several convolution layers at the end of a basic convolution network like SSD [18], SFD [28], and DSFD [29], we introduce the concept of dilated convolution to handle large-scale faces, which is a new trail.

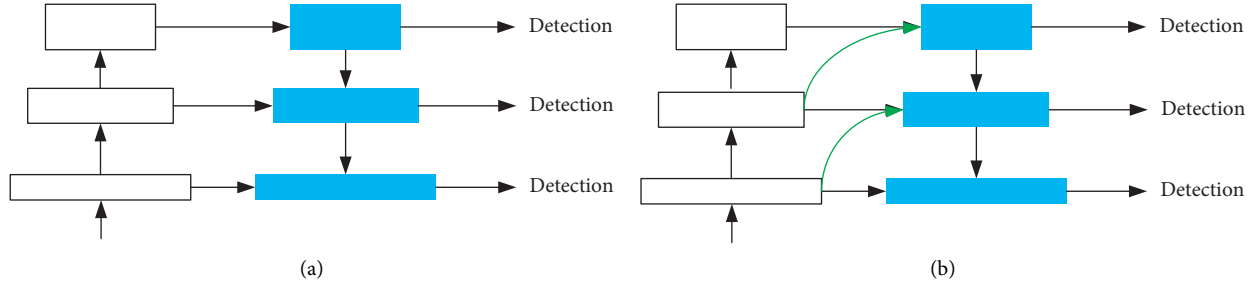


FIGURE 2: The simplified structure of the feature pyramid before and after improvement. (a) Feature Pyramid Network (FPN). (b) Hybrid Feature Pyramid (HFP).

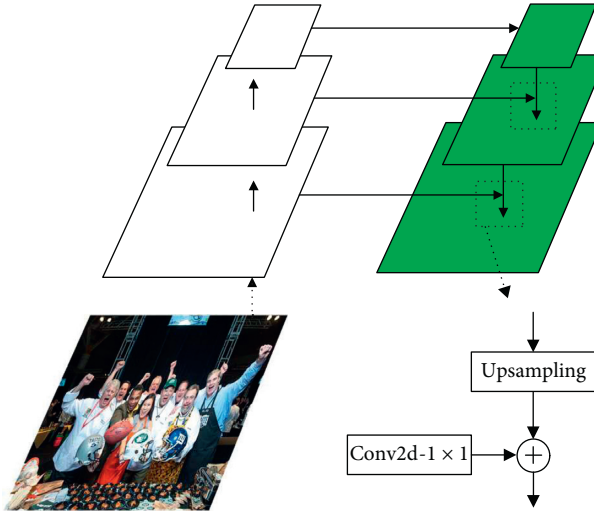


FIGURE 3: The structure of FPN.

There exist some drawbacks in the common dilated convolution. Assume that there is a pixel  $v$  in the  $l$ th layer, and the  $f_d \times f_d$  area that contributes to  $v$  is in the  $l-1$ th layer around the location of  $v$ . Because the dilated kernel introduced several 0, the actual contribution area is still  $f \times f$ . As the dilation factor increases, the contribution area in  $l-1$ th layer enlarges rapidly, while the real contribution area stays the same. Therefore, the local feature information gradually gets lost due to 0 values, and the correlation of information contributing to  $v$  decreased consistently. When several dilated convolution layers are connected in series, this effect will be exacerbated continuously.

Assume that there are three dilated convolution layers forming a structure  $s$ , where the kernel is  $3 \times 3$ , dilation factor is 2, and the sliding stride is 1. With structure  $s$  replacing the  $l$ th layer, the RF area that truly contributes to  $v$  in the  $l-1$ th layer is shown in Figures 4(a)–4(c). The number in the blue grid represents its contribution value and the white grid has no contribution. The value in Figure 4 is calculated under the assumption that the values of the kernel and the  $l-1$ th feature map are all 1.

To make use of the advantages of dilated convolution, as well as avoiding local information loss and correlation reduction, we designed the HDC module. HDC only contains three dilated convolution layers, of which the kernel size is  $3 \times 3$ , dilation factor is 1, 2, and 3, respectively, and sliding

stride is 1. With HDC replacing the  $l$ th layer, the RF area that truly contributes to  $v$  in the  $l-1$ th layer is shown in Figures 5(a)–5(c). It is clear from Figure 5 that, in every stage, all the grids in RF area contribute to  $v$ , and the weights of which increase as getting closer to the location of  $v$ . This structure is obviously reasonable.

**3.1.2. HFP Module.** Though FPN [26] introduced semantic information from the higher layer into the current feature map, there still exist three problems:

FPN generates composite feature map by elementwise addition, which lacks self-adaption and can easily cause feature confusion

FPN ignores information from lower layers when constructing a feature map, which results in a lack of details and location information, thus bringing a challenge to locating and detecting small-scale objects

The composite feature map obtained is used both as high-level semantic information and for detection, which is not a reasonable way as it undertakes too many tasks

Our HFP (as shown in Figure 2(b)) is an improvement of FPN targeting at the above three problems. A summary of its process is as follows: first, it upsamples feature maps from higher layers (using bilinear interpolation) and reduces their channel dimensions to generate composite feature maps by merging with current ones; then the composite feature map is further processed to obtain truly useful semantic information by reducing channel dimensions; after that, similarly, downsampling and channel dimensions reduction are applied to feature maps from lower layers to acquire hybrid feature maps by stitching with composite ones; finally, the hybrid feature maps are used for detection after channel changes and information fusion.

To be more specific, the details in HFP (as shown in Figure 6) are as follows: for high-level feature maps, we use  $1 \times 1$  convolution layer for channel dimension reduction, as the  $1 \times 1$  kernel will not change RF and is more suitable for semantic learning. For composite feature maps, we use  $3 \times 3$  convolution layer when used as high-level semantic information, as the  $3 \times 3$  kernel can avoid feature confusion resulting from upsampling and downsampling. For low-level feature maps, we use  $3 \times 3$  convolution layer with a stride of 2 for downsampling in order to save detailed



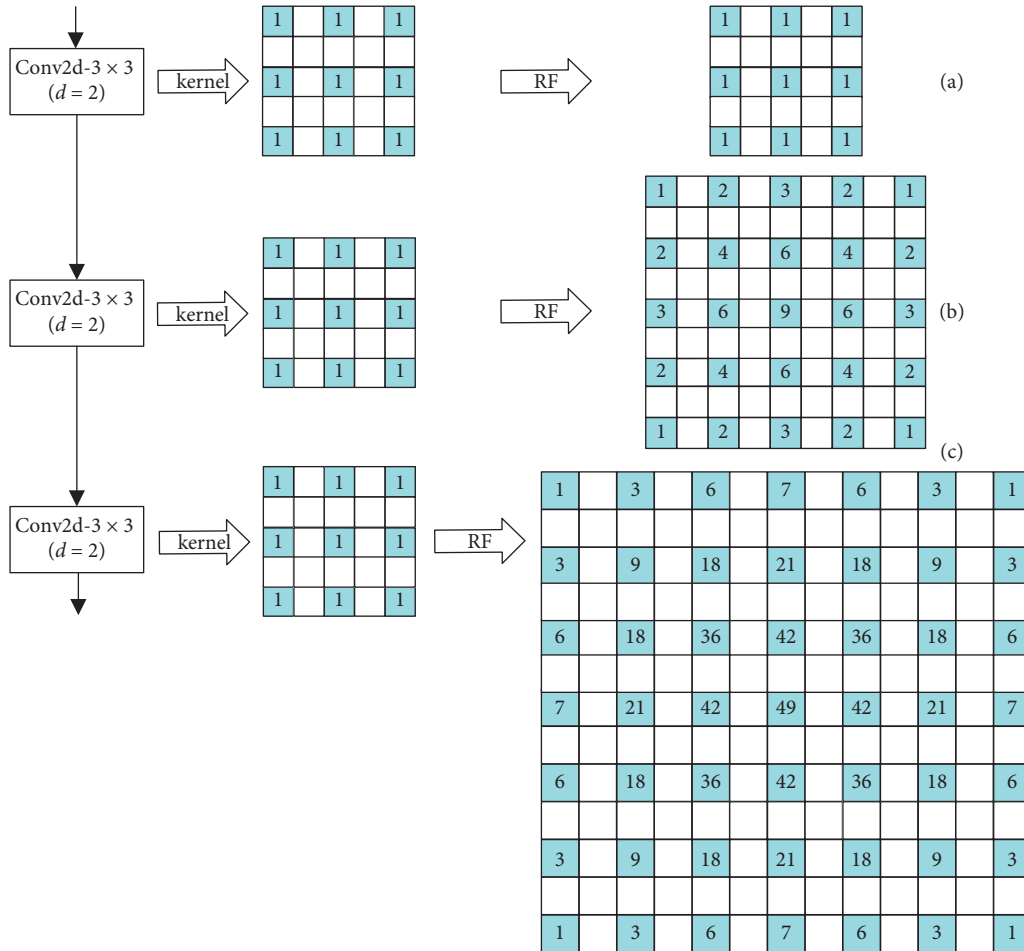


FIGURE 4: An illustration of the area that truly contributes to  $v$  in three series-connected layers with the same factor of 2.

information; then the feature maps undergo another  $3 \times 3$  convolution layer for channel dimension reduction to extract truly needed details. For hybrid feature maps, we use  $3 \times 3$  convolution layer for channel changes and information fusion as well.

Our HFP is different from FPN in several aspects:

In HFP, composite feature map is generated by channel joint, while FPN uses elementwise addition instead.

The processing procedure of feature maps for detection is different. In FPN, composite feature maps are used in detection directly, while our HFP processes composite ones further and combines them with low-level ones before detection.

HFP handles feature maps in a more careful way and adopts a series of dimension operations to acquire effective information.

**3.1.3. CIE Module.** Despite enlarging the window around the candidate proposals, a bigger kernel is a better choice for a one-stage object detection algorithm to obtain information around faces.

SSH [34] adopts this strategy by applying simply two bigger kernels to extract context information. However, a

bigger kernel usually leads to a bigger amount of computation, which can be replaced by several smaller ones connected in series. Inspired by this idea and SHH, we proposed our CIE which only contains convolution layers with  $3 \times 3$  kernels. To reduce the amount of computation further and prevent the correlation of context from decreasing, we adopt a method to share some convolution layers. The detailed structure is shown in Figure 7.

**3.1.4. Improvements of OHEM.** Though OHEM [33] is robust and highly efficient, it only considers hard samples without taking the ratio of positive to negative samples into consideration. As a large number of samples in the dataset are negative, the ones chosen by OHEM may also suffer from an imbalance of two samples, which is obviously detrimental to classification. Therefore, we proposed an improved OHEM, which chooses samples in a more balanced way: assuming that the loss function needs  $N$  samples, first, we sort positive and negative samples in descent order by loss, respectively; then, we choose the first  $S$  positive ones and  $N - S$  negative ones. The default value of  $S$  is set as  $N/4$ . In the ideal case, there are  $N$  samples chosen with a ratio of 1 : 3 [22]. Even if the total number is less than  $N$  and the ratio is not 1 : 3 exactly due to a lack of positive samples, these

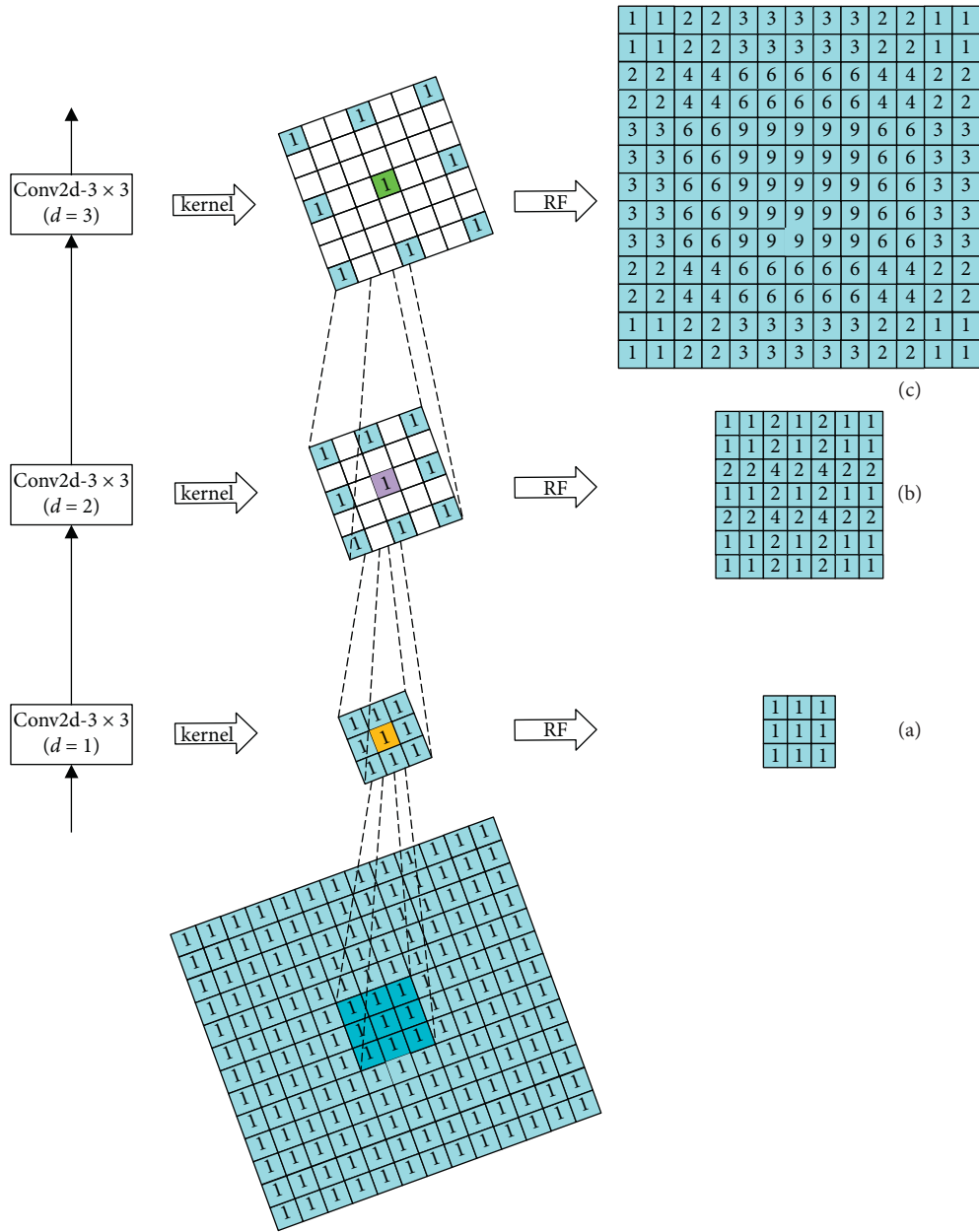


FIGURE 5: An illustration of the area that truly contributes to  $v$  when using HDC.

usually will not do harm to the performance of the algorithm. In contrast, they can enhance the robustness of the algorithm.

**3.2. The Overall Layout of HPCNet.** HPCNet is a one-stage multiscale face detection algorithm. To handle large-scale and small-scale faces, HPCNet introduced HDC module and HFP structure; to address with occlusion and blurring, HPCNet contained CIE.

HPCNet contains the convolution layers in VGG16 [30] as its basic network (as shown in Table 1). The overall structure is shown in Figure 8, where N4, N5, and N6 are three subnetworks for detecting different scales of face, namely, small, medium, and large. One thing that should be

noticed here is that all the convolution layers we used are  $3 \times 3$ , as they cut down on parameters and computation in addition to satisfying needs for processing.

In Figure 8, HDC6 refers to the proposed HDC module, which keeps in line with the architecture of VGG16 (as shown in Figure 9).

HFP module consists of HFP $x$ \_1 (including HFP4\_1 and HFP5\_1) and HFP $x$ \_2 (including HFP4\_2, HFP5\_2, and HFP6\_2). HFP $x$ \_1 (as shown in Figure 10) is to generate composite feature maps, which are passed forward as high-level semantic information; HFP $x$ \_2 is to generate hybrid feature maps, which combines information from both high and low layers. HFP4\_2 (as shown in Figure 11) uses  $3 \times 3$  convolution layer to reduce channel dimension of hybrid feature maps to 256, while

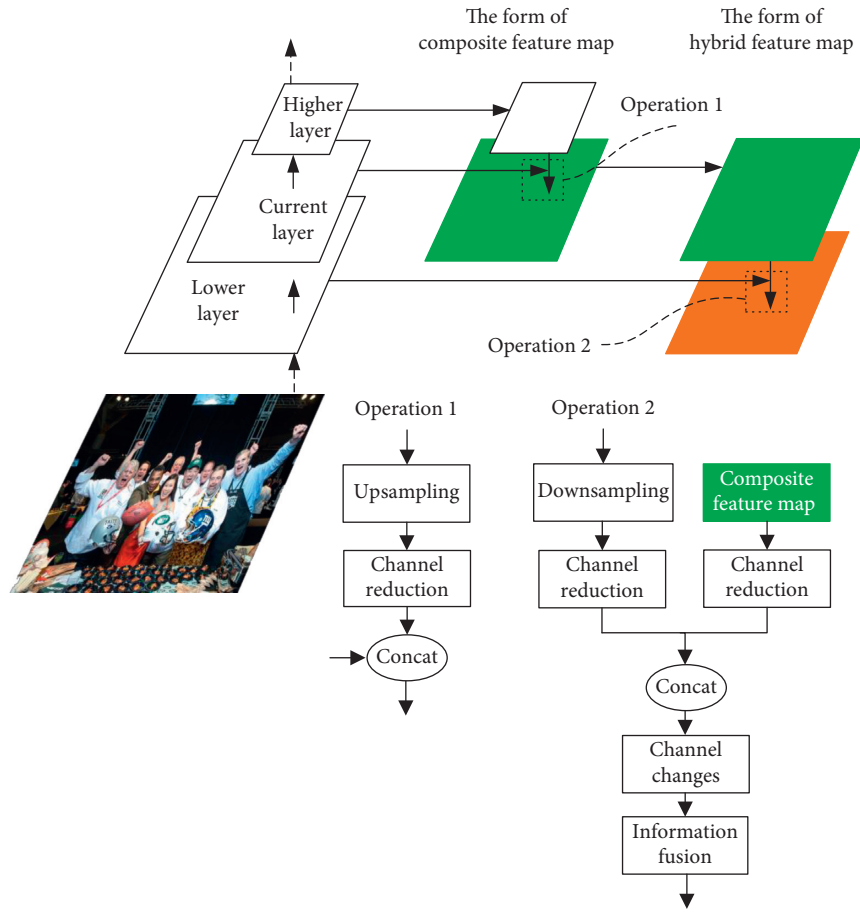


FIGURE 6: An illustration of the detailed structure of HFP.

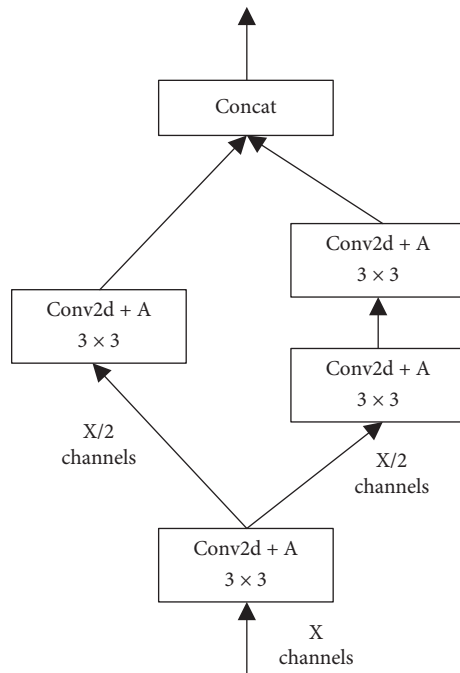


FIGURE 7: An illustration of the detailed structure of CIE.

TABLE 1: The basic structure of HPCNet.

Name	Configurations
Conv1_1	Conv2d + A, $3 \times 3 \times 64_{s1}$
Conv1_2	Conv2d + A, $3 \times 3 \times 64_{s1}$
Downsampling	MAxpool, $2 \times 2_{s2}$
Conv2_1	Conv2d + A, $3 \times 3 \times 128_{s1}$
Conv2_2	Conv2d + A, $3 \times 3 \times 128_{s1}$
Conv2_3	Conv2d + A, $3 \times 3 \times 128_{s1}$
Downsampling	MAxpool, $2 \times 2_{s2}$
Conv3_1	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv3_2	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv3_3	Conv2d + A, $3 \times 3 \times 512_{s1}$
Downsampling	MAxpool, $2 \times 2_{s2}$
Conv4_1	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv4_2	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv4_3	Conv2d + A, $3 \times 3 \times 512_{s1}$
Downsampling	MAxpool, $2 \times 2_{s2}$
Conv5_1	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv5_2	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv5_3	Conv2d + A, $3 \times 3 \times 512_{s1}$
Conv6	Conv2d + A, $3 \times 3 \times 512_{s2}$
HDC6	Conv2d + A, $3 \times 3 \times 512_{s1\_d1}$ Conv2d + A, $3 \times 3 \times 512_{s1\_d2}$ Conv2d + A, $3 \times 3 \times 512_{s1\_d3}$

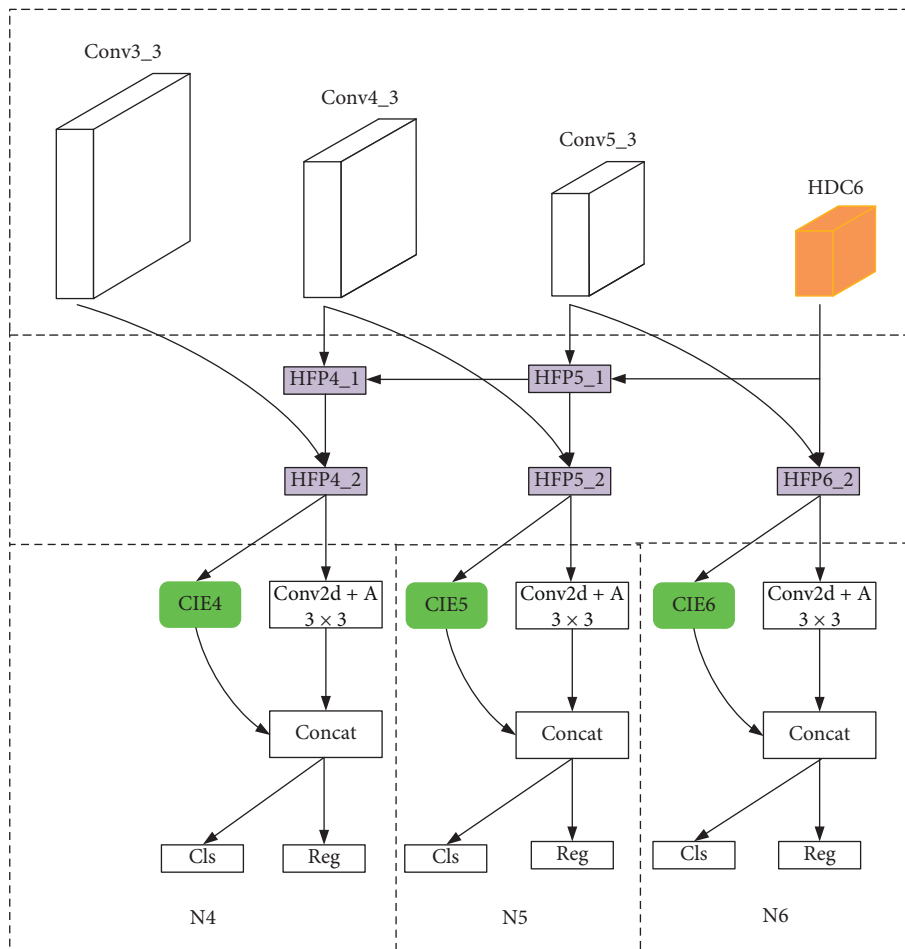


FIGURE 8: The overall layout of HPCNet.

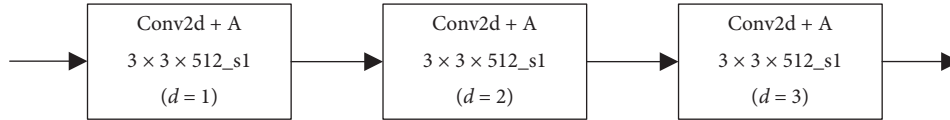


FIGURE 9: The structure of HDC6.

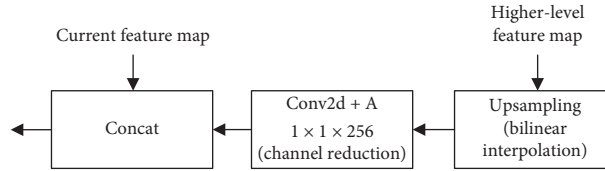


FIGURE 10: The structure of HFPx\_1.

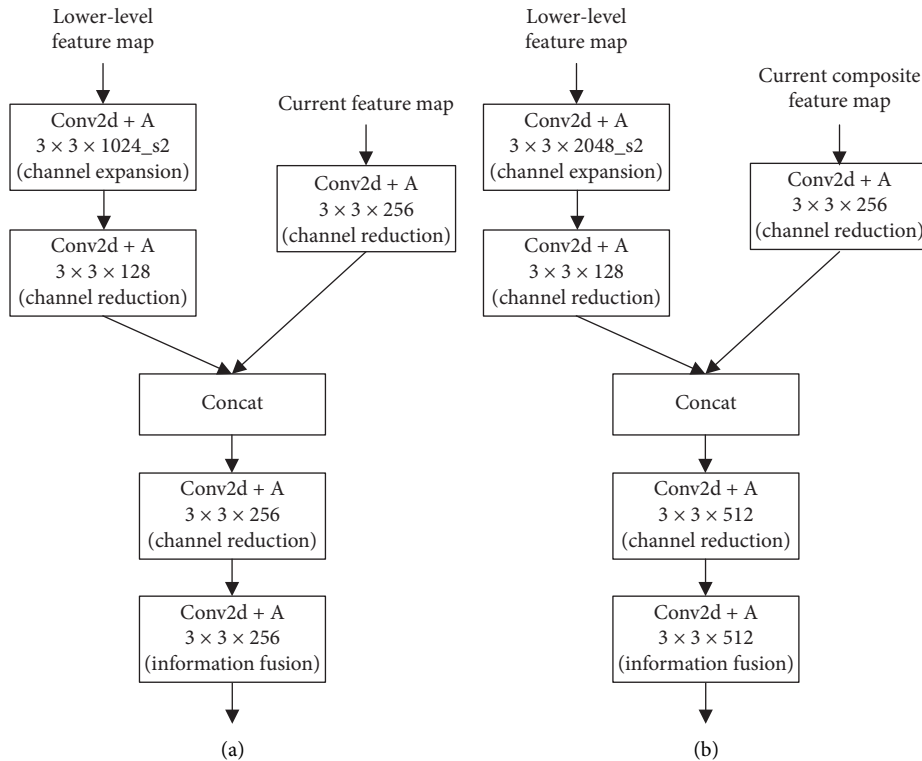


FIGURE 11: The structure of HFPx\_2. (a) HFP4\_2. (b) HFP5\_2 and HFP6\_2.

HFP5\_2 and HFP6\_2 increase channel dimension to 512. The reason for this difference in HFP4\_2 is to reduce memory occupation and keep in line with the following modules.

CIE4, CIE5, and CIE6 are three individual CIE modules, the structures of which are shown in Figure 7. Each sub-network contains one CIE module as one branch and one  $3 \times 3$  convolution layer as another. The feature maps for classification are generated by two branches together, the channel dimension of which is half that in HFPx\_2, respectively (as shown in Figure 8). In N5 and N6, the channel dimension through CIE5 and CIE6 is 256, while in N4, the channel dimension through CIE4 is 128. The reason for fewer channels in CIE4 is to reduce memory occupation and accelerate network convergence.

## 4. Experiment and Analysis

In this section, we first introduced some training strategies and parameter settings of HPCNet. Then, we conducted a series of ablation experiments on the WIDER FACE [27] dataset and compared HPCNet with other advanced algorithms to prove the effectiveness of our method.

### 4.1. Training Details

**4.1.1. Dataset.** All the experiments in this paper are based on WIDER FACE [27], which is the largest and most authoritative face image dataset in the world. In WIDER FACE, there are 32203 images containing 393703 labeled faces,

which is of high variability in terms of scales, occlusion, posture, and other aspects. In this paper, we randomly choose 40%, 10%, and 50% of the dataset as training, validation, and test set, respectively. In each set, the data is divided into three subsets (viz., Easy, Medium, and Hard) according to the difficulty level of detection.

Before entering into HPCNet, all the images are scaled to less than  $S \times L$ . To be more specific, we first scale the height of images to  $S$  pixels. After that, if its width is longer than  $L$  pixels, the width of this image is scaled to  $L$  pixels. During the scaling, the aspect ratio of all images remains unchanged.

**4.1.2. Hard Example Mining.** The feature maps generated by three subnets  $N_4$ ,  $N_5$ , and  $N_6$  correspond to  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$  area in the original image. The prior anchors used in  $N_4$ ,  $N_5$ , and  $N_6$  are  $16 \times 32$ ,  $64 \times 128$ , and  $256 \times 512$ , respectively. All of them have an aspect ratio of 1. During the training, we set the candidate proposals with Intersection over Union (IoU) higher than 0.5 as positive samples, while the ones with IoU lower than 0.3 as negative samples. After that, we process the dataset with an improved OHEM strategy.

**4.1.3. Loss Function.** To handle the problems of classification and regression at the same time, HPCNet adopts multitask loss function, which can be represented as

$$L_{\text{total}} = \sum_i \left( \frac{1}{N_i} \sum_{j \in B_i} L_{\text{conf}}(p_j, g_j) + \frac{\lambda}{N_i} \sum_{j \in B_i} I(g_j = 1) L_{\text{loc}}(b_j, t_j) \right), \quad (4)$$

where  $L_{\text{total}}$  represents the total loss,  $L_{\text{conf}}$  represents the classification loss, and  $L_{\text{loc}}$  represents the regression loss. For  $L_{\text{conf}}$ , we use *Softmax* function targeting at binary classification, in subnet  $N_i$ ,  $N_i$  represents the number of samples,  $B_i$  represents the whole dataset, and  $p_j$  and  $g_j$  represent the class score and label of  $j$ th sample. For  $L_{\text{loc}}$ , we use  $\text{Smooth}_{L_1}$  function with  $I$  representing the characteristic function: if the  $j$ th sample in subnet  $N_i$  is positive (i.e.,  $g_j = 1$ ), then  $I = 1$ ; otherwise  $I = 0$ . Here,  $b_j$  and  $t_j$  represent the coordinate prediction and preset value of the  $j$ th sample in subnet  $N_i$ ;  $\lambda$  controls the ratio of  $L_{\text{conf}}$  to  $L_{\text{loc}}$  (set as 1). If there is no positive sample in subnet  $N_x$ ,  $L_{\text{loc}}$  is set as 0.

**4.1.4. Hyperparameter Setting.** The weights in HPCNet are initialized by Gaussian function with an average of 0 and variance of 0.01. The bias is initialized as 0 and the regularization parameter is set as 0.0005. The training process adopts a batch SGD algorithm with a momentum of 0.9, itersize as 2, batchsize as 1, and initial learning rate as 0.004 (adopting StepLR policy with Gamma of 0.1 and stride as 18,000). Our HPCNet uses four GTX 1080Ti GPU to train for 21,000 times in total.

## 4.2. Ablation Experiment and Results

**4.2.1. Analysis of Improved OHEM.** We trained HPCNet with OHEM and improved OHEM, respectively, the result of which on WIDER FACE is shown in Table 2.

TABLE 2: The result of OHEM and improved OHEM.

Name	Easy	Medium	Hard
OHEM	0.924	0.910	0.795
Improved OHEM	0.920	0.908	0.819

The average precision (AP) of improved OHEM on Hard subset is 2.4% higher than that of OHEM, though it is 0.4% and 0.2% lower on Easy and Medium subsets. As Hard subset contains the most difficult cases which is closer to a real application, it is proved that improved OHEM is better. All the following experiments adopt improved OHEM.

**4.2.2. Analysis of HDC Module.** To test the effects of the HDC module, we get rid of HDC6 in HPCNet and named the network HPCNet-HDC6. The result is shown in Table 3. HPCNet-HDC6 is 1.6%, 1.2%, and 1.2% lower than HPCNet in terms of AP on each subset. Compared with the other two subsets, AP on Easy subset shows a bigger decrease, which proves that HDC6 is effective especially in detecting large-scale faces.

**4.2.3. Analysis of HFP Module.** To clearly illustrate the importance of low-level detailed information in HFP, we changed the architecture of HFPx\_2 by deleting the feature maps from lower layers. The changed HFPx is shown in Figure 12 and the network is named as HPCNet-Lx. From Table 3, we can see that HPCNet-Lx shows an obvious lower AP on all subsets, with an emphasis on the Hard subset (from 81.9% to 79.5%, reduced by 2.4%). This result has proved that low-level feature maps are essential to small-scale face detection.

**4.2.4. Analysis of CIE Module.** In this experiment, we removed the CIEx in HPCNet and set the number of channels in the main branch as the total number (256 in  $N_4$ , 512 in  $N_5$ , and  $N_6$ ). The changed network is named as HPCNet-CIEx. It is clear from Table 3 that HPCNet-CIEx is 1.6%, 1.2%, and 0.8% lower than HPCNet in terms of AP on each subset, which shows the effect of CIE on large-scale occlusion problems.

**4.2.5. Analysis of Multiscale Training.** We adopt multiscale training to HPCNet. To be more specific, it is randomly scaling images to  $400 \times 1600$ ,  $600 \times 1600$ ,  $800 \times 1600$ ,  $1000 \times 1600$ ,  $1200 \times 1600$ ,  $1400 \times 1600$ , and  $1600 \times 1600$ . All of these sizes follow the principle of  $S \times L$ . The result is shown in Table 4 and Figure 13. We name the model after multiscale training as HPCNet\_Pd.

From Table 4, we can see that the AP of HPCNet\_Pd on each subset is 1.3%, 1.6%, and 2.9% higher than HPCNet. The reason for this improvement is that multiscale training is indeed an image boosting strategy, which generates more faces of different scales and therefore enhances adaption and robustness of the model.

TABLE 3: The effect of different components.

Name	Easy	Medium	Hard
<b>HPCNet</b>	<b>0.920</b>	<b>0.908</b>	<b>0.819</b>
HPCNet-HDC6	0.904	0.896	0.807
HPCNet-Lx	0.912	0.900	0.795
HPCNet-CEx	0.904	0.896	0.811

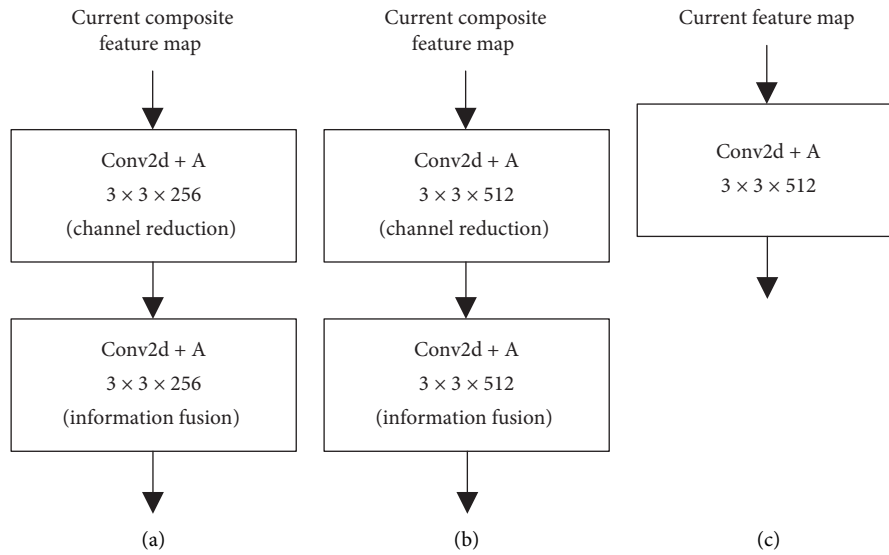


FIGURE 12: The structure of HFPx<sub>2</sub>. (a) HFP4<sub>2</sub>. (b) HFP5<sub>2</sub> and HFP6<sub>2</sub>. (c) HFP5<sub>2</sub> and HFP6<sub>2</sub>.

TABLE 4: Analysis of multiscale training.

Name	Easy	Medium	Hard
HPCNet	0.920	0.908	0.819
<b>HPCNet_Pd</b>	<b>0.933</b>	<b>0.924</b>	<b>0.848</b>

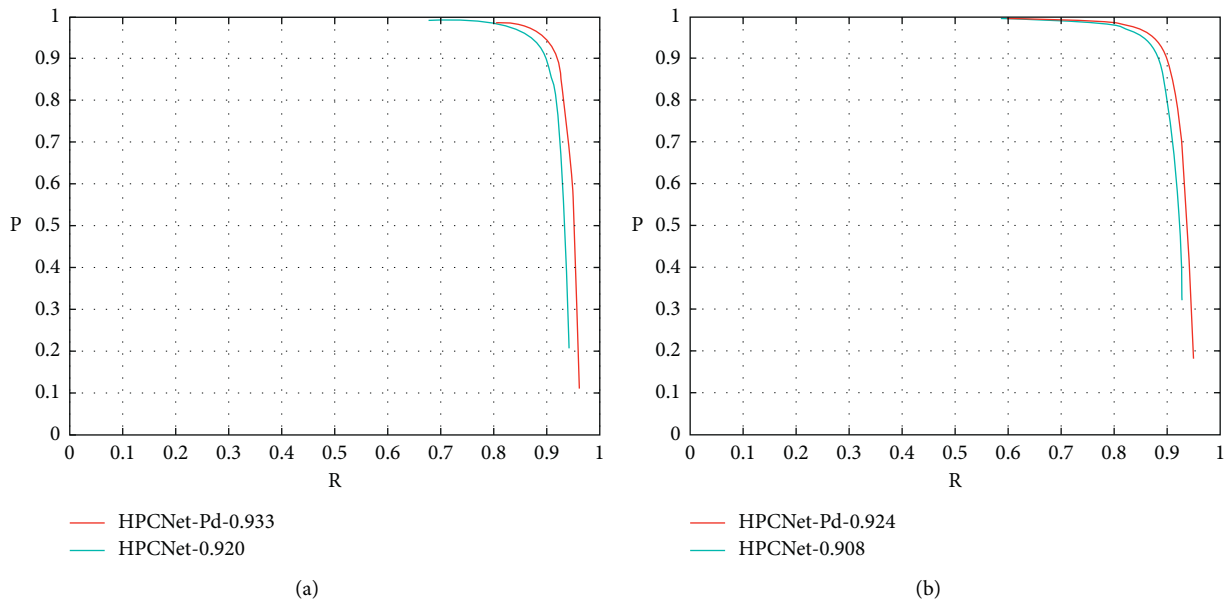


FIGURE 13: Continued.

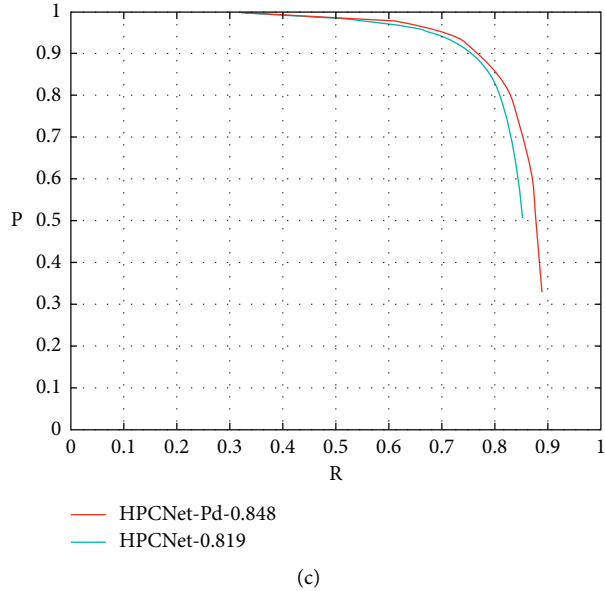
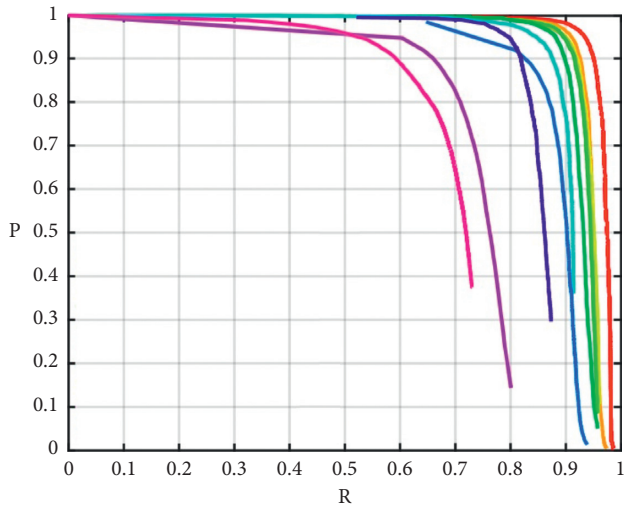
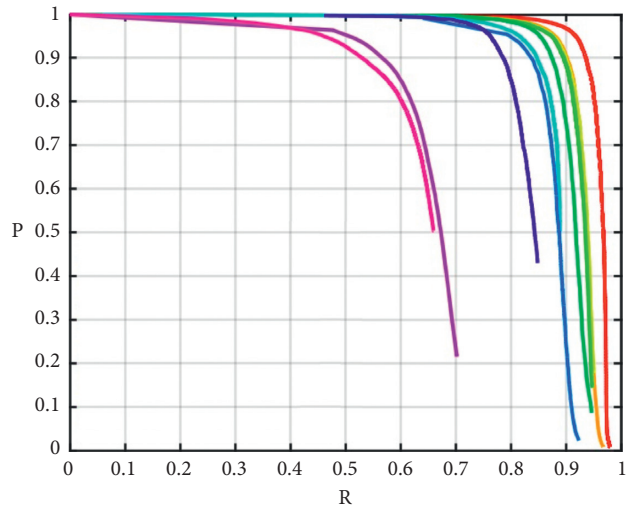


FIGURE 13: The PR curves of HPCNet and HPCNet\_Pd on each subset. (a) Easy, (b) Medium, and (c) Hard.



- DSFD-0.966
- SFD-0.937
- HPCNet-Pd-0.933
- SSH-0.931
- HR-0.925
- CMS-RCNN-0.899
- ScaleFace-0.868
- Multitask Cascade CNN-0.848
- Faceness-WIDER-0.713
- Two-stage CNN-0.681

(a)



- DSFD-0.957
- SFD-0.925
- HPCNet-Pd-0.924
- SSH-0.921
- HR-0.910
- CMS-RCNN-0.874
- ScaleFace-0.867
- Multitask Cascade CNN-0.825
- Faceness-WIDER-0.634
- Two-stage CNN-0.618

(b)

FIGURE 14: Continued.



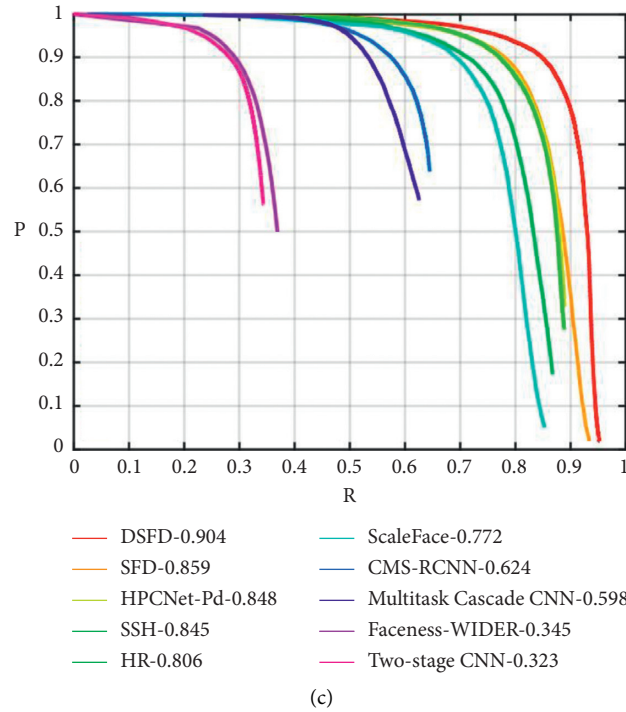


FIGURE 14: The PR curves of all the algorithms on WIDER FACE subset. (a) Easy, (b) Medium, and (c) Hard.

TABLE 5: Comparison between HPCNet and other algorithms.

Name	Easy	Medium	Hard	FPS
Two-stage CNN	0.681	0.618	0.323	<5
FacenessNet	0.713	0.634	0.345	<100
MTCNN	0.848	0.825	0.598	<95
ScaleFace	0.868	0.867	0.772	<5
CMS-RCNN	0.899	0.874	0.624	<15
HR	0.925	0.910	0.806	<5
SSH	0.931	0.921	0.845	<15
SFD	0.937	0.925	0.859	<35
DSFD	0.966	0.957	0.904	<20
<b>HPCNet_Pd</b>	<b>0.933</b>	<b>0.924</b>	<b>0.848</b>	<b>44</b>

4.2.6. *Comparison with Other Algorithms.* We choose several face detection algorithms to compare with HPCNet, namely, two-stage CNN [22], MTCNN [23], ScaleFace [25], SFD [28], DSFD [29], CMS-RCNN [32], SSH [34], HR [35], and FacenessNet [36]. The reason for choosing them is as follows:

- All of them are based on ConvNet
- They are representative in different genres
- They have a good performance on WIDER FACE
- They take both precision and time into consideration

The comparison result is shown in Figure 14 and Table 5. Despite HPCNet, all the AP and curve data are from the website of WIDER FACE [27]. Figure 14 directly shows differences between algorithms, where  $R$  represents recall rate and  $P$  represents precision.

It is clear from Table 5 that HPCNet has a higher AP on three subsets than classical algorithms including two-stage,



FIGURE 15: An illustration of detection result using HPCNet.

FacenessNet, MTCNN, ScaleFace, CNNCMS-RCNN, HR, and SSH. For the most advanced algorithms like SFD and DSFD, though HPCNet shows slightly lower AP, its running speed is much faster. The result has shown that HPCNet can have an advanced detection rate as well as running speed, which proves its reasonability and effectiveness. Figure 15 is an example of small-scale face detection by HPCNet.

## 5. Conclusion

Scaling and occlusion are the most challenging problems for face detection currently. We conducted research targeting these difficulties and proposed a one-stage, fully convolutional face detection framework HPCNet, which contains several designed components. In HPCNet, we introduced the concept of HDC and enlarged RF to handle large-scale faces. Meanwhile, we proposed a new HFP structure

combining high-level and low-level features together to enhance performance on small-scale faces. In addition, aimed at occlusion, we designed the CIE with fewer parameters. Particularly, we took advantage of improved OHEM and multiscale training strategy to balance the number of different samples as well as enhance robustness. By a series of ablation experiments, we proved the superiority of our HPCNet. In the future, the idea of this method can be applied to other computer vision tasks, such as person reidentification.

## Data Availability

The previously reported data were used to support this study and are available at 10.1109/CVPR.2016.596. These prior studies and datasets are cited at relevant places within the text as [27].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank the Center for Public Security Information and Equipment Integration Technology, UESTC, for providing computation platform.

## References

- [1] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 20, 1998.
- [2] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* London, UK, 1998.
- [3] A. Z. Kouzani, F. He, and K. Sammut, "Commonsense knowledge-based face detection," in *Proceedings of the IEEE International Conference on Intelligent Engineering Systems* London, UK, 1997.
- [4] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition* London, UK, 2000.
- [5] C. Liu, "A bayesian discriminating features method for face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 725–740, 2003.
- [6] S. Ferdinando, "Hmm-based architecture for face identification," *Image & Vision Computing*, vol. 23, 1994.
- [7] P. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* London, UK, 2001.
- [8] F. Pedro, Girshick, and B. Ross., "Object detection with discriminatively trained part-based models," in *Proceedings of the IEEE Transactions on Pattern Analysis & Machine Intelligence* London, UK, 2010.
- [9] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing* London, UK, 2002.
- [10] B. Yang, J. Yan, Z. Lei, and Z. Stan, "Aggregate channel features for multi-view face detection," *CoRR*, vol. 23, 2014.
- [11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, vol. 2, 2005.
- [12] I. Hadji and R. Wildes, "What Do We Understand about Convolutional Networks?" 2018.
- [13] Bo Wu, H. Ai, H. Chang, and S. Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition* London, UK, 2004.
- [14] H. Li, Z. Lin, X. Shen, J. Brandt, and H. Gang, "A convolutional neural network cascade for face detection," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* London, UK, 2015.
- [15] R. Joseph, S. Divvala, R. Girshick, and F. Ali, "You Only Look once: Unified, Real-Time Object Detection," 2015.
- [16] R. Joseph and F. Ali, "YOLO9000: better, faster, stronger," *CoRR*, vol. 2016, 2016.
- [17] R. Joseph and F. Ali, "Yolov3: an incremental improvement," *CoRR*, vol. 2018, 2018.
- [18] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *CoRR*, vol. 2015, 2015.
- [19] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and C. Alexander, "Dssd: Deconvolutional single shot detector," *CoRR*, vol. 2017, 2017.
- [20] S. Hou, Y. Li, Y. Pan, X. Yang, and G. Yin, "A face detection algorithm based on two information flow block and retinal receptive field block," *IEEE Access*, vol. 8, pp. 30682–30691, 2020.
- [21] R. Girshick, "Fast r-cnn," *Computer Science*, vol. 2015, 2015.
- [22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [23] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [24] J. Dai, Yi Li, and K. He, "R-fcn: Object Detection via Region-Based Fully Convolutional Networks," 2016.
- [25] S. Yang, Y. Xiong, and X. Tang, "Face Detection through Scale-Friendly Deep Convolutional Networks," 2017.
- [26] T.-Yi Lin, P. Dollár, R. B. Girshick et al., "Feature pyramid networks for object detection," *CoRR*, vol. 2016, 2016.
- [27] S. Yang, P. Luo, and X. Tang, "Computer vision and pattern recognition (cvpr)-wider face: a face detection benchmark," 2016.
- [28] S. Zhang, X. Zhu, Z. Lei et al., "S<sup>3</sup>fd: single shot scale-invariant face detector," *CoRR*, vol. 2017, 2017.
- [29] J. Li, Y. Wang, C. Wang et al., "DSFD: dual shot face detector," *CoRR*, vol. 2018, 2018.
- [30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, vol. 2014, 2014.
- [31] E. Adelson, C. Anderson, J. Bergen, P. Burt, and J. Ogden, "Pyramid methods in image processing," *RCA Engineering*, vol. 29, p. 11, 1983.
- [32] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," 2016.
- [33] A. Shrivastava, A. Gupta, and R. Girshick, "Training Region-Based Object Detectors with Online Hard Example Mining," 2016.

- [34] M. Najibi, P. Samangouei, R. Chellappa, and S. Larry, "SSH: single stage headless face detector," *CoRR*, vol. 2017, 2017.
- [35] P. Hu and D. Ramanan, "Computer vision and pattern recognition (cvpr)-finding tiny faces," 2017.
- [36] S. Yang, P. Luo, and X. Tang, "From facial parts responses to face detection: a deep learning approach," 2015.

## Research Article

# Improving Loanword Identification in Low-Resource Language with Data Augmentation and Multiple Feature Fusion

Chenggang Mi <sup>1</sup>, Shaolin Zhu <sup>2</sup>, and Rui Nie<sup>3</sup>

<sup>1</sup>School of Computer Science, Northwestern Polytechnical University, Xi'an, China

<sup>2</sup>College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, China

<sup>3</sup>Chinese Flight Test Establishment, Xi'an, China

Correspondence should be addressed to Chenggang Mi; [michenggang@nwpu.edu.cn](mailto:michenggang@nwpu.edu.cn)

Received 9 March 2021; Revised 18 March 2021; Accepted 25 March 2021; Published 8 April 2021

Academic Editor: Nian Zhang

Copyright © 2021 Chenggang Mi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Loanword identification is studied in recent years to alleviate data sparseness in several natural language processing (NLP) tasks, such as machine translation, cross-lingual information retrieval, and so on. However, recent studies on this topic usually put efforts on high-resource languages (such as Chinese, English, and Russian); for low-resource languages, such as Uyghur and Mongolian, due to the limitation of resources and lack of annotated data, loanword identification on these languages tends to have lower performance. To overcome this problem, we first propose a lexical constraint-based data augmentation method to generate training data for low-resource language loanword identification; then, a loanword identification model based on a log-linear RNN is introduced to improve the performance of low-resource loanword identification by incorporating features such as word-level embeddings, character-level embeddings, pronunciation similarity, and part-of-speech (POS) into one model. Experimental results on loanword identification in Uyghur (in this study, we mainly focus on Arabic, Chinese, Russian, and Turkish loanwords in Uyghur) showed that our proposed method achieves best performance compared with several strong baseline systems.

## 1. Introduction

Bilingual data play an very important role in cross-lingual natural language processing (NLP) tasks, such as cross-lingual text classification, cross-lingual information retrieval, and neural machine translation. However, bilingual data are often difficult to obtain. Lexical borrowing happens in almost every language; Figure 1 gives several loanwords in Uyghur (the reasons why we choose Uyghur as an example in our study are as follows: (1) there are many loanwords in Uyghur and (2) Uyghur is a low-resource language). If loanwords in low-resource languages can be identified effectively, it will be a novel way to alleviate the data sparseness existing in many cross-lingual NLP tasks.

Loanword identification is a task of finding out loanwords of a specific language (donor language) in texts in another language (receipt language). There are about three kinds of loanword identification methods: (1) rule-based method; (2) statistical-based method; and (3) deep learning-

based method. Early studies on loanword identification often based on rules. For example, McCoy and Frank [1] proposed a string similarity-based loanword identification model that relies on the ED algorithm. With the development of machine learning algorithms in NLP area, statistical-based methods are also proposed [2]. In recent years, deep learning algorithm such as bidirectional LSTM and convolutional neural network (BLSTM + CNN) are also used in loanword identification tasks [3]. Due to the lack of generalization ability of rule-based methods and limitation of training data in statistical-based methods, recent studies often combine the rule and statistical features together to improve the model performance effectively [4, 5]. However, almost all of these methods suffer from data sparseness during model training, especially in low-resource settings.

As a common used method to alleviate the data sparseness, data augmentation is one of the most popular methods in this topic. For example, Liu et al. [6] proposed to use a GAN model consisting of two generators and one

discriminator to produce meaningful natural language sentences. Motivated by this study, we propose to use a lexical constraint-based data augmentation model to generate more training data for loanword identification. Different from [6], we take the loanwords in training data as a lexical constraint to produce more sentences containing the loanwords.

After investigation, we find that there are two important clues in loanword identification: semantic similarity and pronunciation similarity. To incorporate these two features into one feature, we propose to transfer the semantic similarity as word-level feature and pronunciation similarity as character-level feature. Then, we fuse these two features into one feature. Meanwhile, we incorporate the fusion feature, pronunciation feature, and POS feature into a log-linear RNN to achieve the best performance in loanword identification.

The main contributions of this study are as follows:

- (i) First, a lexical constraint-based data augmentation method is proposed to generate more training data for loanword identification task.
- (ii) Second, we incorporate multilevel features, pronunciation similarity feature, and POS feature into a log-linear RNN model to improve the performance of the loanword identification model for low-resource language.
- (iii) Third, we conduct an experiment on loanword (Arabic, Chinese, Russian, and Turkish) identification in Uyghur; experimental results show that our proposed model achieves the best performance compared with several strong baseline systems.

The rest of this paper is organized as follows. Section 2 introduces some recent studies related to our topic. We present details of our proposed method in Section 3. Datasets, settings, and experimental results are described in Section 4. We show the analysis of experimental results in Section 5. In Section 6, we conclude this study and give some possible future directions.

## 2. Related Work

In this section, we present some work related to our study.

*2.1. Loanword Identification.* Lexical borrowing has received relatively little attention in natural language processing area. Tsvetkov and Dyer [7] proposed a morph-phonological transformation model to obtain good performance at predicting donor forms from borrowed forms. Tsvetkov et al. [7] suggested to use the lexical borrowing as a model in an SMT framework to translate OOV words. Gerz et al. [8] analyzed the implication of variation in structural and semantic properties in general language-independent architectures on the language modeling task. Mi et al. [9] used shallow features such as string similarity to detect loanwords in Uyghur. Mi et al. [3] presented a neural network-based loanword identification model that also incorporated several shallow features. However, these methods only trained

loanword identification models based on some monolingual corpora. It fails to project donor language and receipt language into one semantic space. The limitation of training data also exists.

*2.2. Data Augmentation for NLP.* The main goal of data augmentation in NLP is to generate additional, synthetic data using the data you have to alleviate the data sparseness during model training [10]. There are several data augmentation methods in NLP area [11]. The first one is lexical substitution which tries to substitute words present in a text without changing the meaning of the sentence [12]. The second one is back translation, which is commonly used in neural machine translation (NMT). Back translation first trains an intermediate system on the parallel data which is used to translate the target monolingual data into the source language. The result is a parallel corpus where the source side is synthetic machine translation output while the target is genuine text written by humans. The synthetic parallel corpus is then simply added to the real bitext in order to train a final system that will translate from the source to the target language [13]. The syntax-tree manipulation has been used in [14]; the idea is to parse and generate the dependency tree of the original sentence, transform it using rules, and generate a paraphrased sentence. Mixup is a simple yet effective image augmentation technique introduced by Zhang et al. [15]. The idea is to combine two random images in a mini-batch in some proportion to generate synthetic examples for training. The most recent data augmentation method is generative model; this kind of method tries to generate additional training data while preserving the class label [16].

*2.3. Sequence Labeling in NLP.* There are two main types of sequence labeling methods in NLP, such as gradient-based methods and search-based methods [17]. As for the probabilistic gradient-based learning methods such as conditional random fields (CRFs) and recurrent neural network (RNN), they have high accuracy because of the exact computation of the gradient and probabilistic information. Nevertheless, those methods have critical drawbacks. First, the probabilistic gradient-based methods typically do not support search-based optimization (search-based learning or decoding-based learning), which is important in sequence labeling problems with emphasis on the learning speed (e.g., for large-scale datasets). In tasks with complex structures, gradient computation is usually quite complicated sometimes and even intractable. This is mainly because dynamic programming for computing gradient is hard to scale for large-scale datasets. On the other hand, the search technique is easier to scale to large-scale datasets. This is because search-based learning is much simpler than gradient-based learning [18–20]—just search the promising output candidates and compare them with the oracle labels and update the weights accordingly. Another category of sequence labeling methods is the search-based learning methods (i.e., decoding-based learning), such as structured perceptron and MIRA. A major advantage of those methods is that they

	Origin	Source word	Source (in IPA)	Uyghur word	Uyghur (in IPA)	English
Persian		افسوس	[æf'sus]	ئەپسۇس <i>epsus</i>	[epsus]	pity
		گوشت	[go:ʃt]	گۆش <i>gösh</i>	[gøʃ]	meat
Arabic		ساعة	[sa:ʕat]	سائەت <i>saet</i>	[saʔet]	hour
		велосипед	[vɪləsɪ'piɛt]	ۋېلسىپىت <i>wəlsipit</i>	[wəlsipit]	bicycle
Russian		доктор	['dɒktər]	دوختۇر <i>doxtur</i>	[doχtur]	doctor
		поезд	['po:jst]	پويىز <i>poyiz</i>	[pojiz]	train
Chinese		область	['obləsi:t]	ئوبلاست <i>oblast</i>	[oblast]	oblast
		телевизор	[tɛlɪ'vɪzər]	تېلېۋىزور <i>tələwizor</i>	[televizor]	television set
		凉粉, <i>liángfěn</i>	[liɑŋ'fɛn]	لەڭپۇڭ <i>lempung</i>	[lempun]	agar-agar jelly
		豆腐, <i>dòufu</i>	[toʊ'fu]	دۇفۇ <i>dufu</i>	[dufu]	bean curd/tofu
		书记, <i>shūjǐ</i>	[ʃu'tɛi]	شۇجى <i>shuji</i>	[ʃuʃɪ]	secretary
	桌子, <i>zhuōzi</i>	[tʃwótsɪ]	جوزا <i>joza</i>	[ʃɔza]	table	
	冰箱, <i>bīngxiāng</i>	[píŋ'ɛjɑŋ]	بىڭشام <i>bingshang</i>	[biŋʃɑŋ]	refrigerator	

FIGURE 1: Examples of loanwords in Uyghur2.

support search-based learning, such that the gradient is not needed and the learning is done by simply searching and comparing the promising output candidates with the oracle labels and updating the model weights accordingly. As a by-product of the avoidance of gradient computation, those methods have faster training speed compared with probabilistic gradient-based learning methods like CRF.

### 3. Method

In previous studies, a large scale of annotated data is used to train a loanword identification model. They treated the loanword detection as a sequence labeling problem. However, the annotated data for loanword identification are very difficult to obtain. So, one of the contributions of this study is the data augmentation for loanword identification. We propose to use a lexical constraint GAN to generate more sentences for loanword identification model training. Another contribution of this paper is the combination of several features for loanword identification model; we introduce three features such as embedding fusion feature (word level and character level), pronunciation similarity feature, and POS feature.

**3.1. Overall Architecture.** Our proposed method includes two parts:

- (1) Data augmentation for loanword identification.
- (2) Log-linear RNN-based loanword identification model.

To generate more training data for loanword identification, we propose a lexical constraint GAN-based data augmentation model. Recent methods on loanword identification often trained on features such as pronunciation similarity, POS similarity, and so on. However, these kinds

of methods usually suffer from data sparseness or lack of semantic knowledge. To overcome this, we introduce a log-linear RNN-based loanword identification model which combines word-level and character-level embedding fusion features, pronunciation similarity, and POS features to predict Arabic, Chinese, Russian, and Turkish loanwords in Uyghur. The main idea of loanword identification in low-resource languages is as follows: we first use the data augmentation model to generate more training data for loanword identification in Uyghur; then, several features such as word- and character-level embedding features, pronunciation similarity, and POS features are proposed to build a multiple feature fusion-based loanword identification model (Figure 2).

#### 3.2. Data Augmentation for Loanword Identification.

Recent studies on loanword identification task often suffer from limitation of training data. In this study, we propose to use a lexical constraint GAN to generate more annotated data for the loanword identification task. As an extension of traditional GAN, our data augmentation model also includes two main parts: a generator and a discriminator. The difference is that we use two generators and a discriminator to build the data augmentation model for low-resource loanword identification. We introduce the details of our proposed model in this section.

**3.2.1. Generators.** We follow the work of [6] and extend the backward and forward generators to adapt to the loanword identification task. In our study, we use the loanwords of a specific language as the lexical constraint to generate more training data. Similar to [6], given a loanword, the backward generator takes it as the sentence's starting point and generates the first half sentence backwards. Then, the sequence produced by the backward generator is reversed and

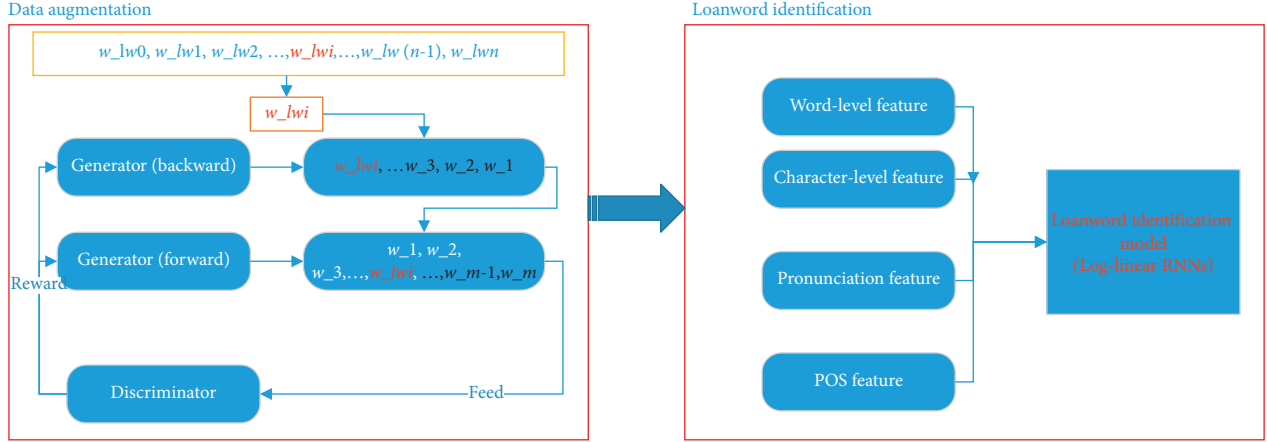


FIGURE 2: The framework of our proposed model.

fed into the forward generator. It then learns to generate the whole sentence with the aim of fooling the discriminator.

We can define the backward generator  $G_{\theta}^{(bw)}$  as

$$P_{\theta}^{(bw)}(s_{<c}|w_{lw}) = \prod_{i=1}^{lw-1} P_{\theta}^{(bw)}(w_{lw-i}|w_{lw}, \dots, w_{lw-i+1}), \quad (1)$$

where  $w_{lw}$  denotes a given loanword and  $l$  indicates the length of generated training sentence. The generated sentence is  $s = w_1 w_2, \dots, w_{lw}, \dots, w_l$ . The backward generator generates the first half of the sentence, while another half of sentence is generated by the forward generator.  $\theta$  and  $\theta'$  are parameters of the backward and forward generators.

The generator of the entire sentence can be defined as

$$G(s|w_c t; n\theta q, h\theta') = P_{\theta}^{(bw)}(s_{<c}|w_{lw}) P_{\theta'}^{fw}(s_{<c}|s_{1:lw}), \quad (2)$$

where  $P_{\theta}^{(bw)}(s_{<c}|w_{lw})$  and  $P_{\theta'}^{fw}(s_{<c}|s_{1:lw})$  are described as above.

The two generators have the same structure but have distinct parameters. To improve the coherence of the constrained sentence, we employ an LSTM-based language model with dynamic attention mechanism (called attRNN-LM) as generator.

**3.2.2. Discriminator.** Another important component in our proposed method is the discriminator, which takes sentence pairs as input and distinguishes whether a given sentence pair is real or generated. It guides the joint training of two generators by assigning proper reward signals. This module can be a binary classifier or a ranker. Following previous methods [21], we use Text-CNN as the discriminator which outputs a probability indicating whether the input is generated by humans or machines in the experiment.

**3.2.3. Data Augmentation Model.** To train the data augmentation model effectively, we first pretrain the backward and forward generators by standard MLE loss. Different from [6], we sample a loanword in our loanword list as the lexical constraint rather than select it randomly. Then, we use two generators and the lexical constraint to generate the

training sentence. The discriminator is trained based on real sentence as positive sample and sentences generated by generators as negative samples. The discriminator's output is the probability that the generated sentence is written by humans. We use the discriminator's output as the reward to encourage the two generators to work together to generate sentences which are indistinguishable from human-written sentence. To make the training stable and prevent the perplexity value skyrocketing, we apply teacher forcing to give the generators access to the gold-standard targets after each policy training step.

### 3.3. Multiple Feature Fusion-Based Loanword Identification.

Loanword identification can be defined as a sequence labeling problem. However, different from a traditional sequence labeling task, loanword identification task can apply some additional knowledge such as semantic similarity, pronunciation similarity, and POS tagging. As the data augmentation can provide us more annotated data for model training, we propose to use a deep neural network model to identify loanword in low-resource settings. The principle feature we used is the fusion of word- and character-level features, which combines the word relation and pronunciation similarity in loanword identification. We also incorporate external features such as pronunciation similarity and POS information into our method. In this section, we first describe features used in our proposed method and then define the details of the loanword identification method.

**3.3.1. Features.** We use three kinds of features in our proposed method: the fusion feature, pronunciation similarity, and POS feature.

**Fusion Feature.** In loanword identification task, word co-occurrence often plays a very important role. For example, in the English sentence "Tiananmen square is the most famous tourist destination in Beijing," the Chinese loanword "Tiananmen" is most related to the Chinese loanword "Beijing." In previous work, word embedding can capture word similarity and word relations with other words in a

sentence. Therefore, we apply self-attention to obtain word embedding in our study. The most important advantage of the self-attention is that it can model dependencies between words.

We use the dot-product attention in this study:

$$\text{DotAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^T)\mathbf{V}, \quad (3)$$

where  $\mathbf{Q}, \mathbf{K}, \mathbf{V}$  are query, key, and value vectors, respectively. It should be noted that the self-attention was obtained without scaling. We set

$$\mathbf{Q} = \mathbf{K} = \mathbf{V} = x_t^w, \quad (4)$$

and at time step  $t$ , the word embedding at time  $t$  based on self-attention can be defined as

$$h_t^{wl} = \text{DotAtt}(x_t^w, x_t^w, x_t^w). \quad (5)$$

The most important feature in loanword identification task is the pronunciation similarity between the word in receipt language and its corresponding word in donor language. As convolutional neural networks (CNNs) have been proven to capture the character-level information in NLP tasks, CNNs can process the sequences in the current receptive field akin to the attention mechanism [22]. Meanwhile, we also use max pooling to capture character-level features. The way we use CNN in our proposed method can be defined as

$$\text{Conv}(x_t^c) = \text{Mask}(x_t^c) * U. \quad (6)$$

We follow the study of [23] and use a CNN with a redundant position of input sequences masked to extract the character-level features.  $U$  is the filter width  $k$  set as 3. The convolution operation is denoted with  $*$ , and the padded position of input sequences is set as 0.

Max means a max pooling operation. We use it to capture the significant features assigned with the highest value for a given filter. Therefore, in the time step  $t$ , the character-level representation from local view is obtained as

$$h_t^{cl} = \text{Max}(\text{Conv}(x_t^c)). \quad (7)$$

To fuse the word-level and character-level features together, we propose to concatenate two features with automatic adjustment (Figure 3). The final fusion representation can be defined as

$$Z = \lambda_1 h_t^{wl} + \lambda_2 h_t^{cl}, \quad (8)$$

where  $h_t^{wl}$  and  $h_t^{cl}$  are word-level and character-level features, respectively, and  $\lambda_1$  and  $\lambda_2$  are corresponding parameters.

*Pronunciation Similarity Feature.* Intuitively, we find that a loanword often has a similar pronunciation with its corresponding donor word. A sample method to detect loanwords is to use a string similarity algorithm to compute the string similarity scores between the candidate loanword and a list of words in donor language. Then, we rank the scores and take the word with the best score as the donor word. In loanword identification task, we first transform donor and receipt language texts into a same writing system. For example, in Chinese

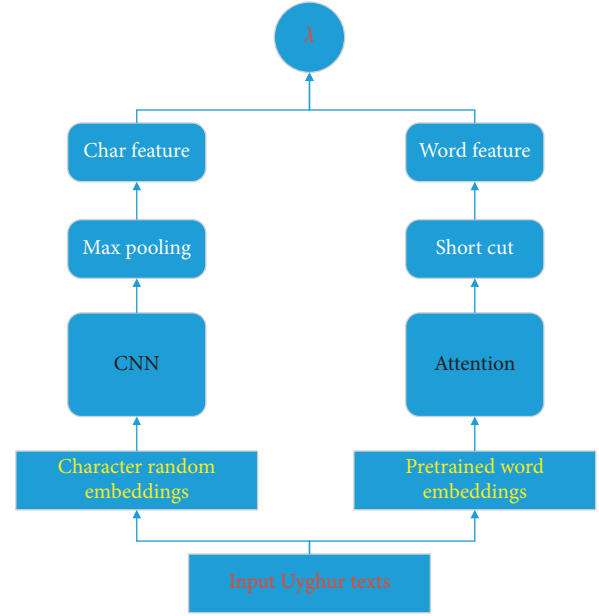


FIGURE 3: The multilevel feature fusion method used in our proposed loanword identification model. Character embeddings and word embeddings are taken as input for the feature selection layer.

loanword identification in Uyghur, we first convert these two language texts into Latin. Then, we apply the most commonly used string similarity algorithm—minimum edit distance (MinED)—in our loanword identification task.

$$h_{\text{med}}(\mathbf{l}_w, \mathbf{a}_{\text{crt}}, \mathbf{u}) = \sum_{j=0}^{l_{\text{a}_{\text{crt}}}} \sum_{i=0}^{l_{\mathbf{u}}} \text{Pr}(l_{w_i} | \text{med}(u_i, \mathbf{a}_{\text{crt}}_j)), \quad (9)$$

where  $l_{\text{a}_{\text{crt}}}$  and  $l_{\mathbf{u}}$  are lengths of donor word list and receipt word list, respectively,  $\mathbf{a}_{\text{crt}}$  and  $\mathbf{u}$  represent donor languages (Arabic, Chinese, Russian, and Turkish) and receipt language (in this study indicates Uyghur),  $l_{w_i}$  is the loanword label of the  $i$ th receipt word, and  $\text{med}(u_i, \mathbf{a}_{\text{crt}}_j)$  is the minimum edit distance of two words. To adapt the loanword identification task, we first conduct text normalization on all datasets, which transform a text into a canonical (standard) form. Then, we carry on morphological segmentation on morphologically rich languages, such as Uyghur, Russian, and Turkish.

*POS Feature.* As loanwords are often nouns, we propose a part-of-speech (POS) feature to further constrain the loanword identification model. We first pretrain POS tagging models for donor languages and receipt language. Considering both the language resource and performance, we select CRF as the framework of POS tagging model. As POS models are ready, if a word in receipt and its corresponding candidate donor word are all nouns, we set the POS features as 1.

*3.3.2. Loanword Prediction Model.* Log-linear models play a considerable role in statistics and machine learning. The most important reason we chose the log-linear model as the basic framework of our proposed loanword prediction



model was because features can be easily added into it. Additionally, the log-linear model has been widely used in NLP tasks such as SMT and NMT.

To adapt the loanword prediction task and include rich features such as BiLSTM, POS, and semantic feature into the model, we use log-linear RNNs [24] as the basic framework in our task. Log-linear RNN is similar to a RNN model. It allows a more general form of input to the network at each time step; that is, instead of allowing only the latest symbol  $x_t$  to be used as input, along with the condition  $C$ , it now allows an arbitrary feature vector  $\psi(C, x_1, x_2, \dots, x_{t-1}, x_t)$  to be used as input; this feature vector is of fixed dimensionality  $|\psi|$  and allows it to be computed in an arbitrary (but deterministic) way from the combination of the currently known prefix  $x_1, x_2, \dots, x_{t-1}, x_t$  and the context  $C$ . This is a relatively minor change, but one that usefully expands the expressive power of the network.

The hidden state at time  $t$  in our loanword identification task can be defined as

$$\begin{aligned} p_{\theta,t}(x) \\ \propto b(C, x_1, x_2, \dots, x_{t-1}, x_t) \\ \cdot \exp(a_{\theta,t}^T \phi(C, x_1, x_2, \dots, x_{t-1}, x_t)). \end{aligned} \quad (10)$$

We assume that we have a priori fixed a certain background function  $b(C, x_1, x_2, \dots, x_{t-1}, x_t)$  and also defined  $M$  features defining a feature vector  $\phi(C, x_1, x_2, \dots, x_{t-1}, x_t)$  of fixed dimensionality  $\phi(C, x_1, x_2, \dots, x_{t-1}, x_t)$ .

Therefore, the loanword label of  $t + 1$  word  $x_{t+1}$  can be defined as

$$x_{t+1} \sim p_{\theta,t}(\cdot). \quad (11)$$

During training of our proposed loanword identification model, we use the cross-entropy loss to optimize the performance of our model [25].

## 4. Experiments

In this section, we evaluate the effectiveness of our proposed method.

**4.1. Data.** To fully evaluate the effectiveness of our proposed model, we conduct Arabic, Chinese, Russian, and Turkish loanword identification in Uyghur. The datasets used in our experiments are listed in Table 1. We crawl these corpora from the Internet. Then, we annotate a small part with loanword label by hands. In all texts, we assure that each sentence includes at least one loanword.

To train the data augmentation model, we also collect some monolingual data from Internet for each language (Table 2).

### 4.2. Settings

**4.2.1. Data Augmentation.** We train the data augmentation model on datasets described in Table 2. We set the same hyperparameters for forward and backward generators. All

TABLE 1: Size of datasets.

Data type	Size			
	Arabic	Chinese	Russian	Turkish
Sentences	100, 780	125, 085	143, 290	132, 500
Loanwords	690	2,450	1,274	2,009

TABLE 2: Size of monolingual data.

Languages	Uyghur	Arabic	Chinese	Russian	Turkish
Size (words)	0.32	1.05 B	1.70 B	1.14 B	1.49 B

generators include 2-layer char-level LSTMs with 1024 hidden units. The dimension of word embeddings is set to 1024; the batch size, dropout rate, threshold of element-wise gradient clipping, and initial learning rate of Adam optimizer are set to 128, 0.5, 5.0, and 0.001; layer normalization is also applied. We set both backward and forward generators to one layered word-level LSTM with 1024 hidden units when training on datasets described in Table 2. For the hyperparameters of the discriminator, the filter window size is set to be 3, 4, 5, 6, and 7, and the kernel number of each filter is 512. We set the batch size as 64 and the number of iterations as 5000.

**4.2.2. Loanword Identification.** We implemented the log-linear RNNs by ourselves. We also developed the extended version of edit distance algorithm to adapt the loanword identification task. For the POS feature, we first pretrained a Uyghur POS tagging model; then, we tagged all Uyghur sentences based on this model.

We compared our method with several strong baseline systems: Rule [1], CRF [2], BLSTM-CNN [3], and CIEmbedding [4].

**4.3. Results on Data Augmentation.** Results on data augmentation and size of training data can be found in Tables 3 and 4, respectively.

**4.4. Results on Loanword Identification.** The results on loanword identification on different methods can be found in Table 5.

## 5. Analysis

Table 3 presents experimental results on data augmentation for loanword identification. We can find that our proposed lexical constraint method achieves the best performance compared with other strong baseline systems in all evaluation metrics. The most important reason is that our method guarantees the fluency and semantic consistency of generated sentence at the same time. Table 4 shows the size of Uyghur sentence (with loanword in different donor languages) generated by our proposed data augmentation model. For loanwords in different donor languages, we obtain the largest Uyghur datasets with Turkish loanwords; one possible reason is that Uyghur and Turkish are closely

TABLE 3: Evaluation of data augmentation methods.

Donor	Metrics	B/F-LM	BF-MLE	Ours
Arabic	BLEU-4	0.15	0.15	0.21
	Self-BLEU	64.32	64.58	63.46
	TER	66.19	66.44	65.82
Chinese	BLEU-4	0.16	0.17	0.23
	Self-BLEU	64.05	64.30	63.78
	TER	64.23	65.02	63.98
Russian	BLEU-4	0.18	0.18	0.23
	Self-BLEU	62.76	63.05	62.64
	TER	63.69	63.92	63.45
Turkish	BLEU-4	0.19	0.20	0.25
	Self-BLEU	62.51	62.86	62.18
	TER	62.46	63.14	62.04

TABLE 4: Size of training data generated in data augmentation (Uyghur sentences).

Lang	Arabic	Chinese	Russian	Turkish
Size	302, 480	325, 790	314, 208	336, 852

TABLE 5: Loanword identification experimental results on different methods.

Donor	Model	Loanword identification results (%)					
		P	P(+)	R	R(+)	F1	F1(+)
Russian	Rule (+)	72.04	72.89	69.31	70.18	70.65	71.28
	CRF (+)	71.63	72.45	67.28	68.15	69.39	70.23
	BLSTM-CNN (+)	71.45	72.26	70.50	71.31	70.97	71.78
	CIEmbedding (+)	73.12	73.94	71.84	72.62	72.47	73.27
	Ours (+)	74.80	75.62	73.64	74.20	74.22	74.90
Arabic	Rule (+)	69.05	69.84	68.17	69.02	68.61	69.43
	CRF (+)	69.83	70.65	67.42	68.29	68.60	69.45
	BLSTM-CNN (+)	68.70	69.52	69.85	70.67	69.27	70.09
	CIEmbedding (+)	72.95	73.76	72.03	72.85	72.49	73.30
	Ours (+)	73.91	74.62	72.35	73.06	73.12	73.83
Turkish	Rule (+)	72.02	72.86	69.87	70.50	70.93	71.66
	CRF (+)	71.46	72.29	69.02	69.95	70.22	71.10
	BLSTM-CNN (+)	71.25	72.04	70.43	71.18	70.84	71.61
	CIEmbedding (+)	72.96	73.64	73.08	73.85	73.02	73.74
	Ours (+)	75.24	76.09	74.36	75.14	74.80	75.61
Chinese	Rule (+)	70.32	71.13	69.77	70.58	70.04	70.85
	CRF (+)	70.85	71.64	69.24	70.05	70.04	70.84
	BLSTM-CNN (+)	70.58	71.34	69.98	70.79	70.28	71.06
	CIEmbedding (+)	71.67	72.48	71.35	72.14	71.51	72.31
	Ours (+)	74.30	75.07	72.88	73.95	73.58	74.51

related. We obtain the fewest sentences with Arabic; it is because Uyghur and Turkish have very different grammar and syntax.

The first part in Table 5 describes experimental results on different methods with the original training data. We found that the CRF and rule-based model outperform BLSTM-CNN method; one possible reason is the limitation of annotated data. Because the CIEmbedding model can exploit semantic information obtained from monolingual data, the CIEmbedding model achieves slightly better results compared with the CRF and rule-based model. Compared with other baseline models, our method incorporates word-level and character-level features pretrained from monolingual corpora into one model; therefore, our method achieves best results, but the improvement is not significant. This is because our method also suffers from data sparseness during model training.

The second part of Table 5(with (+)) presents loanword identification results on different methods with our generated training data (data augmentation). We can find that the generated training data improve all baseline models significantly. The CRF-based model has the ability of generalization, but the data sparseness still weakens the loanword identification performance significantly. The BLSTM-CNN + method also achieves better performance compared with the BLSTM-CNN. Both CRF+ and BLSTM-CNN + benefit from data augmentation. Although CIEmbedding + relies on monolingual data, it also obtains performance improvements due to loanword identification results are added. Our proposed method incorporates RNN features and external features into one model, so it achieves the best performance among all baseline systems.

Table 6 presents results on different features in our proposed method (we take Turkish and Chinese loanword identification as examples). We find that models with all features achieve best performance in both Turkish and Chinese loanword identification tasks. As for single feature, the fusion feature is more important than others; one possible reason is that the fusion feature combines word-level and character-level features at the same time. Except the fusion feature, pronunciation similarity feature outperforms other features because the pronunciation similarity is the most intuitive feature in loanword identification task. Although the POS cannot achieve comparative performance with others, we find that the combination features with POS always outperform others.

In Table 5, we describe results on different donor languages. We can easily find that our method achieves best performance on Turkish loanword identification task. One important reason is that Turkish and Uyghur belong to the same language family, and they share much vocabulary and grammar compared with other donor languages. Our model also achieves better results on Russian loanword identification than Chinese and Arabic; one possible reason is that Russian has a deep influence on Uyghur, and Uyghur is sometimes written in a Cyrillic alphabet, which is the basic writing system in Russian. Because people who can speak

TABLE 6: Loanword identification results on different features (Turkish and Chinese loanword identification as examples).

Donor	Feature(s)	Loanword identification results (%)					
		P	P(+)	R	R(+)	F1	F1(+)
Turkish	+fusion	74.14	74.95	73.28	74.16	73.71	74.55
	+pronun	73.96	74.68	73.02	73.94	73.49	74.31
	+pos	72.54	73.36	72.25	73.07	72.39	73.21
	+fusion, pronun	73.40	74.20	72.64	73.40	73.02	73.80
	+fusion, pos	74.63	75.42	73.70	74.52	74.16	74.97
	+pronun, pos	74.25	75.06	73.45	74.24	73.85	74.65
	+all	75.24	76.09	74.36	75.14	74.80	75.61
Chinese	+fusion	73.15	73.94	71.74	72.56	72.44	73.24
	+pronun	72.76	73.52	71.32	72.16	72.03	72.83
	+pos	71.30	72.09	70.58	71.25	70.94	71.67
	+fusion, pronun	72.43	73.25	71.02	71.84	71.72	72.54
	+fusion, pos	73.61	74.40	72.26	73.02	72.93	73.70
	+pronun, pos	73.25	74.03	71.97	72.89	72.60	73.46
	+all	74.30	75.07	72.88	73.95	73.58	74.51

Uyghur can often speak Chinese fluently, Chinese has a significant impact on Uyghur. Although Uyghur and Arabic share the same writing system, two languages belong to different language families. So, Arabic loanword identification achieves the worst performance.

## 6. Conclusion

The main goal of this study is to improve the performance of loanword identification for low-resource language. Our contribution includes two parts: (1) data augmentation for loanword identification and (2) loanword identification based on multiple feature fusion. In particular, data augmentation alleviates the data sparseness occurring in the loanword identification model training; we optimize the loanword identification model by introducing several features such as fusion feature of word- and character-level embeddings, pronunciation similarity, and POS feature into one model based on a log-linear RNN. To evaluate the effectiveness of our proposed method, we conduct experiments on several baseline models. Experiments show that our proposed loanword identification method achieves the best performance.

In our future work, we plan to improve the robustness of the loanword identification model by generating more diverse training data and incorporating richer contextual information into it.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was funded by the National Natural Science Foundation of China (no. 61906158).

## References

- [1] R. T. McCoy and R. Frank, "Phonologically informed edit distance algorithms for word alignment with low-resource languages," in *Proceedings of the Society for Computation in Linguistics (SCiL'18)*, pp. 102–112, Salt Lake City, UT, USA, January 2018.
- [2] G. Akın Şeker and G. Eryiğit, "Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content 1," *Semantic Web*, vol. 8, no. 5, pp. 625–642, 2017.
- [3] C. Mi, Y. Yang, L. Wang, Xi Zhou, and T. Jiang, "A neural network based model for loanword identification in Uyghur," in *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan, May 2018.
- [4] C. Mi, Y. Yang, L. Wang, Xi Zhou, and T. Jiang, "Toward better loanword identification in Uyghur using cross-lingual word embeddings," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3027–3037, Santa Fe, NM, USA, August 2018.
- [5] C. Mi, L. Xie, and Y. Zhang, "Loanword identification in low-resource languages with minimal supervision," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 3, pp. 1–22, 2020.
- [6] D. Liu, J. Fu, Q. Qu, and J. Lv, "BFGAN: backward and forward generative adversarial networks for lexically constrained sentence generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2350–2361, 2019.
- [7] Y. Tsvetkov and C. Dyer, "Lexicon stratification for translating out-of-vocabulary words," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2, Beijing, China, July 2015.
- [8] D. Gerz, I. Vulić, E. M. Ponti, R. Reichart, and K. Anna, "On the relation between linguistic typology and (limitations of) multilingual language modeling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 316–327, Brussels, Belgium, November 2018.
- [9] C. Mi, Y. Yang, X. Zhou, L. Wang, Li Xiao, and T. Jiang, "Recurrent neural network based loanwords identification in Uyghur," in *Proceedings of the 30th Pacific Asia Conference on Language, Information, and Computation: Oral Papers*, pp. 209–217, Seoul, Korea, October 2016.
- [10] J. Liu, Y. Pan, F. X. Wu, and J. Wang, "Enhancing the feature representation of multi-modal MRI data by combining multi-view information for MCI classification," *Neurocomputing*, vol. 400, pp. 322–332, 2020.
- [11] C. Amit, "A visual survey of data augmentation in NLP," 2020, <https://amitnss.com/2020/05/data-augmentation-for-nlp>.
- [12] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," 2015, <https://arxiv.org/abs/1509.01626>.
- [13] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," 2018, <https://arxiv.org/abs/1808.09381>.

- [14] C. Coulombe, "Text data augmentation made simple by leveraging nlp cloud apis," 2018, <https://arxiv.org/abs/1812.04718>.
- [15] H. Zhang, M. Cisse, Y. N. Dauphin, and L.-P. David, "Mixup: beyond empirical risk minimization," 2017, <https://arxiv.org/abs/1710.09412>.
- [16] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," 2020, <https://arxiv.org/abs/2003.02245>.
- [17] X. Sun, S. Ma, Y. Zhang, and X. Ren, "Towards easier and faster sequence labeling for natural language processing: a search-based probabilistic online learning framework (SAPO)," *Information Sciences*, vol. 478, pp. 303–317, 2019.
- [18] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 213–220, Edmonton, Canada, September 2003.
- [19] X. Sun, "Structure regularization for structured prediction," *NIPS*, vol. 14, pp. 2402–2410, 2014.
- [20] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt et al., "Accelerated training of conditional random fields with stochastic gradient methods," in *Proceedings of the 23rd International conference on Machine learning*, pp. 969–976, Pittsburgh, PA, USA, June 2006.
- [21] L. Yu, W. Zhang, J. Wang et al., "Seqgan: sequence generative adversarial nets with policy gradient," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.
- [22] J. Liu, D. Zeng, R. Guo, M. Lu, F. X. Wu, and J. Wang, "Mmhge: detecting mild cognitive impairment based on multi-atlas multi-view hybrid graph convolutional networks and ensemble learning," *Cluster Computing*, vol. 24, no. 1, pp. 103–113, 2021.
- [23] Z. Yang, H. Chen, J. Zhang et al., "Attention-based multi-level feature fusion for named entity recognition," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI-20)*, pp. 3594–3600, Yokohama, Japan, January 2020.
- [24] M. Dymetman and C. Xiao, "Log-linear RNNs: towards recurrent neural networks with flexible prior knowledge," 2016, <https://arxiv.org/abs/1607.02467>.
- [25] J. Cheng, J. Liu, H. Yue, H. Bai, Y. Pan, and J. Wang, "Prediction of glioma grade using intratumoral and peritumoral radiomic features from multiparametric MRI images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

## Research Article

# A New Random Forest Algorithm Based on Learning Automata

Mohammad Savargiv <sup>1</sup>, Behrooz Masoumi <sup>1</sup> and Mohammad Reza Keyvanpour<sup>2</sup>

<sup>1</sup>Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

<sup>2</sup>Department of Computer Engineering, Alzahra University, Tehran, Iran

Correspondence should be addressed to Behrooz Masoumi; [masoumi@qiau.ac.ir](mailto:masoumi@qiau.ac.ir)

Received 12 February 2021; Revised 9 March 2021; Accepted 16 March 2021; Published 27 March 2021

Academic Editor: Nian Zhang

Copyright © 2021 Mohammad Savargiv et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The goal of aggregating the base classifiers is to achieve an aggregated classifier that has a higher resolution than individual classifiers. Random forest is one of the types of ensemble learning methods that have been considered more than other ensemble learning methods due to its simple structure, ease of understanding, as well as higher efficiency than similar methods. The ability and efficiency of classical methods are always influenced by the data. The capabilities of independence from the data domain, and the ability to adapt to problem space conditions, are the most challenging issues about the different types of classifiers. In this paper, a method based on learning automata is presented, through which the adaptive capabilities of the problem space, as well as the independence of the data domain, are added to the random forest to increase its efficiency. Using the idea of reinforcement learning in the random forest has made it possible to address issues with data that have a dynamic behaviour. Dynamic behaviour refers to the variability in the behaviour of a data sample in different domains. Therefore, to evaluate the proposed method, and to create an environment with dynamic behaviour, different domains of data have been considered. In the proposed method, the idea is added to the random forest using learning automata. The reason for this choice is the simple structure of the learning automata and the compatibility of the learning automata with the problem space. The evaluation results confirm the improvement of random forest efficiency.

## 1. Introduction

Random forest is one of the methods of ensemble learning that comes under the homogeneous base learner category in terms of the type of constructive classifiers. As the name implies, all base learners are decision trees, and therefore they have a simpler structure than similar methods [1]. The random forest structure has two advantages. The first category is from a computational point of view, and the second category is from a statistical point of view. Advantages that can be considered from a computational point of view are: the random forest has the ability to deal with both regression and classification issues. The train and prediction processes in this classifier are performed at high speed, and therefore the random forest is known as one of the fast classic classifiers. Another advantage of the random forest is its ability to be used directly in high-dimensional issues [2]. The advantages of the second view of the random forest are its

characteristics, namely, prioritization of features, attribution of different weight coefficients to different classes, and illustration and unsupervised learning ability.

According to the literature, the random forest method is one of the most practical methods of ensemble learning. Weighting the base learners in ensemble learning is one of the main challenges in aggregating the basic classifiers in order to achieve a stronger classifier [3]. The reason for weighting base learners, or in other words, determining the impact factor for each base learner, is to increase the scalability of the data mining algorithm with the problem space. This becomes even more apparent when the environment is dynamic, and different or sometimes contradictory behaviours are observed from data in different situations. The text data environment has such an interesting behaviour that it challenges data mining algorithms. For example, placing one word on one domain may create a positive polarity, but it may also create a negative polarity on another domain. This

difference in polarity is created without any change in the form of the word and without any change in the role of the word from a grammatical point of view. The word “small” in both the electronic domain and the restaurant domain has such a behaviour. This behaviour poses a major challenge to the opinion mining algorithms [4].

The classical solution in the literature to overcome this challenge is based on the use of lexical-based approaches. This approach is based on frameworks such as unigram, n-gram, aspect-based, and similar methods, and all of them are data-dependent. In addition to the urgent need for predefined data, these methods lose their efficiency if they are met with an unspecified word or metaphor in the opinion mining field. In other words, they are not compatible with the problem space. The way random forest works is that with the sequential placement of training data and feature vectors that are injected into each of the base learners, it tries to find the best subset of features, and by increasing their impact factor in the classifier, it achieves the highest performance among all the aggregated base learners [5]. However, this method is not effective in relation to data such as text, in which a word can have different polarities in different domains because, in the classification algorithm, there is no ability to adapt to the conditions of the problem space.

In this paper, we intend to empower random forest with the idea of reinforcement learning and improve its efficiency. In the proposed method, learning automata is used to aggregate and weigh base learners. The way learning automata works is to receive feedback from the environment and perform one of the actions based on the type of feedback. In the learning automata, feedbacks are divided into two categories of reinforcement signals: reward signals and penalty signals. For each reinforcement signal received by the learning automata, it updates the probability of selecting the selected action in the previous step. This process continues until the probability of action selections converges to one of the actions; in other words, the best option for running in the current situation is found. In the proposed method, learning automata actions are appropriate when one of the base learners selected leads to the maximum reward that can be received from the environment. Since at each stage of learning automata execution, the learning algorithm tries to select the best option, achieving global optima in the problem space is guaranteed. This is proof of the adaptability of the proposed method. In the proposed method, the subprocess of replacing features in the feature vector is removed, and all the features in the feature vector are used. As a practical application in the field of opinion mining, if the Bag of Word (BoW) method is used to create the feature vector, the advantage of considering all the features of the feature vector will also cover cases that occur rarely. In other words, in the proposed method, the aspect of independence from the domain in the processes such as opinion mining is considered.

Our contribution is summarized as follows:

In this paper, a brief review of random forest in terms of application scope is given.

In this paper, a learning automata-based method is proposed to improve the random forest performance.

The proposed method operates independently of the domain, and it is adaptable to the conditions of the problem space.

The rest of the paper is organized as follows. In Section 2, related work is introduced. Section 3 presents the introduction to learning automata. The proposed method is explained in Section 4. Section 5 includes evaluation. Discussion is given in Section 6, and finally, the conclusion and future work are described in Section 7.

## 2. Related Work

In this section, theories and literature on the subject of random forest are examined. The purpose of this section is to review the innovations that have been introduced around random forest in recent years.

Random forest is considered as one of the methods of ensemble learning in the homogeneous ensemble learning subgroup. In the random forest, each decision tree, or in other words, each base learner, has access to a random subset of feature vectors [6]. Therefore, the feature vector is defined as follows:

$$x = (x_1, x_2, \dots, x_p), \quad (1)$$

, where  $p$  is the dimension property of the available vector for the base learner. The main goal is to find the prediction function as  $f(x)$  that predicts the  $Y$  parameter. The prediction function is defined as follows:

$$L(Y, f(x)), \quad (2)$$

where  $L$  is known as the loss function, and the goal is to minimize the expected value of the loss. For regression applications and classification applications, squared error loss and zero-one loss are common choices, respectively. These two functions are defined as follows in equations (3) and (4), respectively.

$$L(Y, f(x)) = (Y - f(x))^2, \quad (3)$$

$$L(Y, f(x)) = I(Y \neq f(x)) = \begin{cases} 0, & \text{if } Y = f(x), \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

To create an ensemble, a set of base learners come together. If base learners are defined as follows:

$$h_1(x), h_2(x), \dots, h_j(x), \quad (5)$$

for regression applications, the averaging will be based on equation (6), and for classification applications, the voting will be based on equation (7).

$$f(x) = \frac{1}{J} \sum_{j=1}^J h_j(x), \quad (6)$$

$$f(x) = \arg \max \sum_{j=1}^J I(y = h_j(x)). \quad (7)$$

The Random Forest pseudocode for classification applications is shown in Algorithm 1.

As can be seen in Algorithm 1, in the random forest, an attempt is made to find a subset of features using the various replacements of training data and features that maximize the efficiency and accuracy of the output. This set of features is used to identify a new instance.

The following is a brief review of the random forest subject literature. It should be noted that we intend to introduce the background of the subject, and this paper is not a review paper, and the presented review is a brief review and does not mention all the previous works undoubtedly. However, the authors have tried to refer to the latest and most authoritative research work published in the recent years.

*2.1. Astronomy, Bioinformatics, and Economics fields.* In the astronomy field, Markel and Bayless [7] use RF for the classification of type IA and core-collapse supernovae. Chen et al. [8] propose an approach to detect the potential signal photons by RF. In the bioinformatics, Pang et al. [9] propose a method to mitigate the computational complexity of RNA simulation software by a typical random forest. Darmawan et al. [10] propose an age estimation model in the bioinformatics field. In the economics field, Park et al. [11] propose two stages of short-term load forecasting by random forest and deep neural networks to reduce energy costs. [12] use a typical RF to solve the e-commerce product classification problem. Modeling consumer credit risk by RF is the main goal of [13]. [14] increase tree correlation by controlling the probability of placing splits along with strong predictors to deal with high-dimensional settings. Sikdar et al. [15] proposed a variable selection method based on RF to identify the key predictors of price change in amazon.

*2.2. General and Global Problem fields.* In the general field, Giffon et al. [16] use the mean of orthogonal matching pursuit algorithms for calculating the weights of the linear combination for producing a linear combination of trees with minimum training error. Combining RF and generalized linear mixed models is the main idea of [17] to model clustered and longitudinal binary outcomes. Mohapatra et al. [18] optimize the random forest by use of unequal weight voting strategy. Ji et al. [19] propose a hybrid model for crowd counting by a combination of convolutional neural networks (CNN) and deep regression forest. Santra et al. [20] propose a deterministic dropout to remove unimportant connections in NN by RF. Proposing the oblique RF without explicit regularization techniques by minimizing the structural risk is the main goal of [21]. Katuwal et al. [22]

use an oblique hyperplane to split the data for increasing the accuracy of the trees and reduce the depth of RF. Probst et al. [23] tune the hyper-parameters to achieve higher performance to improve the RF. Kim et al. [24] propose a method for interpreting and simplifying a black-box model of a deep RF by quantifying the feature contributions and frequency of the fully trained deep RF. Jain et al. [25] propose dynamic weighing scheme for RF using the correlation between decision tree and data samples. In the global problem field, Stafoggia et al. [26] estimate daily particulate matter for weather forecasting by RF. Modeling the global forest area by RF is the main target of [27]. Breidenbach and Saravi [28] present research on land-subsidence spatial modeling and its assessment. Analyzing the net ecosystem carbon exchange is the goal of [29]. Prediction about the global climate problem using the index quantization ability of random forest and the optimizing ability of PSO in the NN prediction model is the main purpose of [30]. Li et al. [31] solve the class imbalance by detecting serial case pairs.

*2.3. Healthcare field.* Diagnosis detection and prediction of obesity in patients by RF are the main goals of [32, 33], respectively. El-Sappagh et al. [34] use RF in the simple form for the detection of Alzheimer's disease progression. In [35], RF is introduced as one useful machine learning tool for healthcare domain, especially for COVID-19 modeling. Khedkar et al. [36] use Patients Electronic Health Records for predicting the heart failure risks by RF. Hane et al. [37] propose a model for prediction of the dissolution behaviour of a wide variety of oxide glasses. Subudhi et al. [38] propose a method by RF to detect the ischemic stroke by a sequence of MRI images. Javadi et al. [39] propose a method to predict the immunogenic peptides of intracellular parasites. Identifying the key risk factors associated with acute rejection in organ transplantation is the main propose of [40]. In Singh et al. [41], RF has been used as one of the classifiers to classify the covid-19 spread. Na et al. [42] propose an automatic walking mode change of the above-knee prosthesis. Clustering and predicting vital signs by RF is the goal of [43]. Zhu et al. [44] optimize the parameters of the random forest by improved fish Swarm algorithm for predicting the knee contact force. A method for identifying foreign particles for quality detection of liquid pharmaceutical products is presented by [45]. Lee and Jung [46] consider the relation between teacher attachment and student growth. [47] propose a practical method for SIF downscaling. Guanter et al. [48] present a method based on RF for predicting diabetes. Subasi et al. [49] propose a decision support system for the diagnosis of migraine by RF. Classification of the driver's stress level is the main goal of [50]. Ayata et al. [51] propose an emotion recognition algorithm from multimodal physiological signals by using the random forest as one of the machine learning methods for recognition.

*2.4. Industrial and Network fields.* Zeraatpisheh et al. [52] use typical RF for producing the feature map in the industrial field. Du et al. [53] propose a rapid and accurate detection technique for pesticide detection by RF to

Let  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$  denote the training data, with  $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,p})^T$

For  $j = 1$  to  $J$ :

Take a bootstrap sample  $D_j$  of size  $N$  from  $D$ .

Using the bootstrap sample,  $D_j$  as the training data fit a tree.

(a) Start with all observations in a single node.

(b) Repeat the following steps recursively for each node until the stopping criterion is met: (i) Select  $m$  predictors at random from the  $p$  available predictors.

Find the best binary split among all binary splits in the predictors from step (i).

Split the node into two descendant nodes using the split from step (ii).

To make a prediction at a new point  $x$ .

$$\hat{f}(x) = \operatorname{argmax}_y \sum_{j=1}^J I(\hat{h}_j(x))$$

Where  $\hat{h}_j(x)$  is the prediction of the response variable at  $x$  using the  $j$ th tree.

ALGORITHM 1: The random forest pseudocode for classification applications [1].

construct a quantitative detection model. Improving the performance of mapping for mineral is the main goal of reference [54]. Liu et al. [55] propose an adaptive electrical period partition algorithm for open-circuit fault detection. Software fault prediction by ensemble techniques is investigated by [56]. In [57], the RF is used to build a distributed energy system. A comprehensive image processing model is proposed by [58]. Ho et al. [59] uses RF to propose a framework that uses climate data to model hydropower generation. Zhou et al. [60] use RF for small and unbalanced datasets to create a risk prediction model for decision-making tool. Deng et al. [61] propose an authentication method for protecting high-value food products by RF. The forecast for agricultural products by RF is proposed by [62]. Jeong and Kim [63] use weighted random forest for the link prediction model. Khorshidpour et al. [64] present an approach to model an attack against classifiers with non-differentiable decision boundary. Fusing multi-domain entropy and RF is the main goal of [65] for proposing a fault diagnosis method of the inter-shaft bearing. Analyzing the wine quality is presented by [66]. In the network field, Madhumathi and Suresh [67] develop a model to predict the future location of a dynamic sensor node in wireless communications. Fang et al. [68] propose an encrypted malicious traffic identification method. Detecting the intrusion in the network by typical RF is proposed by [69], and intrusion detection in the network security by tuning the RF parameter of the Moth-Flame optimization algorithm is presented by [70].

*2.5. Physics, Text Processing, Tourism, and Urban Planning fields.* In the physics field, Mingjing [71] measure and quantify the pH of soil by RF. [72] propose a model for extracting complex relationships between energy modulation and device efficiency. Zhang et al. [73] propose a model to accurately and effectively predict the UCS of LWSCC by a beetle antennae search algorithm for tuning the hyper-parameters of RF. The prediction of geotechnical parameters by typical RF is made by [74]. Creep index prediction by the RF algorithm to determine the optimal combination of variables is the main goal of [75]. In the text processing field, the comparison between RF and other

classifiers is presented by [76] for finding the best classifiers in the subject literature of text classification. The random forest is used as one of the base learners of the ensemble model for fake news detection by [77]. Analyzing the reviewer's comment for sentiment analysis is the main goal of [78]. Zhang et al. [79] propose two novel label flipping attacks to evaluate the robustness of NB under noise by random forest. Recognizing newspaper text by RF is done by [80]. Madichetty and Sridevi [81] use RF as one of the classifiers for detecting the damage assessment tweets. Madasu and Elango [82] use the typical RF for feature selection for sentiment analysis. Chang et al. [83] use online customer reviews for opinion mining by RF. Text classification by simple RF is the goal of [84]. Onan and Toçouglu [85] present a method for document clustering and topic modeling on massive open online courses. Sentiment analysis of technical words in English by the Gini index for feature selection is done by [86]. Beck [87] uses ensemble learning and deep learning for sentiment classification scheme with high predictive performance in massive open online courses' reviews. Onan [88] present a deep learning based approach to sentiment analysis. This approach uses TF-IDF weighted Glove word embedding with CNN LSTM architecture. Onan and Tocoglu [89] present an effective sarcasm identification framework on social media data by pursuing the paradigms of neural language models and deep neural networks. In the tourism field, Rodriguez-Pardo et al. [90] propose a method based on simple RF for predicting the behaviour of tourists. Predicting the travel time to reduce traffic congestion is the main goal of [91]. Jamatia et al. [92] propose a method for tourist destinations' prediction. In urban planning, Baumeister et al. [93] rank the urban forest characteristics for cultural ecosystem services supply by typical RF. Forecasting road traffic conditions is done by [94]. The simulation of urban space development by RF is presented by [95]. Investigating the information on a gross domestic product for the analysis of economic development is presented by [96]. Mei et al. [97] propose a method to identify the spatiotemporal commuting patterns of the transportation system. In this brief review, the mentioned references are categorized in terms of innovation and functionality.

As can be seen from Table 1, RF has a high range of applications and variations in scope. In contrast, both in



TABLE 1: Brief review of RF literature on functionality and innovation.

Type	Field	Paper
Functionality	Astronomy	[7], [8]
	Bioinformatics	[9], [10]
	Economics	[11], [12], [13]
	Global problem	[26], [27], [28]
	Healthcare	[32], [33], [34], [35], [36], [41], [98], [37], [39], [40], [42], [43], [45], [46], [47], [48], [49], [50], [51],
	Industrial	[52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62]
	Network	[63], [67], [68], [69], [99], [100]
	Physics	[71], [72]
	Text processing	[76], [77], [78], 80 [81], [82], [83], [84]
	Tourism	[91], [92]
Innovative method	Urban planning	[93], [94], [95], [96], [97]
	Economics	[14], [15]
	General	[16], [17], [18], [19], [101], [21], [22], [23], [24], [25]
	Global problem	[30], [31]
	Healthcare	[44]
	Industrial	[65]
	Network	[64]
	Physics	[73], [75]
	Text processing	[79], [86]

terms of quantity and quality, their innovations are often limited to set various parameters, and there is no significant innovation in the base learner combinations.

### 3. Learning Automata

Learning Automata (LA) is one of the learning algorithms that, after selecting different actions at different times, identify the best practices in terms of responses received from a random environment. LA selects an action from the set of actions in the vector of probabilities, and this action is evaluated in the environment. By using the received signal from the environment, the LA updates the probability vector and, by repeating this process, the optimal action is gradually identified. The classification problem can be formulated as a team of LA that operates collectively to optimize an objective function [102]. In Figure 1, the interaction of the learning automata and the environment is shown.

Finding the global optimum in the solution space is another advantage of using the LA. The LA can be formally represented by the quadruple

$$LA = \{\alpha, \beta, P, T\}, \quad (8)$$

in which

$$\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\} \quad (9)$$

is the set of actions (outputs) of the LA; in other words, the set of inputs of the environment.

$$\beta = \{\beta_1, \beta_2, \dots, \beta_r\}, \quad (10)$$

is the set of inputs of the LA; in other words, the set of outputs of the environment.

$$P = \{p_1, p_2, \dots, p_r\}, \quad (11)$$

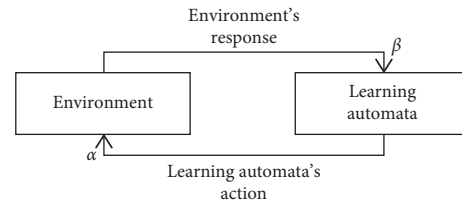


FIGURE 1: Interaction of learning automata with the environment.

is the probability vector of the LA actions and

$$P(n+1) = T[P(n), \alpha(n), \beta(n)], \quad (12)$$

is the learning algorithm.

In LA, three different models can be defined in the environment. In the P-Model, the environment presents the values of 0 or 1 as the output. In the Q-Model, the output values of the environment are discrete numbers between 0 and 1. In the S-Model, the output of the environment is the continuous value between 0 and 1. The selected actions by the LA are updated by both the signal received from the environment and using reward and penalty functions. The amount of allocated reward and penalty to the LA action can be defined in four ways: LRP, where the number of rewards and penalties are considered the same; LR&P in which the amount of penalty is several times smaller than the reward; LRI in which the penalty amount is considered 0; and LIP, where the reward amount is considered 0 [103].

At each instant  $n$ , the action probability vector  $\pi(n)$  is updated by the linear learning algorithm given in equation (13) if the chosen action  $a_i(k)$  is rewarded by the environment, and it is updated according to equation (14) if the chosen action is penalized [104].

$$\begin{cases} p_i(n+1) = p_i(n) + a[1 - p_i(n)], \\ p_j(n+1) = (1-a)p_j(n), \quad \forall j, j \neq i, \end{cases} \quad (13)$$

$$\begin{cases} p_j(n+1) = (1-b)p_j(n), \\ p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n), \quad \forall j; j \neq i, \end{cases} \quad (14)$$

, where “ $a$ ” is the reward parameter, “ $b$ ” is the penalty parameter, and “ $r$ ” is the number of actions. The authors applied the LA in the proposed method, because:

- (i) The LA presents an acceptable performance in uncertain situations.
- (ii) The LA does search action in the probability space.
- (iii) The LA requires simple feedback from the environment to optimize its state.
- (iv) Since the LA has a simple structure, it has a simpler implementation in both software and hardware.
- (v) The LA is not constrained to use accuracy criteria for optimization usage.
- (vi) The LA is applicable in real-time usage since the LA is not involved with light computational complexity [105].

#### 4. Proposed Method

The random forest is one of the methods of ensemble learning that all constructor classifiers are same type (i.e., decision tree). Therefore, the random forest is a homogeneous ensemble learning method. In this article, we intend to use the idea of reinforcement learning to increase the efficiency of random forest and add the ability to adapt to the conditions of the problem for this data mining algorithm. The details of the proposed method are described below.

The method proposed in this paper is based on the idea of reinforcement learning, and it employs the learning automata to implement the idea. The learning automata is the core of the proposed method, and by receiving feedback from the environment for each action, it updates the probability selection of the actions. In the proposed method, each base learner, all of which are decision tree, are considered as learning automata actions.

In the proposed method, the training data are first randomly divided into  $N$  sections. In this division,  $N$  corresponds to the number of trees we want to have in the forest. Unlike the random forest, in which the predictive model works by averaging or voting between trees, in the proposed method, the predictive model is created using learning automata, which forms the core of the algorithm. The block diagram of the proposed method is shown in Figure 2.

The preprocessing step in the proposed method is a general step, and based on what type of data the processing area is dealing with, the details of this phase are determined. In the proposed method, at first, similar to the random forest method, the training data are divided into the number of

base learners and randomly injected into the base learners. The difference between this step and the similar step in the random forest is that all the features in the feature vector are given to all base learners, and the feature replacement option is removed.

After the first run, the prediction models are created in the base learners and placed in a pool that is actually an interactive environment with the learning automata. The results obtained from the base learners for each new sample are given in the form of a reinforcement signal to the learning automata, which we know as the primary feedback of the environment. Depending on whether the received reinforcement signal is a reward or a penalty, the chances of selecting each of the base learners, -which they are the actions of the learning automata - are updated. It should be noted that the initial probability of selecting these actions is considered equal at the start. If we have  $R$  base learners to form the ensemble, the probability of the initial selection of each of them is equal to

$$p(DT_r) = (1/R). \quad (15)$$

It is clear that the sum of the probabilities of all actions will be equal to 1.

$$\sum_{i=1}^R (pDT) = 1. \quad (16)$$

The initial probability of selecting actions is considered equal because all of them are homogeneous in terms of separating power.

In the proposed method, integration of the base learners is performed by the LA. Therefore, for each input in the test set, a linear LA is defined, and the action of each LA corresponds to selecting the base learners. The process of running base learners and receiving feedback from the environment continues until the probability of selecting actions converges to one of the base learners, or the number of repetitions for learning automata exceeds the pre-determined limit. Once the probability of selections converges, then the result of the base learner, which has the highest probability of selection, is determined as the result of the ensemble for that particular input. In such a case, finding the global optimal is guaranteed by the algorithm, and because all the features in the feature vector are examined, rare modes are also covered, and the ability to adapt to the conditions of the problem space and independence from the domain is stabilized. In the proposed method, the random selection of subsets causes interdependence between trees. The depth of all the decision trees in the proposed method is considered equal. Each decision tree divides the training data differently at the leaf level. The pseudocode of the proposed method is shown in Algorithm 2.

In the learning automata block in Figure 2, there are two functions called the reward function and penalty function. Activation of one of these two functions is based on the type of reinforcement signal received from the environment. The received signal from the environment determines whether the result of the base learner activity or the selected action in the previous step was useful or not. If the result is useful, that

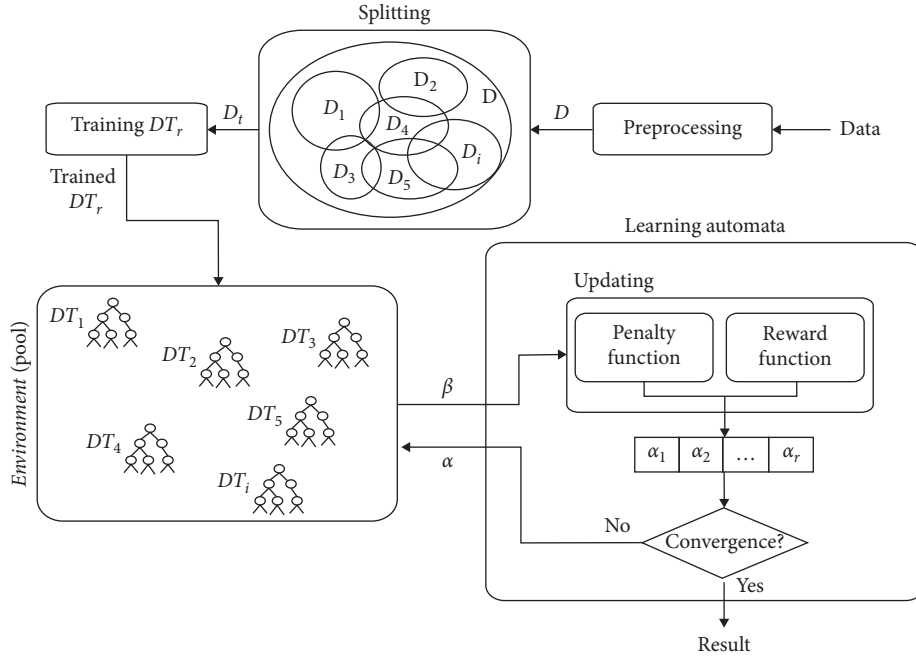
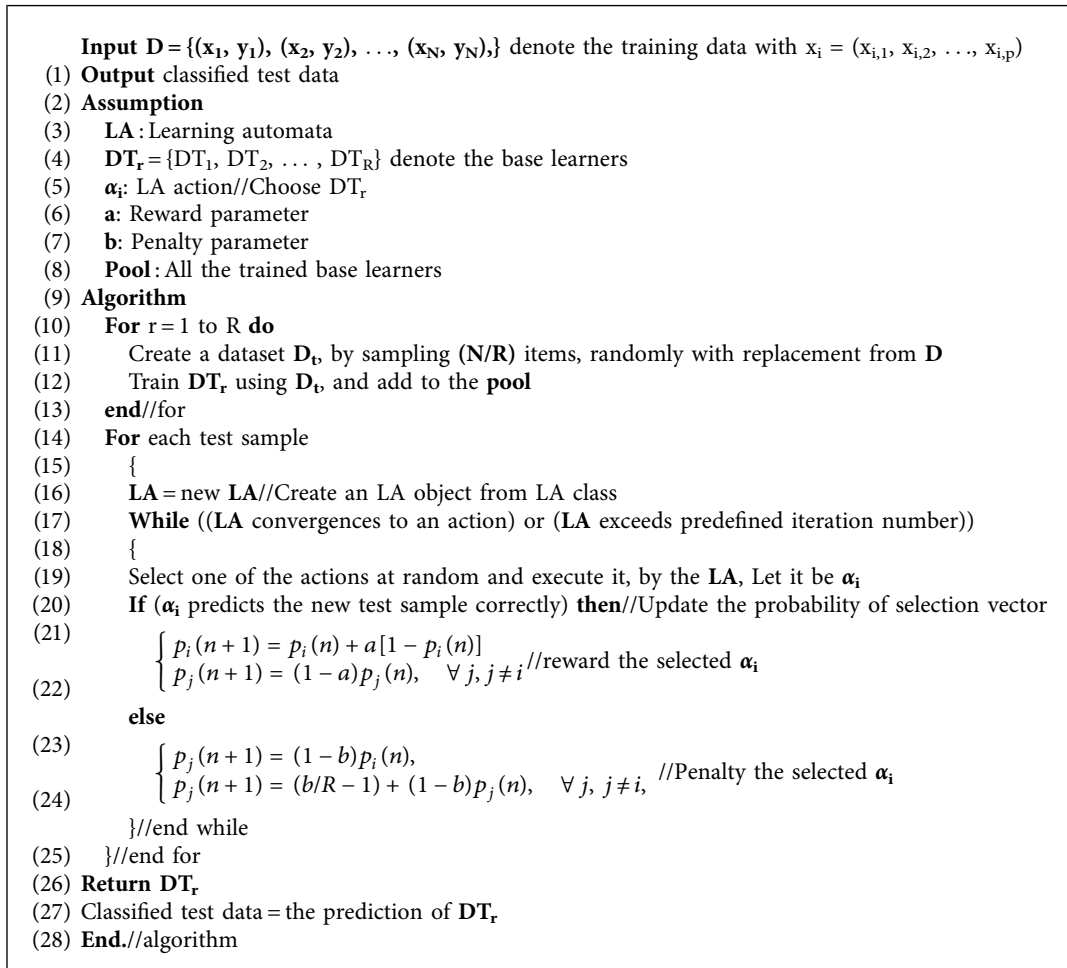


FIGURE 2: The block diagram of the proposed method.



ALGORITHM 2: The pseudocode of the proposed method.

action must be rewarded or, in other words, increase the probability of its selection. The increase in the probability of the selected action is determined by the parameters “a” and “b,” which are called the reward parameter and the penalty parameter, respectively.

To comply with (16), that is, the sum of the probabilities of all actions being equal to one, the probability of all other actions is reduced according to the size of the parameter “a.” If the result of the selected action is not useful, that action must also be penalized. In other words, the probability of that action must be reduced. To do this, the probability of selecting that action is reduced to the size of parameter “b,” and as a rewarding mode, and to observe (16), the probability of selecting other actions is increased by the size of the parameter “b.”

In the proposed method, the learning automata model environment is assumed to be the P-Model, where the environment defines zero and one values as outputs. Zero means reward, and one means penalty. If the correct answer is received from the selected base learner by the LA, the action of choice will be rewarded; otherwise, it will be penalized.

## 5. Evaluation

In order to thoroughly evaluate the efficiency of the proposed method, in this section, the details of the evaluation of the proposed method are presented separately from the data used and the experimental results.

*5.1. Datasets.* In order to evaluate the proposed method and to create an environment with the dynamic behaviour of data, different domains of applications have been selected. As mentioned in the previous sections, dynamic behaviour refers to the different results that an instance exhibits in different environmental conditions. Variety in the results of different environments is created by a specific domain. Text data are one of the most well-known types of data that exhibit such dynamic behaviour. In other words, these types of data are one of the optimal options for creating a dynamic environment, which proves the adaptability of the proposed method. The details of the selected data for the evaluation phase are shown in Table 2.

*5.2. Experimental Result.* In order to evaluate the proposed method, eighteen datasets in different domains introduced in the previous section have been used. In the literature on learning automata, different modes have been considered for tuning learning automata; in this paper, three modes have been used to evaluate the proposed method. The LIP mode is not considered due to poor results. The evaluation results of each of the LRI, LR $\epsilon$ P, and LRP modes are shown in separate figures. In order to determine the optimal value for the reward and penalty parameters, six text datasets have been selected. The reason for this choice is the high diversity in the behaviour of textual data as well as a large number of samples and a large number of features of these six datasets. In the LRI mode, the value of the penalty parameter is

considered to be zero, and the results of the proposed method in this mode are shown in Figure 3.

Based on the literature on learning automata in the LR $\epsilon$ P mode, the value of the penalty parameter is considered to be much smaller than the value of the reward parameter. The results of the proposed method are shown in the LR $\epsilon$ P mode in Figure 4.

As mentioned in the learning automata section, in the LRP mode, the values of the penalty and reward parameters are considered equal. The results of the proposed method in this mode are also shown in Figure 5.

A comparison of the results obtained from the implementation of the proposed method in three adjustable modes for learning automata shows that the settings on the LRP mode have resulted in the highest accuracy for identification. Then there are LR $\epsilon$ P and LRI modes. In the LR $\epsilon$ P mode, the setting  $a=0.01$ ,  $b=0.01$  is not considered, because these values are equal to the first values set in the LRP mode, and in order to prevent duplication of results in different tables, these settings have been removed from the LR $\epsilon$ P mode. For this reason, the number of experiments performed on LR $\epsilon$ P mode evaluations is one less than the other two. Considering that the settings of reward and penalty parameters in the LRP mode with the values of  $a=0.5$ ,  $b=0.5$  have resulted in the highest efficiency, evaluation has been done on other datasets with these settings. A comparison of the proposed method and similar approaches in the subject literature is shown in Table 3.

As can be seen in Table 3 from the point of view of accuracy, the proposed method offers better performance than the methods available in the subject literature, which indicates an improvement in the aggregation model of the base learners. This improvement is due to the use of reinforcement learning ideas of the method of aggregation of basic classifiers, which is known as base learner. The use of reinforcement learning ideas has improved the ability of the created ensemble, and it improved the ability to address issues in which data exhibit dynamic behaviour. The results of experiments performed on different data confirm the capabilities added to the random forest by the proposed method. As mentioned earlier, in the field of opinion mining, the type of text data is the most obvious data that exhibit such dynamic behaviour. Therefore, the optimal values for the reward and penalty parameters have been determined in these types of data, and these settings have been used for other types of data.

In addition to the accuracy criterion, other statistical criteria have been examined to evaluate the proposed method. As can be seen in Table 4, the proposed method has shown better results in both positive and negative classes than the methods available in the literature. Among the statistical criteria, Precision (P) determines the exactness of the results obtained from the classifier, and Recall (R) determines the completeness of the results obtained from the classifier. The results obtained from the test in the mentioned statistical criteria show that the proposed method has a high performance.

TABLE 2: Details of textual data used for evaluation.

Domain	Name	# Feature	# Instance
Text	Stanford—Sentiment 140 corpus [106]	Bag of word	160000
	Large dataset of movie reviews [107]	Bag of word	50000
	Sentence polarity dataset v1.0 [108]	Bag of word	10662
	Internet movie database [105]	Bag of word	1400
	Yelp review [105]	Bag of word	598000
	Amazon review [105]	Bag of word	1000000
Healthcare	Heart disease dataset [105]	13	200
	Breast cancer dataset [105]	30	569
	Arrhythmia dataset [105]	279	454
	Parkinson dataset [105]	45	241
	Caesarean section dataset [105]	5	81
	Gene expression dataset [105]	255	801
Physical	Diabetes dataset [105]	7	765
	Statlog (heart) dataset [105]	13	271
	Ionosphere dataset [105]	34	352
	Sonar, mines vs. rocks dataset [105]	60	208
Sound	Voice dataset [105]	20	3168
	Emotions from music dataset [105]	28	592

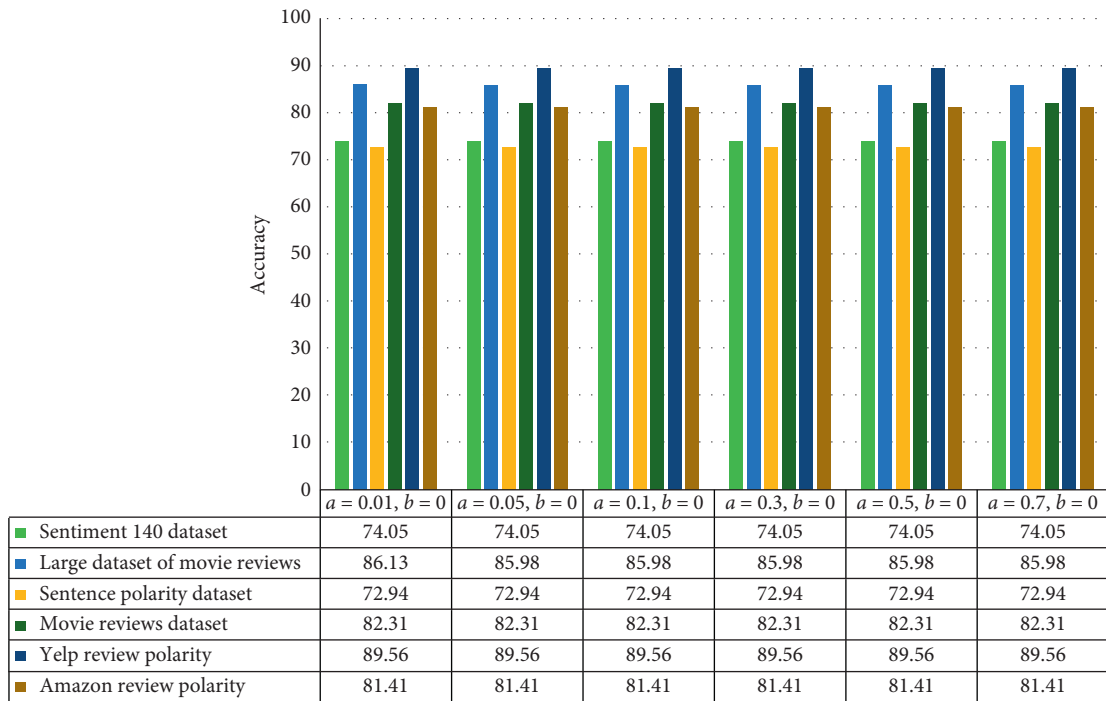


FIGURE 3: The results of the proposed method in LRI mode.

## 6. Discussion

In this section, more details of the proposed method are explained along with the reasons for the need to address these details. These include the details of the preprocessing step, tuning the learning automata parameters, as well as

ranking the set of these parameters based on their performance.

*6.1. Preprocessing.* As explained in the proposed method section, the preprocessing step is a general step. In order for

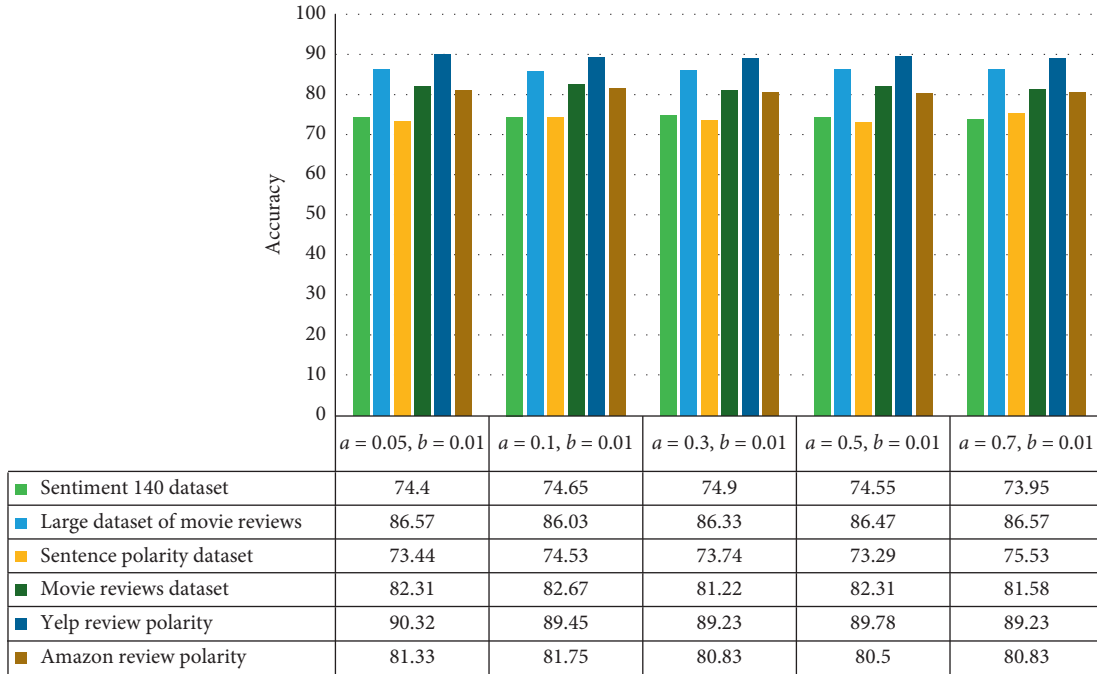


FIGURE 4: The results of the proposed method in the LR&amp;P mode.

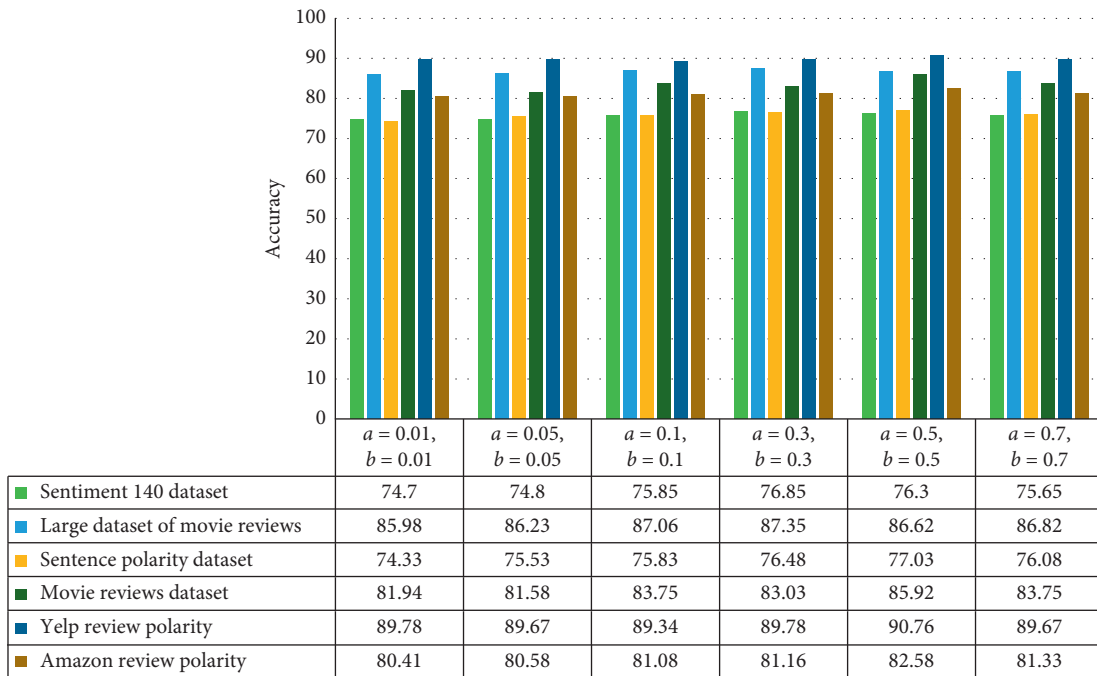


FIGURE 5: The results of the proposed method in the LRP mode.

the evaluation, different data from different domains were examined. The preprocessing of textual data, along with the relevant details, is described below. It should be noted that preprocessing for other types of data, such as feature extraction, feature selection, normalization, noise removal, and other related preprocessing, has not been performed because all of them are taken as clean data from the UCI

Repository [109]. And their basis for accuracy is based on previous research works that have used these data.

In order to prepare textual data for the main process, the opinion mining domain is selected and the related preprocessing is as follows. The details of the preprocessing step for text data in opinion mining are shown in Figure 6.

TABLE 3: Comparison of the proposed method with similar approaches in the subject literature.

	Dataset	Averaging	Majority Voting	Random Forest	Our Method
Text	Sentiment140 dataset	74.54	75.50	74.30	<b>76.30</b>
	Large dataset of movie reviews	86.28	86.86	86.42	<b>86.62</b>
	Sentence polarity dataset	73.75	74.63	73.38	<b>77.03</b>
	Movie reviews dataset	81.58	81.58	81.67	<b>85.92</b>
	Yelp review polarity	89.47	90.32	89.74	<b>90.76</b>
	Amazon review polarity	80.86	81.66	80.97	<b>82.58</b>
Healthcare	Heart disease dataset	58.00	57.50	57.50	<b>65.00</b>
	Breast cancer dataset	97.41	97.36	96.49	<b>98.24</b>
	Arrhythmia dataset	80.71	85.71	81.31	<b>85.71</b>
	Parkinson dataset	63.95	64.58	64.58	<b>68.75</b>
	Caesarean section dataset	60.31	62.50	43.75	<b>68.75</b>
	Gene expression dataset	95.59	95.62	96.27	<b>98.75</b>
Physical	Diabetes dataset	75.77	75.32	74.67	<b>76.62</b>
	Statlog (heart) data set	81.20	81.48	79.62	<b>85.18</b>
	Ionosphere dataset	91.05	91.54	92.95	<b>95.77</b>
Sound	Sonar, mines vs. rocks dataset	85.23	85.71	73.80	<b>88.09</b>
	Voice dataset	76.38	76.18	76.49	<b>88.95</b>
	Emotions from music dataset	78.23	78.15	82.35	<b>84.03</b>

TABLE 4: Comparison of statistical criteria.

Method	Positive class			Negative class			Method	Positive class			Negative class		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)		P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
<i>Sentiment140 dataset</i>							Parkinson dataset						
MV	72.44	70.35	71.38	70.90	72.96	71.92	MV	69.23	66.67	67.92	<b>59.09</b>	61.9	60.47
RF	72.44	74.58	73.49	76.46	74.43	75.43	RF	69.23	66.67	67.92	<b>59.09</b>	61.9	60.47
OM	<b>75.20</b>	<b>80.36</b>	<b>77.69</b>	<b>81.17</b>	<b>76.16</b>	<b>78.59</b>	OM	<b>76.92</b>	<b>68.97</b>	<b>72.73</b>	<b>59.09</b>	<b>68.42</b>	<b>63.41</b>
<b>Large dataset of movie reviews</b>							<b>Caesarean section data set</b>						
MV	87.50	87.33	87.41	87.41	87.59	87.50	MV	55.56	71.43	62.5	<b>71.43</b>	55.56	62.5
RF	85.83	76.16	80.70	73.37	83.93	78.29	RF	33.33	50	40	57.14	40	47.06
OM	<b>87.80</b>	<b>87.97</b>	<b>87.88</b>	<b>88.10</b>	<b>87.93</b>	<b>88.01</b>	OM	<b>66.67</b>	<b>75</b>	<b>70.59</b>	<b>71.43</b>	<b>62.5</b>	<b>66.67</b>
<b>Sentence polarity dataset</b>							<b>Gene expression dataset</b>						
MV	75.63	72.83	74.20	72.80	75.61	74.18	MV	92.31	94.12	93.2	97.25	96.36	96.8
RF	74.62	67.78	71.08	65.95	72.94	69.27	RF	94.23	93.23	94.23	97.25	97.25	97.25
OM	<b>76.04</b>	<b>73.29</b>	<b>74.64</b>	<b>73.29</b>	<b>76.04</b>	<b>76.64</b>	OM	<b>98.08</b>	<b>98.08</b>	<b>98.08</b>	<b>99.08</b>	<b>99.08</b>	<b>99.08</b>
<b>Movie reviews dataset</b>							<b>Diabetes dataset</b>						
MV	83.10	82.52	82.81	81.84	82.09	81.78	MV	90.29	76.86	83.04	45.1	69.7	54.76
RF	77.46	71.43	74.32	76.41	73.98	70.54	RF	90.29	76.23	82.67	43.14	68.75	53.01
OM	<b>84.51</b>	<b>83.33</b>	<b>83.92</b>	<b>82.22</b>	<b>83.46</b>	<b>82.84</b>	OM	<b>91.26</b>	<b>77.69</b>	<b>83.93</b>	<b>47.07</b>	<b>72.73</b>	<b>57.14</b>
<b>Yelp review polarity</b>							<b>Voice dataset</b>						
MV	<b>90.85</b>	88.04	88.42	87.11	<b>90.11</b>	88.59	MV	64.58	76.09	69.86	84.85	76.24	80.31
RF	80.64	85.94	83.21	86.22	81.00	83.53	RF	64.58	76.75	70.14	85.4	76.35	80.62
OM	90.00	<b>89.43</b>	<b>89.71</b>	<b>88.89</b>	89.49	<b>89.19</b>	OM	87.82	86.55	87.18	89.81	90.81	90.3
<b>Amazon review polarity</b>							<b>Emotions from music dataset</b>						
MV	82.68	79.45	81.03	<b>79.38</b>	82.62	80.97	MV	83.58	78.87	81.16	71.15	77.08	74
RF	79.12	74.56	76.77	73.98	87.61	76.22	RF	88.06	<b>81.94</b>	84.89	<b>75</b>	82.98	78.79
OM	<b>83.70</b>	<b>79.52</b>	<b>81.56</b>	79.21	<b>83.45</b>	<b>81.28</b>	OM	<b>92.54</b>	81.58	<b>86.71</b>	73.08	<b>88.37</b>	<b>80</b>
<b>Heart disease dataset</b>							<b>Sonar, mines vs. Rocks dataset</b>						
MV	<b>61.90</b>	59.09	60.47	52.63	55.56	54.05	MV	<b>87.5</b>	<b>87.5</b>	<b>87.5</b>	<b>83.33</b>	<b>83.33</b>	<b>83.33</b>
RF	<b>61.90</b>	59.09	60.47	52.63	55.56	54.05	RF	79.17	76	77.55	66.67	70.59	68.57
OM	<b>61.90</b>	<b>68.42</b>	<b>65.00</b>	<b>68.42</b>	<b>61.90</b>	<b>65.00</b>	OM	<b>87.5</b>	<b>87.5</b>	<b>87.5</b>	<b>83.33</b>	<b>83.33</b>	<b>83.33</b>
<b>Breast cancer data set</b>							<b>Statlog (heart) data set</b>						
MV	95.74	97.83	96.77	<b>98.51</b>	97.06	97.78	MV	<b>79.19</b>	79.19	79.19	83.33	83.33	83.33
RF	<b>97.87</b>	93.88	95.83	95.52	98.46	96.97	RF	75	78.26	76.6	83.33	80.65	81.67
OM	<b>97.87</b>	<b>97.87</b>	<b>97.87</b>	<b>98.51</b>	<b>98.51</b>	<b>98.51</b>	OM	<b>79.19</b>	<b>86.36</b>	<b>82.61</b>	<b>90</b>	<b>84.38</b>	<b>87.1</b>
<b>Arrhythmia data set</b>							<b>Ionosphere data set</b>						
MV	88.37	<b>82.61</b>	85.39	<b>83.33</b>	88.89	86.02	MV	82.14	95.83	88.46	97.67	89.36	93.33
RF	83.72	78.26	80.9	79.17	79.74	73.85	RF	85.71	96	90.57	97.67	91.3	94.38
OM	<b>93.02</b>	80	<b>86.02</b>	79.19	<b>92.68</b>	<b>85.39</b>	OM	<b>89.29</b>	<b>100</b>	<b>94.34</b>	<b>100</b>	<b>93.48</b>	<b>96.63</b>

P, R, and F1 refer to Precision, Recall, and F1-score. MV: majority voting, RF: random forest, and OM: our method.

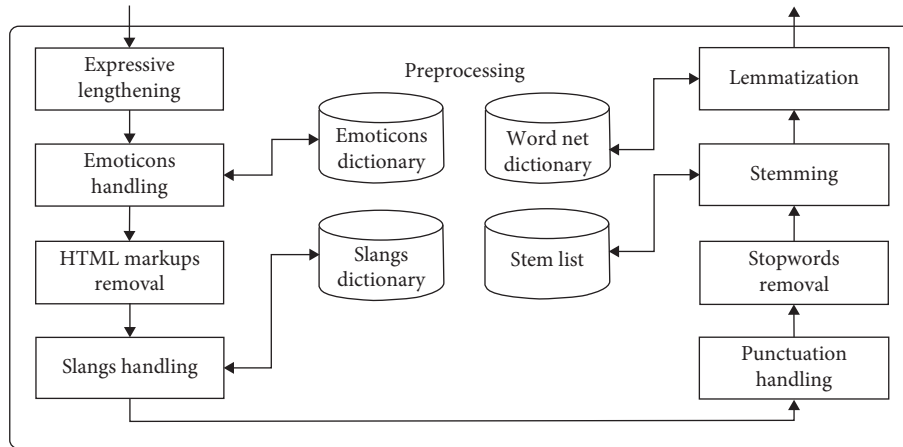


FIGURE 6: Details of the preprocessing step for text data.

*Expressive Lengthening.* Word lengthening or word stretching refers to the words that are elongated to express a particular emotion strongly, and the words with wrong spellings are corrected and replaced with their original words.

*Emoticons Handling.* It refers to the emoticons mentioned in the text that are replaced with their meaning, which makes it easier to analyze the emoticons.

*HTML Markups Removal.* HTML markups presented in the text are removed as they do not have any sentimental value attached to it.

*Slangs Handling.* The slangs are used for writing a given word, in short syllables, which depict the same meaning but save the time of typing. In slangs handling, the slangs presented in the text are replaced with their original words.

*Punctuation Handling.* Punctuations are used in a text to separate sentences and their elements, and to clarify their meaning. At punctuation handling, once the apostrophes are handled, all the remaining punctuations and numbers are removed.

*Stopwords Removal.* Stopwords do not carry much meaning and have no importance in the text. Stopwords are removed to get a simplified text.

*Stemming.* It refers to finding out the root or stem of a word. Removing various suffixes to reduce the number of words is the purpose of stemming.

*Lemmatization.* It returns the base or dictionary form of a word, which is known as the lemma. It is very similar to stemming, but it is more akin to synonym replacement.

*BoW creation.* The bag of word creation is the latest preprocess that is performed on the text preparation.

**6.2. Tuning the Parameters of Reward and Penalty.** In the subject literature of the learning automata, three different modes have been defined to tune the parameters of reward and penalty. In the proposed method, in which the idea of

reinforcement learning is implemented using learning automata, all three adjustable modes of the parameters of reward and penalty are examined. The results of these three modes were presented in the experimental result section. In this paper, Friedman test statistical verification is used to determine which mode and which settings are best adjustable for the reward and penalty parameters. The values set for parameters “*a*” and “*b*” are shown in Table 5. Determining the numerical value of these parameters is based on the subject literature of learning automata. Of course, a wide variety of values can be considered for these two parameters. In this paper, an attempt has been made to tune the parameters in such a way that all the modes are considered so that they can be used to prove the efficiency of the proposed method compared to the previous methods.

**6.3. Ranking.** Friedman test statistical verification [110] is a ranking method that, the difference between the ranks assigned to each of the input samples, determines the optimal level of each option. In this paper, this verification method has been used to determine the optimal value of reward and penalty parameters as well as to compare the proposed method with the conventional methods in the subject literature of ensemble learning. The results are shown in Table 6.

As can be seen in Table 6, there is a significant difference between the rankings of the proposed method and the rankings of the traditional methods, which indicate an improvement in the efficiency of the proposed method compared to other methods. Among the three modes considered for tuning reward and penalty parameters, it is observed that the rankings have increased in LRI, LReP, and LRP modes, respectively. In the LRP mode, where the values of the reward and penalty parameters are considered the same, the highest efficiency is also observed. There is a significant difference between the Mean Rank of the best set of the reward and penalty parameters in the proposed method and this rank in the random forest method. The difference between the ranks is proof that the proposed method is optimal versus the traditional methods of



TABLE 5: Numerical values tuned for reward and penalty parameters.

Mode	Parameter						
LRI	$a$	0	0.1	0.1	0.3	0.5	0.7
	$b$	0	0	0	0	0	0
LReP	$a$	0.1	0.1	0.3	0.5	0.7	
	$b$	0	0	0	0	0	
LRP	$a$	0	0.1	0.1	0.3	0.5	0.7
	$b$	0	0.1	0.1	0.3	0.5	0.7

TABLE 6: Friedman test statistical verification results for ranking the parameters of reward and penalty and comparing the proposed method with the literature.

Method	Tuning	Mean rank	Final rank
LRP	$a = 0.5, b = 0.5$	19.17	1
LRP	$a = 0.3, b = 0.3$	16.83	2
LRP	$a = 0.7, b = 0.7$	15.58	3
MV	Majority voting	14.67	4
LRP	$a = 0.1, b = 0.1$	13.92	5
LReP	$a = 0.05, b = 0.01$	12.17	6
LReP	$a = 0.1, b = 0.01$	11.83	7
LReP	$a = 0.5, b = 0.01$	10.08	8
LRP	$a = 0.05, b = 0.05$	9.58	9
RF	Random forest	9.17	10
LRP	$a = 0.01, b = 0.01$	8.75	11
LIR	$a = 0.01, b = 0$	8.42	12
LIR	$a = 0.05, b = 0$	7.67	13
LIR	$a = 0.1, b = 0$	7.67	13
LIR	$a = 0.3, b = 0$	7.67	13
LIR	$a = 0.5, b = 0$	7.67	13
LIR	$a = 0.7, b = 0$	7.67	13
AV	Averaging	7.58	14
LReP	$a = 0.3, b = 0.01$	7.17	15
LReP	$a = 0.7, b = 0.01$	6.75	16

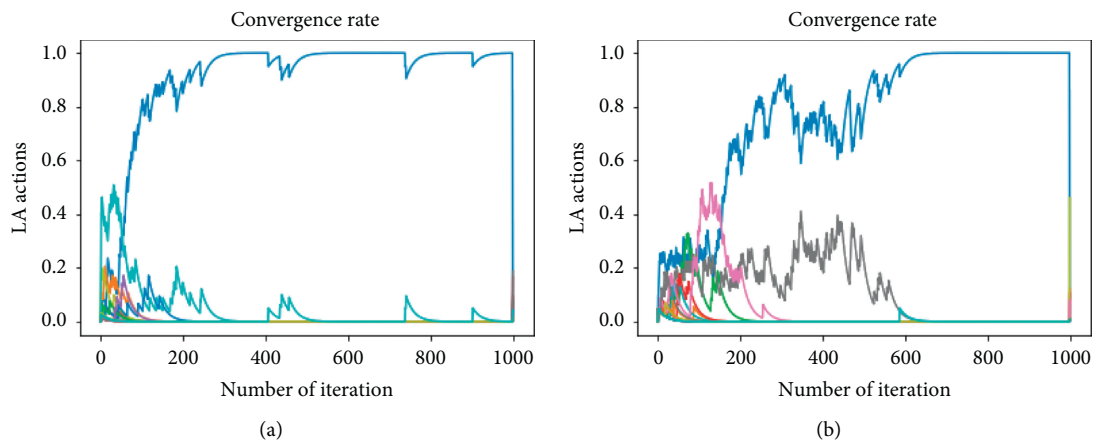


FIGURE 7: Continued.

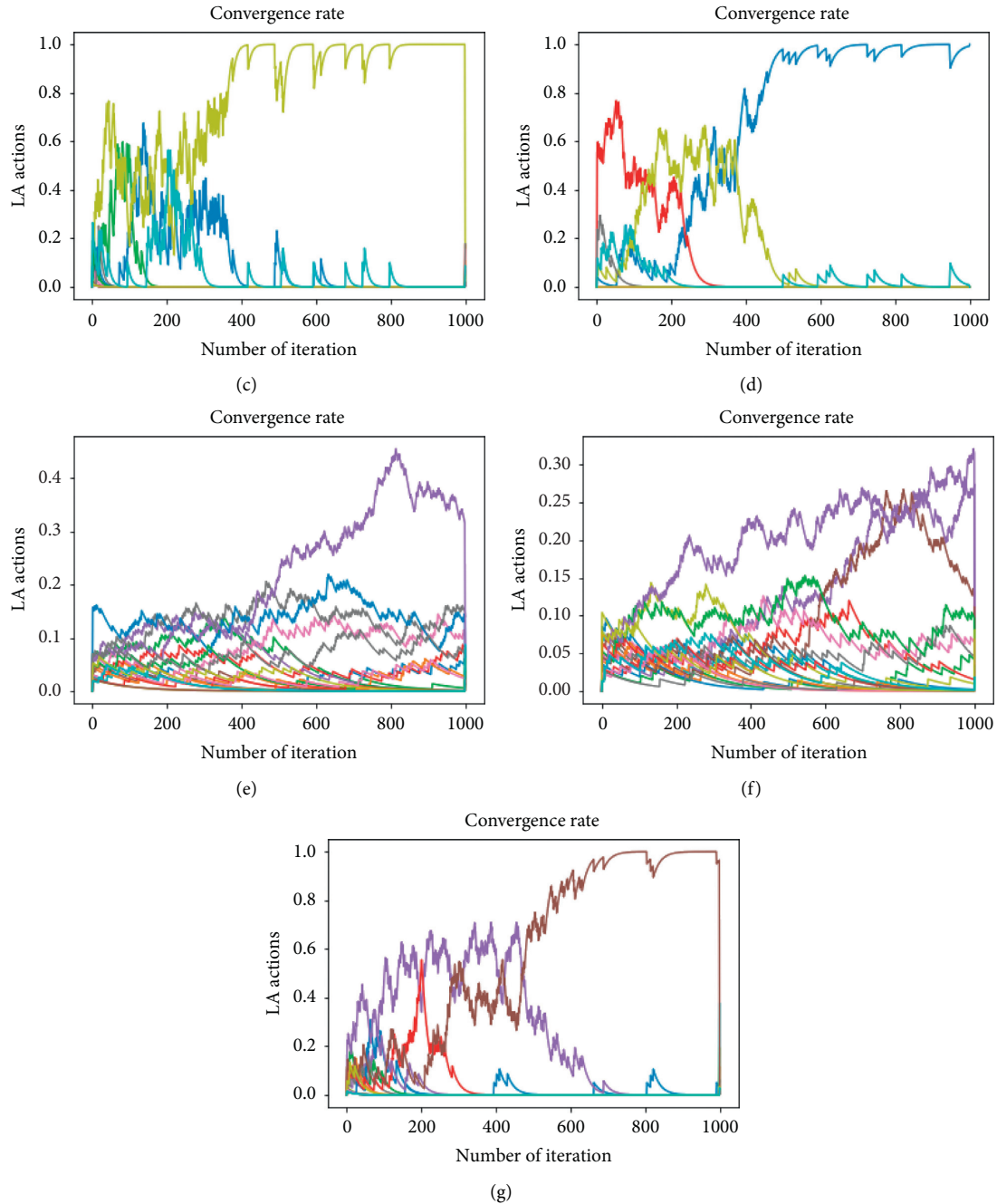


FIGURE 7: Convergence rate for different reward and penalty parameters. (a)  $a = 0.5, b = 0.5$ ; (b)  $a = 0.3, b = 0.3$ ; (c)  $a = 0.7, b = 0$ ; (d)  $a = 0.1, b = 0.1$ ; (e)  $a = 0.01, b = 0$ ; (f)  $a = 0.05, b = 0.05$ ; (g)  $a = 0.3, b = 0$ .

aggregating classifiers to achieve a strong classification method.

**6.4. Checking Convergence Rate.** To more accurately address the proposed method in terms of efficiency, LA convergence has been investigated. Figure 7 shows the convergence of LA actions for different amounts of reward and penalty variables. In most of the different settings for these two parameters, the convergence rate is high, and convergence to one of the actions usually occurs before reaching a certain

number of iterations. As shown in Table 5, convergence at a lower rate occurred in some of the other settings that scored lower on the Friedman test.

**6.5. Noise Resistance.** In order to more accurately evaluate the proposed method and determine the resistance of the proposed method to noise, another evaluation has been performed on the data presented in the previous section. This evaluation was performed by injecting 20% noise into clean data. The results of the evaluation on noisy data show

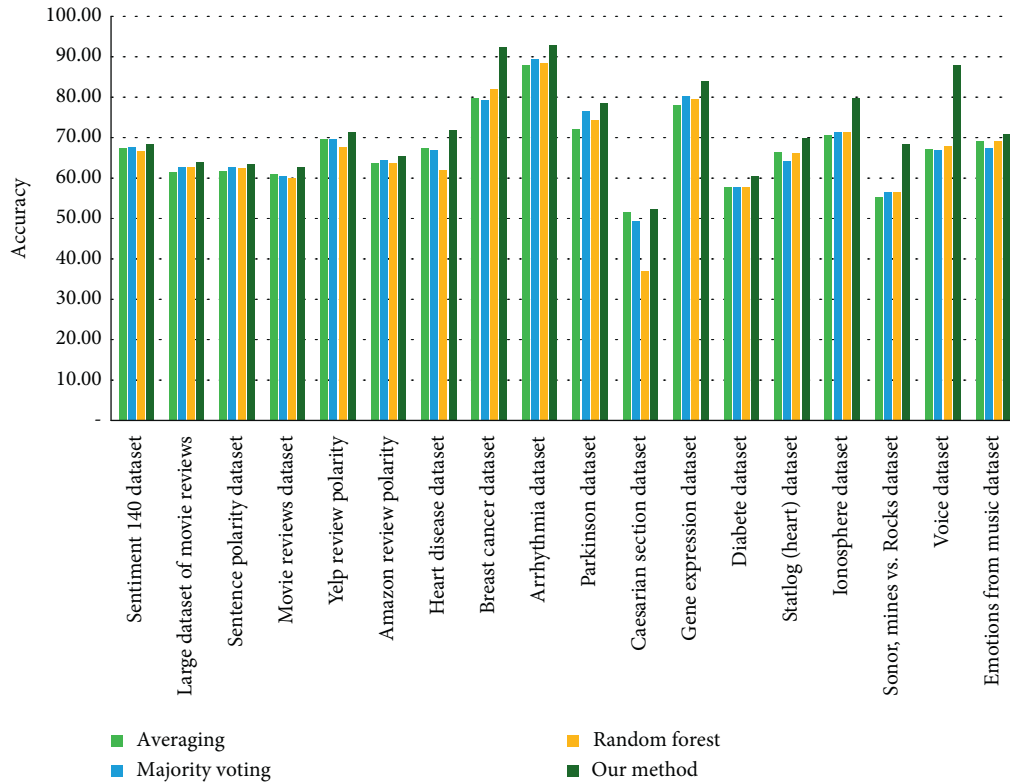


FIGURE 8: The evaluation of the proposed method in the presence of noise.

that the proposed method, due to the use of learning automata, has high adaptability to the problem conditions, and in the presence of noise, contrary to conventional methods in the literature, the proposed method does not suffer a sharp decline, and in such conditions, it shows high efficiency compared to traditional methods. The evaluation of the proposed method in the presence of noise is shown in Figure 8.

## 7. Conclusion and Future Work

Base learner aggregation in ensemble learning should be done in such a way that the following points are met. First point: selecting a base learner leads to the highest performance achievable in the current situation. Second point: if the situation changes due to the dynamics of the problem, the structure of the ensemble will change in such a way that it has the greatest amount of compatibility with the conditions of the new environment. Therefore, in order to meet the above points and achieve an ensemble that is able to adapt to the dynamic conditions of the problem, in this paper, a new method based on the idea of reinforcement learning is proposed to integrate the base learners in the random forest. In the proposed method, learning automata is used to receive feedback from the environment and perform actions on it. The general procedure is to receive feedback from the environment, where the environment is a set of base learners that we intend to combine to achieve a better performance than individual base learners. Learning automata actions include choosing one of the base learners as the best base

learner. The choice of action is based on receiving feedback from the environment. This causes the dynamic behaviour of data to be covered by using the idea of reinforcement learning. On the other hand, given that at each stage, learning automata strives to achieve the highest amount of achievable rewards, it is guaranteed to find the global optima in the problem space. Adaptability is another advantage of the proposed method compared to similar methods in the subject literature.

Due to the fact that in each step learning automata operates based on environmental conditions and received feedback from the environment, the ability to adapt to the problem is met. The results of the evaluations performed in different data show that the proposed method has the ability to achieve all the desired items mentioned above. Despite the fact that, unlike the random forest mechanism, all features are injected into all base learners in the proposed method, the efficiency of the proposed method in dealing with large-volume data has not decreased, and the results are more favorable than the classical methods. The proposed method is independent of the data type and has the ability to handle any other type of data in any field. In order to substantiate this claim, and in order to evaluate the proposed method, different types of data have been chosen. However, there are no restrictions on the proposed method for dealing with different types of data. In this paper, a new method for aggregating the base learners of the random forest using learning automata is proposed. Determining the optimal value for the parameters of reward and penalty in the form of self-tuning is one of the future works that the authors intend to do.

## Data Availability

The authors declare that all the data are available publicly at the UCI repository.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*, Springer, New York, NY, USA, 2012.
- [2] A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, and F. Martínez-Álvarez, “Multi-step forecasting for big data time series based on ensemble learning,” *Knowledge-Based Systems*, vol. 163, pp. 830–841, 2019.
- [3] M. Savargiv, B. Masoumi, and M. R. Keyvanpour, “A new ensemble learning method based on learning automata,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–16, 2020.
- [4] M. Kang, J. Ahn, and K. Lee, “Opinion mining using ensemble text hidden Markov models for text classification,” *Expert Systems with Applications*, vol. 94, pp. 218–227, 2018.
- [5] Y. Zhang, D. Miao, J. Wang, and Z. Zhang, “A cost-sensitive three-way combination technique for ensemble learning in sentiment classification,” *International Journal of Approximate Reasoning*, vol. 105, pp. 85–97, 2019.
- [6] T.-H. Lee, A. Ullah, and R. Wang, *Bootstrap Aggregating and Random Forest*, pp. 389–429, Springer, New York, NY, USA, 2020.
- [7] J. Markel and A. J. Bayless, “Performance of random forest machine learning algorithms in binary supernovae classification,” 2019, <http://arxiv.org/abs/1907.00088>.
- [8] B. Chen, Z. Li et al., “Forest signal detection for photon counting LiDAR using Random Forest,” *Remote Sensing Letters*, vol. 11, no. 1, pp. 37–46, 2020.
- [9] W. Pang, X. Liu, Z. Wang, Y. Fan, and J. Wang, “Predicting RNA molecular specific hybridization via random forest,” in *Proceedings of the 2019 IEEE 7th International Conference on Bioinformatics and Computational Biology (ICBCB)*, pp. 35–38, Hangzhou, China, 2019.
- [10] M. F. Darmawan, A. F. Zainal Abidin, S. Kasim, T. Sutikno, and R. Budiarto, “Random forest age estimation model based on length of left hand bone for Asian population,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, p. 549, 2020.
- [11] S. Park, J. Moon, S. Jung, S. Rho, S. W. Baik, and E. Hwang, “A two-stage industrial load forecasting scheme for day-ahead combined cooling, heating and power scheduling,” *Energies*, vol. 13, no. 2, p. 443, 2020.
- [12] J. Dai, T. Wang, and S. Wang, “A deep forest method for classifying e-commerce products by using title information,” in *Proceedings of the 2020 International Conference On Computing, Networking And Communications (ICNC)*, pp. 1–5, Big Island, HI, USA, 2020.
- [13] M. Papoušková and P. Hajek, “Modelling loss given default in peer-to-peer lending using random forests,” in *Proceedings of the Intelligent Decision Technologies 2019*, pp. 133–141, Springer, Malta, Europe, 2020.
- [14] D. Borup, B. J. Christensen, N. Mühlbach, and M. S. Nielsen, *The Effects Of Targeting Predictors In A Random Forest Regression Model*, 2020.
- [15] S. Sikdar, V. Kadiyali, and G. Hooker, *Price Dynamics on Amazon Marketplace: A Multivariate Random Forest Variable Selection Approach*, 2019.
- [16] L. Giffon, C. Lamothe, L. Bouscarrat, P. Milanesi, F. Cherfaoui, and S. Koço, *Pruning Random Forest with Orthogonal Matching Trees*, 2020.
- [17] J. L. Speiser, B. J. Wolf, D. Chung, C. J. Karvellas, D. G. Koch, and V. L. Durkalski, “BiMM forest: a random forest method for modeling clustered and longitudinal binary outcomes,” *Chemometrics and Intelligent Laboratory Systems*, vol. 185, pp. 122–134, 2019.
- [18] N. Mohapatra, K. Shreya, and A. Chinmay, “Optimization of the random forest algorithm,” in *Advances In Data Science And Management*, pp. 201–208, Springer, New York, NY, USA, 2020.
- [19] Q. Ji, T. Zhu, and D. Bao, “A hybrid model of convolutional neural networks and deep regression forests for crowd counting,” *Applied Intelligence*, pp. 1–15, 2020.
- [20] B. Santra, A. Paul, and D. P. Mukherjee, “Deterministic dropout for deep neural networks using composite random forest,” *Pattern Recognition Letters*, vol. 131, pp. 205–212, 2020.
- [21] M. A. Ganaie, M. Tanveer, and P. N. Suganthan, “Oblique decision tree ensemble via twin bounded SVM,” *Expert Systems with Applications*, vol. 143, Article ID 113072, 2020.
- [22] R. Katuwal, P. N. Suganthan, and L. Zhang, “Heterogeneous oblique random forest,” *Pattern Recognition*, vol. 99, Article ID 107078, 2020.
- [23] P. Probst, M. N. Wright, and A.-L. Boulesteix, “Hyperparameters and tuning strategies for random forest,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge, vol. 9, no. 3, Article ID e1301*, 2019.
- [24] S. Kim, M. Jeong, and B. C. Ko, “Interpretation and simplification of deep forest,” 2020, <http://arxiv.org/abs/2001.04721>.
- [25] V. Jain, J. Sharma, K. Singhal, and A. Phophalia, “Exponentially weighted random forest,” *Pattern Recognition And Machine Intelligence*, pp. 170–178, Lecture Notes in Computer Science, vol. 11941, Springer, Cham, Switzerland, 2019.
- [26] M. Stafoggia, P. Glantz et al., “A random forest approach to estimate daily particulate matter, nitrogen dioxide, and ozone at fine spatial resolution in Sweden,” *Atmosphere*, vol. 11, no. 3, p. 239, 2020.
- [27] S. Hauglin and P. Montesano, “Modelling above-ground biomass stock over Norway using national forest inventory data with ArcticDEM and Sentinel-2 data,” *Remote Sensing of Environment*, vol. 236, p. 111501, 2020.
- [28] H. R. Breidenbach and M. M. Saravi, “Land-subsidence spatial modeling using the random forest data-mining technique,” *Spatial Modeling In GIS and R for Earth And Environmental Sciences*, pp. 147–159, Elsevier, Amsterdam, Netherlands, 2019.
- [29] J. Cai, K. Xu, Y. Zhu, F. Hu, and L. Li, “Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest,” *Applied Energy*, vol. 262, Article ID 114566, 2020.
- [30] L. Wen and X. Yuan, “Forecasting CO2 emissions in Chinas commercial department, through BP neural network based on random forest and PSO,” *Science of The Total Environment*, vol. 718, Article ID 137194, 2020.
- [31] Y.-S. Li, H. Chi, X.-Y. Shao, M.-L. Qi, and B.-G. Xu, “A novel random forest approach for imbalance problem in crime linkage,” *Knowledge-Based System, Article ID 105738*, 2020.

- [32] S. K. Mohapatra and M. N. Mohanty, "Big data analysis and classification of biomedical signal using random forest algorithm," *New Paradigm In Decision Science And Management*, pp. 217–224, Springer, New York, NY, USA, 2020.
- [33] A. Joshi, T. Choudhury, A. Sai Sabitha, and K. Srujan Raju, "Data mining in healthcare and predicting obesity," in *Proceedings of the Third International Conference on Computational Intelligence and Informatics*, pp. 877–888, Hyderabad, India, 2020.
- [34] S. El-Sappagh, R. Sahal et al., "Alzheimer's disease progression detection model based on an early fusion of cost-effective multimodal data," *Future Generation Computer Systems*, vol. 115, pp. 680–699, 2021.
- [35] Y. Saleh, A. Halidou, and P. T. Kapen, "A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19," *Applied Intelligence*, vol. 50, no. 11, pp. 3913–3925, 2020.
- [36] S. Khedkar, P. Gandhi, G. Shinde, and V. Subramanian, *Deep Learning and Explainable AI in Healthcare Using EHR*, pp. 129–148, Springer, New York, NY, USA, 2020.
- [37] T. Han, N. Stone-Weiss, J. Huang, A. Goel, and A. Kumar, "Machine learning as a tool to design glasses with controlled dissolution for healthcare applications," *Acta Biomaterials*, vol. 107, pp. 286–298, 2020.
- [38] A. Subudhi, M. Dash, and S. Sabut, "Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 277–289, 2020.
- [39] A. Javadi, A. Khamesipour, F. Monajemi, and M. Ghazisaeedi, "Computational modeling and analysis to predict intracellular parasite epitope characteristics using random forest technique," *Journal of Public Health*, vol. 49, no. 1, p. 125, 2020.
- [40] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation," *Biomedical Signal Processing and Control*, vol. 52, pp. 456–462, 2019.
- [41] K. K. Singh, S. Kumar, P. Dixit, and M. K. Bajpai, "Kalman filter based short term prediction model for COVID-19 spread," *Applied Intelligence*, pp. 1–13, 2020.
- [42] S.-J. Na, J.-W. Shin, S.-H. Eom, and E.-H. Lee, "A study on random forest-based estimation model for changing the automatic walking mode of above knee prosthesis," *The Journal of IKEEE*, vol. 24, no. 1, pp. 9–18, 2020.
- [43] M. Alloghani, T. Baker, D. Al-Jumeily, A. Hussain, J. Mustafina, and A. J. Aljaaf, "Prospects of machine and deep learning in analysis of vital signs for the improvement of healthcare services," *Nature-Inspired Computation In Data Mining And Machine Learning*, pp. 113–136, Springer, New York, NY, USA, 2020.
- [44] Y. Zhu, W. Xu, G. Luo, H. Wang, J. Yang, and W. Lu, "Random Forest enhancement using improved Artificial Fish Swarm for the medial knee contact force prediction," *Artificial Intelligence in Medicine*, vol. 103, p. 101811, 2020.
- [45] H. Zhang et al., "Deep multi-model cascade method based on CNN and random forest for pharmaceutical particle detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 7028–7042, 2020.
- [46] H. Lee and E. Jung, "An Analysis of Annual Changes on the Determining Factors for Teacher Attachment with Random Forest," pp. 463–470, Springer, New York, NY, USA, 2020.
- [47] X. Liu, L. Liu et al., "Downscaling of solar-induced chlorophyll fluorescence from canopy level to photosystem level using a random forest model," *Remote Sensing of Environment*, vol. 231, Article ID 110772, 2019.
- [48] S. Guanter and J. Santosh Kumar, "Performance evaluation of random forest with feature selection methods in prediction of diabetes," *International Journal of Electrical and Computer Engineering*, vol. 10, 2020.
- [49] A. Subasi, A. Ahmed, E. Aličković, and A. Rashik Hassan, "Effect of photic stimulation for migraine detection using random forest and discrete wavelet transform," *Biomedical Signal Processing and Control*, vol. 49, pp. 231–239, 2019.
- [50] N. El Haouij, J.-M. Poggi, R. Ghozi, S. Sevestre-Ghalila, and M. Jaidane, "Random forest-based approach for physiological functional variable selection for driver's stress level classification," *Statistical Methods & Applications*, vol. 28, no. 1, pp. 157–185, 2019.
- [51] D. Ayata, Y. Yaslan, and M. E. Kamasak, "Emotion recognition from multimodal physiological signals for emotion aware healthcare systems," *Journal of Medical and Biological Engineering*, vol. 40, pp. 149–157, 2020.
- [52] M. Zeraatpisheh, E. Bakhshandeh, M. Hosseini, and S. M. Alavi, "Assessing the effects of deforestation and intensive agriculture on the soil quality through digital soil mapping," *Geoderma*, vol. 363, Article ID 114139, 2020.
- [53] X. Du, P. Wang, L. Fu, H. Liu, Z. Zhang, and C. Yao, "Determination of chlorpyrifos in pears by Raman spectroscopy with random forest regression analysis," *Analytical Letters*, vol. 53, no. 6, pp. 821–833, 2020.
- [54] J. Wang, R. Zuo, and Y. Xiong, "Mapping mineral prospectivity via semi-supervised random forest," *Natural Resources Research*, vol. 29, no. 1, pp. 189–202, 2020.
- [55] S. Liu, X. Qian, H. Wan, Z. Ye, S. Wu, and X. Ren, "NPC three-level inverter open-circuit fault diagnosis based on adaptive electrical period partition and random forest," *Journal of Sensor and Actuator Networks*, vol. 2020, Article ID 9206579, 18 pages, 2020.
- [56] S. S. Rathore and S. Kumar, "An empirical study of ensemble techniques for software fault prediction," *Applied Intelligence*, pp. 1–30, 2020.
- [57] T. Ahmad and H. Chen, "Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems," *Sustainable Cities and Society*, vol. 45, pp. 460–473, 2019.
- [58] S. Gupta, J. Sarkar, M. Kundu, N. R. Bandyopadhyay, and S. Ganguly, "Automatic recognition of SEM microstructure and phases of steel using LBP and random decision forest operator," *Measurement*, vol. 151, Article ID 107224, 2020.
- [59] L. T. T. Ho, L. Dubus, M. De Felice, and A. Troccoli, "Reconstruction of multidecadal country-aggregated hydro power generation in Europe based on a random forest model," *Energies*, vol. 13, no. 7, p. 1786, 2020.
- [60] Y. Zhou, S. Li, C. Zhou, and H. Luo, "Intelligent approach based on random forest for safety risk prediction of deep foundation pit in subway stations," *Journal of Computing in Civil Engineering*, vol. 33, no. 1, Article ID 05018004, 2019.
- [61] X. Deng, Y. Zhan et al., "Predictive geographical authentication of green tea with protected designation of origin using a random forest model," *Food Control*, vol. 107, Article ID 106807, 2020.
- [62] S. A. Liu, P. Ngare, and D. Ikpe, "Probabilistic forecasting of crop yields via quantile random forest and Epanechnikov

- Kernel function,” *Agricultural and Forest Meteorology*, vol. 280, Article ID 107808, 2020.
- [63] H. J. Jeong and M. H. Kim, “Utilizing adjacency of colleagues and type correlations for enhanced link prediction,” *Data & Knowledge Engineering*, vol. 125, Article ID 101785, 2020.
- [64] Z. Khorshidpour, S. Hashemi, and A. Hamzeh, “Evaluation of random forest classifier in security domain,” *Applied Intelligence*, vol. 47, no. 2, pp. 558–569, 2017.
- [65] J. Tian, L. Liu, F. Zhang, Y. Ai, R. Wang, and C. Fei, “Multi-domain entropy-random forest method for the fusion diagnosis of inter-shaft bearing faults with acoustic emission signals,” *Entropy*, vol. 22, no. 1, p. 57, 2020.
- [66] B. Shaw, A. K. Suman, and B. Chakraborty, *Wine Quality Analysis Using Machine Learning*, pp. 239–247, Springer, New York, NY, USA, 2020.
- [67] K. Madhumathi and T. Suresh, *Node Localization in Wireless Sensor Networks Using Multi-Output Random Forest Regression*, pp. 177–186, Springer, New York, NY, USA, 2020.
- [68] Y. Fang, Y. Xu, C. Huang, L. Liu, and L. Zhang, “Against malicious SSL/TLS encryption: identify malicious traffic based on random forest,” in *Proceedings of the Fourth International Congress on Information And Communication Technology*, pp. 99–115, London, UK, 2020.
- [69] T. T. Bhavani, M. K. Rao, and A. M. Reddy, “Network intrusion detection system using random forest and decision tree machine learning techniques,” in *Proceedings of the First International Conference On Sustainable Technologies For Computational Intelligence*, pp. 637–643, London, UK, 2020.
- [70] P. S. Chaithanya, M. R. G. Raman, S. Nivethitha, K. S. Seshan, and V. S. Sriram, “An efficient intrusion detection approach using enhanced random forest and moth-flame optimization technique,” *Computational Intelligence In Pattern Recognition*, pp. 877–884, Springer, New York, NY, USA, 2020.
- [71] Z. Mingjing, “A novel strategy for quantitative analysis of soil pH via laser-induced breakdown spectroscopy coupled with random forest,” *Plasma Science Technology*, vol. 22, no. 7, p. 74003, 2020.
- [72] M.-H. Lee, “Robust random forest based non-fullerene organic solar cells efficiency prediction,” *Organic Electronics*, vol. 76, Article ID 105465, 2020.
- [73] J. Zhang, G. Ma, Y. Huang, J. sun, and F. Aslani, “Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression,” *Construction and Building Materials*, vol. 210, pp. 713–719, 2019.
- [74] W. Nener, C. Wu, H. Zhong, Y. Li, and L. Wang, “Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization,” *Geoscience Frontiers*, vol. 12, no. 1, pp. 469–477, 2020.
- [75] P. Zhang, Z.-Y. Yin, Y.-F. Jin, and T. H. T. Chan, “A novel hybrid surrogate intelligent model for creep index prediction based on particle swarm optimization and random forest,” *Engineering Geology*, vol. 265, p. 105328, 2020.
- [76] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A comparative analysis of logistic regression, random Forest and KNN models for the text classification,” *Augmented Human Research*, vol. 5, no. 1, pp. 1–16, 2020.
- [77] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, “An ensemble machine learning approach through effective feature extraction to classify fake news,” *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021.
- [78] S. N. Singh and T. Sarraf, “Sentiment analysis of a product based on user reviews using random forests algorithm,” *Data Science & Engineering*, vol. 32, pp. 112–116, 2020.
- [79] H. Zhang, N. Cheng, Y. Zhang, and Z. Li, “Label flipping attacks against Naive Bayes on spam filtering systems,” *Applied Intelligence*, 2021.
- [80] R. P. Kaur, M. Kumar, and M. K. Jindal, “Newspaper text recognition of Gurumukhi script using random forest classifier,” *Multimedia Tools and Applications Journal*, pp. 1–14, 2019.
- [81] S. Madichetty and M. Sridevi, “A novel method for identifying the damage assessment tweets during disaster,” *Futur. Gener. Comput. Syst.* vol. 116, pp. 440–454, 2020.
- [82] A. Madasu and S. Elango, “Efficient feature selection techniques for sentiment analysis,” *Multimedia Tools and Applications*, vol. 79, no. 9–10, pp. 6313–6335, 2020.
- [83] A.-C. Chang, C. V. Trappey, A. J. C. Trappey, and L. W. L. Chen, “Web mining customer perceptions to define product positions and design preferences,” *International Journal on Semantic Web and Information Systems*, vol. 16, no. 2, pp. 42–58, 2020.
- [84] R. Kumar and J. Kaur, “Random forest-based sarcastic tweet classification using multiple feature collection,” in *Multimedia Big Data Computing For IoT Applications*, pp. 131–160, Springer, New York, NY, USA, 2020.
- [85] A. Onan and M. A. Toçouglu, “Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts,” *Computer Applications in Engineering Education*, 2020.
- [86] O. M. Baez-Villanueva and M. Zambrano, “RF-MEP: a novel Random Forest method for merging gridded precipitation products and ground-based measurements,” *Remote Sensing of Environment*, vol. 239, Article ID 111606, 2020.
- [87] A. Beck, “Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach,” *Computer Applications in Engineering Education*, 2020.
- [88] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” *Computer Applications in Engineering Education*, Article ID e5909, 2020.
- [89] A. Onan and M. A. Tocoglu, “A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification,” *IEEE Access*, vol. 9, pp. 7701–7722, 2021.
- [90] C. Rodriguez-Pardo, M. A. Patricio, A. Berlanga, and J. M. Molina, “Machine Learning for Smart Tourism and Retail,” pp. 311–333, IGI Global, 2020.
- [91] W. Song and Y. Zhou, “Road travel time prediction method based on random forest model,” in *Smart Trends In Computing And Communications*, pp. 155–163, Springer, New York, NY, USA, 2020.
- [92] A. Jamatia, U. Baidya, S. Paul, S. DebBarma, and S. Dey, “Rating prediction of tourist destinations based on supervised machine learning algorithms,” *Computational Intelligence In Data Mining*, pp. 115–125, Springer, New York, NY, USA, 2020.
- [93] C. F. Baumeister, T. Gerstenberg, T. Plieninger, and U. Schraml, “Exploring cultural ecosystem service hotspots: linking multiple urban forest features with public participation mapping data,” *Urban Forestry & Urban Greening*, vol. 48, Article ID 126561, 2020.
- [94] J. Evans, B. Waterson, and A. Hamilton, “Forecasting road traffic conditions using a context-based random forest

- algorithm,” *Transportation Planning and Technology*, vol. 42, no. 6, pp. 554–572, 2019.
- [95] L. Zhou, X. Dang, Q. Sun, and S. Wang, “Multi-scenario simulation of urban land change in Shanghai by random forest and CA-Markov model,” *Sustainable Cities and Society*, vol. 55, Article ID 102045, 2020.
- [96] H. Liang, Z. Guo, J. Wu, and Z. Chen, “GDP spatialization in Ningbo City based on NPP/VIIRS night-time light and auxiliary data using random forest regression,” *Advances in Space Research*, vol. 65, no. 1, pp. 481–493, 2020.
- [97] Z. Mei, W. Ding, C. Feng, and L. Shen, “Identifying commuters based on random forest of smartcard data,” *IET Intelligent Transport Systems*, vol. 14, no. 4, pp. 207–212, 2020.
- [98] Q. Li, L. Chen, X. Li et al., “A progressive random forest-based random walk approach for interactive semi-automated pulmonary lobes segmentation,” *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 2221–2235, 2020.
- [99] S. L. S. Darshan and C. D. Jaidhar, “An empirical study to estimate the stability of random forest classifier on the hybrid features recommended by filter based feature selection technique,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 2, pp. 339–358, 2020.
- [100] S. K. Lakshmanaprabu, K. Shankar, M. Ilayaraja, A. W. Nasir, V. Vijayakumar, and N. Chilamkurti, “Random forest for big data classification in the internet of things using optimal features,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 10, pp. 2609–2618, 2019.
- [101] P. Liu, X. Wang, L. Yin, and B. Liu, “Flat random forest: a new ensemble learning method towards better training efficiency and adaptive model size to deep forest,” *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 2501–2513, 2020.
- [102] M. Goodwin and A. Yazidi, “Distributed learning automata-based scheme for classification using novel pursuit scheme,” *Applied Intelligence*, vol. 50, no. 7, pp. 2222–2238, 2020.
- [103] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*, Courier Corporation, Chelmsford, CA, USA, 2012.
- [104] A. Rezvani, A. M. Saghiri, S. M. Vahidipour, M. Esnaashari, and M. R. Meybodi, *Recent Advances In Learning Automata*, Vol. vol. 754, Springer, New York, NY, USA, 2018.
- [105] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2017.
- [106] B. Pang and L. Lee, “Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales,” 2005, <http://arxiv.org/abs/0506075>.
- [107] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings Of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Buffalo, NY, USA, 2011.
- [108] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques,” 2002, <http://arxiv.org/abs/0205070>.
- [109] D. Dua and C. Graff, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, 2019.
- [110] C. López-Vázquez and E. Hochsztain, “Extended and updated tables for the Friedman rank test,” *Communications in Statistics - Theory and Methods*, vol. 48, no. 2, pp. 268–281, 2019.

## Research Article

# Automatic Detection of Obstructive Sleep Apnea Events Using a Deep CNN-LSTM Model

Junming Zhang,<sup>1,2,3,4,5</sup> Zhen Tang,<sup>1</sup> Jinfeng Gao ,<sup>1,2</sup> Li Lin,<sup>1</sup> Zhiliang Liu,<sup>1</sup> Haitao Wu,<sup>1,2</sup> Fang Liu,<sup>1,3</sup> and Ruxian Yao <sup>1,2</sup>

<sup>1</sup>College of Information Engineering, Huanghuai University, Zhumadian, Henan 463000, China

<sup>2</sup>Henan Key Laboratory of Smart Lighting, Zhumadian, Henan 463000, China

<sup>3</sup>Henan Joint International Research Laboratory of Behavior Optimization Control for Smart Robots, Zhumadian, Henan 463000, China

<sup>4</sup>Zhumadian Artificial Intelligence & Medical Engineering Technical Research Centre, Zhumadian, Henan 463000, China

<sup>5</sup>Academy of Industry Innovation and Development, Huanghuai University, Zhumadian, Henan 463000, China

Correspondence should be addressed to Ruxian Yao; yaostudy@163.com

Received 9 February 2021; Revised 5 March 2021; Accepted 13 March 2021; Published 23 March 2021

Academic Editor: Nian Zhang

Copyright © 2021 Junming Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Obstructive sleep apnea (OSA) is a common sleep-related respiratory disorder. Around the world, more and more people are suffering from OSA. Because of the limitation of monitor equipment, many people with OSA remain undetected. Therefore, we propose a sleep-monitoring model based on single-channel electrocardiogram using a convolutional neural network (CNN), which can be used in portable OSA monitor devices. To learn different scale features, the first convolution layer comprises three types of filters. The long short-term memory (LSTM) is used to learn the long-term dependencies such as the OSA transition rules. The softmax function is connected to the final fully connected layer to obtain the final decision. To detect a complete OSA event, the raw ECG signals are segmented by a 10 s overlapping sliding window. The proposed model is trained with the segmented raw signals and is subsequently tested to evaluate its event detection performance. According to experiment analysis, the proposed model exhibits Cohen's kappa coefficient of 0.92, a sensitivity of 96.1%, a specificity of 96.2%, and an accuracy of 96.1% with respect to the Apnea-ECG dataset. The proposed model is significantly higher than the results from the baseline method. The results prove that our approach could be a useful tool for detecting OSA on the basis of a single-lead ECG.

## 1. Introduction

Obstructive sleep apnea (OSA) is a major sleep-disordered breathing (SDB) syndrome that is an independent risk factor of coronary heart disease, hypertension, and arrhythmia [1]. According to the manual of the American Academy of Sleep Medicine (AASM) [2], OSA in adults is scored when there is a 90% or more reduction in the baseline of the oral and nasal respiration amplitude for 10 s or more, occurring during sleep. This condition is associated with repetitive airflow limitation and sleep fragmentation, decreasing the sleep time and degrading the sleep quality of the OSA patients [3]. OSA not only

causes excessive daytime neurocognitive deficits, drowsiness, depression, fatigue, and heart stroke [4–6] but can also cause a brain stroke, high blood pressure, arrhythmias, myocardial infarction, and ischemia [7–9]. According to the AASM [2], polysomnography (PSG) is considered to be the gold standard for OSA detection, which is based on a comprehensive evaluation of the sleep signals [10]. PSG involves overnight recording of the patient and the measurement of many signals using the sensors attached to the body, e.g., an electroencephalogram (EEG), electromyogram (EMG), electrocardiogram (ECG), and electrooculogram (EOG), to monitor the respiratory effort and other biophysiological signals [1].



After collecting the PSG data, physicians inspect them using statistical tools to score the OSA events.

However, PSG has several disadvantages. First, patients need to sleep in the hospital for at least one night, which consumes a considerable amount of time and is expensive. Furthermore, many patients cannot sleep well in hospitals. Second, many electrodes have to be connected to the body of a patient. These electrodes will interrupt their sleep, which will result in the deviation of the measurement results. Therefore, it is important to develop methods that can reliably diagnose OSA with a few signals and that can be used at home. According to Mietus and Peng [11], the heart beat interval of patients fluctuates periodically during the occurrence and recovery of OSA. Zarei and Asl [12] indicated that significant changes in heart rate or abnormal activities of the heart may indicate OSA. Additionally, according to our clinical research, patients' compliance is very low when they wear the pressure transducer sensor to obtain the oral and nasal respiration. Patients often pull out the nasal cannula. Therefore, when compared with the ECG signal, nasal airflow data can be unstable due to lead falling off. Hence, in this study, we use ECG signals to detect OSA events.

Traditional visual OSA scoring is a very tedious and time-consuming process for a physician to conduct. Therefore, many alternative OSA detection methods have been developed [13]. These methods were based on bio-signals such as the respiratory [14], snoring [15–17], SpO2 [8, 9, 18], and ECG [12, 19–24] signals, and many authors have obtained a high performance level in terms of OSA detection. However, almost all these methods involved data preprocessing, feature extraction, feature selection, and classification. Although feature extraction is essential to ensure good performance, this process requires considerable domain expertise and is particularly limited to high-dimensional data [25].

Deep learning is an attractive alternative because it can automatically learn and extract features from raw data and can be merged with a classification procedure. In particular, convolutional neural networks (CNNs), which are a popular deep-learning model, have gained considerable success owing to their excellent performance in various domains, including visual imagery [26], speech recognition [27], and text recognition [28]. CNNs have also been applied to biosignal classification problems. For example, in our previous study [29], a CNN can be used to score the sleep stages. Banluesombatkul et al. [30] used metalearning to classify sleep stages. Piriyaajakonkij et al. [31] proposed a SleepPoseNet to recognize sleep postures. An event-related potential encoder network was applied to ERP-related tasks [32]. Wilaiprasitporn et al. [33] used a deep-learning approach to improve the performance of affective EEG-based person identification. Recently, some models based on CNNs have been employed to detect OSA. Urtnasan et al. [25] proposed a method for the automated detection of OSA from a single-lead ECG using a CNN. Ho et al. [10] developed an approach for OSA event detection using a CNN and a single-channel nasal pressure signal. Banluesombatkul et al. [34] used a

CNN to extract ECG signal features and fully connected neural networks for OSA events detection. McCloskey et al. [35] used a CNN and wavelets to analyze the nasal airflow and detect the OSA events. However, most of these methods score OSA events by minute-by-minute analysis. According to the AASM ruler [2], OSA events occur in 10 s or more. Therefore, minute-by-minute analysis will lose some OSA events. At the same time, the duration of each OSA event is different. Multiple OSA events can occur as briefly within only a single minute (i.e., one epoch); at times, one OSA event can be prolonged over multiple epochs. Therefore, it is difficult to detect complete OSA events for these methods.

According to Guilleminault et al. [36], there is a relation between the OSA events and heart rate variability. They indicated that the heart rate decelerates at the beginning of an OSA event and that it suddenly increases when normal breathing is resumed [36]. Because long short-term memory (LSTM) maintains internal memory and utilizes feedback connections to learn temporal information from sequences of inputs, in this study, we propose a new method for OSA detection using the CNN and LSTM. The LSTM [37] is used to learn these dependencies, such as the transition rules employed by physicians, to identify future OSA events from previous ECG epochs. To detect complete OSA events, a window overlapping method is required to accurately detect the OSA events, which can identify the start and end positions of the event. Therefore, the proposed method can alert for OSA events of long duration, which will reduce the rate of sudden death caused by OSA events [38].

This study is organized as follows: the datasets are presented in Section 2, and the methods are presented in Section 3. The experimental results and discussion are presented in Section 4, and Section 5 concludes this study.

## 2. Dataset and Preprocessing

The Apnea-ECG dataset [39], downloaded from <https://www.physionet.org/content/apnea-ecg/1.0.0>, was used to evaluate the proposed approach. The dataset comprises 70 PSG recordings, among which 35 are used in the training set and 35 are used in the test set. The training set was used to update the parameters of the proposed model, and the test set was used to perform independent performance assessments. Each recording contains a continuous digitized ECG signal, a set of apnea annotations (derived by human experts on the basis of the simultaneously recorded respiration and related signals), and a set of machine-generated QRS annotations. The sampling rate for the ECG was 100 Hz with a 12 bit resolution. The records contain variable lengths from 7 to 10 hours. The age of the subjects is between 27 and 63 years, and their weights are 35–135 kg.

First, according to Urtnasan et al. [25], a Chebyshev type-II band-pass filter (5–11 Hz) was used to remove undesirable noise from the single-lead ECG data. Second, the data were segmented into epochs (10 s long) to train the proposed model. Table 1 presents the distribution of all the epochs in the training and test sets. Abnormal epoch means an OSA event.

TABLE 1: The number of normal epochs (NE) and abnormal epochs (AE).

Training set		Test set	
NE	AE	NE	AE
210680	130050	213830	13102

### 3. Methods

**3.1. Convolutional Neural Network.** In this study, we used a one-dimensional (1D) CNN to classify the ECG signals. The CNN comprised convolutional, pooling, and fully connected layers. The net input of neuron  $j$  in layer  $l$  is defined as follows:

$$Z_j^l = \sum_{i \in M_j} w_{j,i}^l * x_i^{l-1} + b_j^l, \quad (1)$$

where  $M_j$  represents the selection of input maps,  $w_{j,i}$  denotes the weight or the filter associated with the connection between neurons  $j$  and  $i$ ,  $x_i^{l-1}$  is the output signal from neuron  $i$  in layer  $l-1$ ,  $b_j^l$  is the bias associated with neuron  $j$  in layer  $l$ , and  $*$  denotes vector convolution. To acquire an output map, an activation function is required as follows:

$$x_j^l = f(z_j^l). \quad (2)$$

When compared with other activation functions, a rectified linear unit (ReLU) exhibits robust training performance. Hence, in this study, we used ReLU as the activation function for the output maps, which can be expressed as follows:

$$f(z_j^l) = \max(0, z_j^l). \quad (3)$$

After the convolutional layer, a pooling layer was placed, which was used to reduce the dimensions of the feature maps, network parameters, and the computational cost associated with successive layers using specific functions to summarize the subregions, such as by considering the average value or the maximum value. Additionally, the pooling layer allowed the CNN to learn features that were scale invariant or can be attributed to the orientation changes [40]. The pooling operation consisted of sliding a window across the previous feature map. Herein, max pooling was used after the convolutional layer was activated. Finally, a dense layer, which was generally used in the final stages of the CNN, was fully connected to the outputs of the previous layers.

**3.2. Batch Normalization.** During the training of a CNN, a change in the distribution of the inputs of each layer will affect the outputs of all the succeeding layers. This can result in difficulty when attempting to train models with saturated nonlinearities [41]. Therefore, batch normalization (BN) was used to solve this problem.

Suppose  $X = \{x_1, x_2, \dots, x_d\}$  is the input to a layer with dimension  $d$ . The corresponding minibatch is  $mb$ . The mean

of all the inputs in the same minibatch can be expressed as follows:

$$\mu = \frac{1}{mb} \sum_{i=1}^{mb} x_i. \quad (4)$$

The variance of the input in a minibatch can be expressed as follows:

$$\sigma^2 = \frac{1}{mb} \sum_{i=1}^{mb} (x_i - \mu)^2. \quad (5)$$

Therefore, BN can be expressed as follows:

$$y_i = \gamma x_i^{\sim} + \beta, \quad (6)$$

where  $x_i^{\sim} = x_i - \mu / \sqrt{\epsilon + \sigma^2}$ ,  $\gamma$ , and  $\beta$  are learnable parameters. The training efficiency of a CNN can be improved using BN. At the same time, BN helps the CNN to train faster and provides high accuracy [41].

**3.3. Long Short-Term Memory.** LSTM controls the cell state via three gates, i.e., a forgetting gate, an input gate, and an output gate. The output features obtained from the previous dense layer of a CNN layer are passed to the gate units. The memory cells constituting the LSTM update their states via the activation of each gate unit controlled to a continuous value between 0 and 1. The hidden state of the LSTM cell  $h_t$  is updated after every  $t$  steps. The input gate, forget gate, and output gate can be written as shown in equations (7)–(9) [37], respectively.

$$i^t = \text{sigmoid}(W_{ni}X^t + W_{hi}X^{t-1} + W_{ci} \circ c^{t-1} + b_i), \quad (7)$$

$$f^t = \text{sigmoid}(W_{nf}X^t + W_{hf}h^{t-1} + W_{cf} \circ c^{t-1} + b_{bf}), \quad (8)$$

$$o^t = \text{sigmoid}(W_{no}X^t + W_{ho}h^{t-1} + W_{co} \circ c^t + b_{bo}), \quad (9)$$

where  $\circ$  represents point-wise multiplication.

The cell states and hidden states can be expressed using equations (10) and (11), respectively.

$$c^t = f^t \circ c^{t-1} + i^t \circ \text{sigmoid}(W_{nc}X^t + W_{hc}h^{t-1} + b_{bc}), \quad (10)$$

$$h^t = o^t \circ \text{sigmoid}(c^t). \quad (11)$$

The CNN and LSTM can be used as backpropagation algorithms to update the parameters of the model during training.

## 4. Experiments

**4.1. Statistical Evaluation Methods.** In this study, we use the kappa coefficient (KP) [42], which is a robust statistical measure of the inter-rater agreement, to evaluate the performance of our method. Additionally, the total accuracy (TAC), sensitivity (SE), specificity (SP), positive predictive value (PPV), and negative predictive value (NPV) were

calculated according to an epoch-by-epoch analysis as follows:

$$TAC = \frac{TP + TN}{TP + FN + FP + TN} \%, \quad (12)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \%, \quad (13)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \%, \quad (14)$$

$$PPV = \frac{TP}{FP + TP} \%, \quad (15)$$

$$NPV = \frac{TN}{TN + FN} \%, \quad (16)$$

where TP, TN, FP, and FN denote the true positive, true negative, false positive, and false negative, respectively. We implement our experiments on a workstation with a GeForce GTX2060 GPU in a Windows environment. The TensorFlow framework is used to train the proposed model.

#### 4.2. The Proposed Deep Model Architecture and Parameters.

To build an optimal OSA detection architecture, we need to understand the characteristics of the input data. The sampling rate of the ECG was 100 Hz, and the 10 s input dimension was 1000. To extract different scale features, we need to set up different size filters. Therefore, experiments are implemented while varying the filters size of the convolution layer to identify the optimal parameters for automated OSA detection. According to existing study [25, 29], we design a network model, which contains a convolution, BN, pooling, dropout, and dense layer, as shown in Figure 1.  $N$  denotes the number of the filters. The parameters and results are shown in Table 2. From Table 2, we can see that model\_2 performs best and model\_1 is the second. However, the parameters of model\_2 are large than those of model\_1. For portable OSA devices or real-time OSA analysis systems, model\_1 is more appropriate. Therefore, model\_1 is used to learning the features representation of ECG. To learn the transition rules of OSA, LSTM is used. The proposed model contains the BN, convolutional, pooling, LSTM, and dense layer, as shown in Figure 2.

The detailed parameters of the proposed model are presented in Table 3. This table shows the number of filters, the size, and stride in each convolution layer, the size and stride of the kernel in each pooling layer, and the output size of each layer, including the LSTM layer. The batch size is 30, the training epoch is 100, and the learning rate is 0.1. Figure 3 shows the learning results in terms of accuracy and loss obtained as the number of epochs is varied. The results show that the accuracy and loss reach stable values after several iterations of learning when applied to the validation dataset. Figure 4 shows the filter morphology and training time with each training epoch. From Figure 4(a), we can see that, after 90 training epochs, the morphology of the filter almost does not change. Figure 4(b) indicates that the speed of model training is fast.

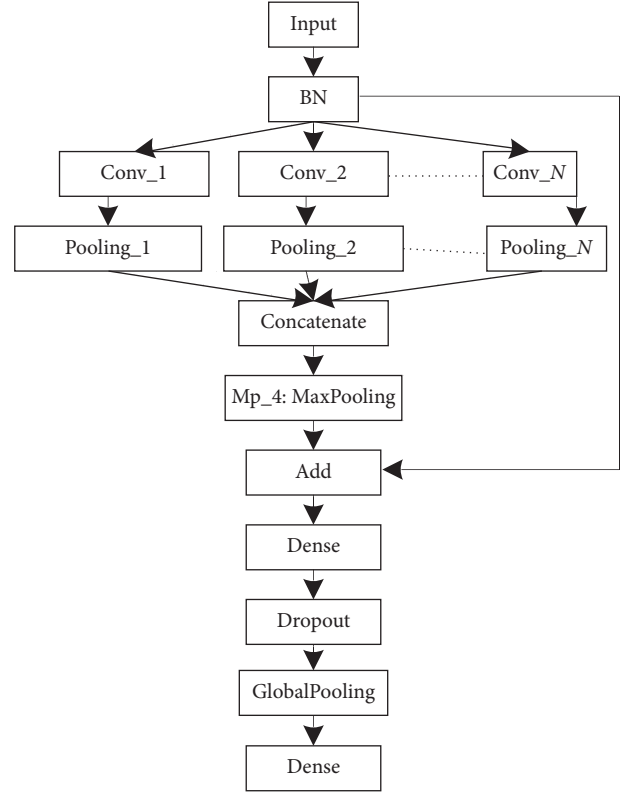


FIGURE 1: Schematic of the proposed CNN model for the automated detection of OSA.

4.3. Performance Results. Table 4 presents the performances of the proposed model for the automated detection of OSA from a single-lead ECG signal. When applied to the test dataset, we obtained a KP of 0.92, an SE of 96.1%, an SP of 96.2%, a TAC of 96.1%, a PPV of 97.6%, and an NPV of 93.8%. As can be seen, the proposed model performed very well for the detection of OSA.

From Table 4, we can observe that 3.9% of the AEs were misclassified as NEs and that 3.8% of the NEs were misclassified as AEs. According to our research, these misclassifications could have been caused by two probable reasons. One reason is that a transition epoch from NE to AE or AE to NE is difficult to classify. For example, Figure 5 shows a transition epoch from NE to AE, whereas Figure 6 shows a transition epoch from AE to NE. A skilled physician would be able to classify these epochs based on the contextual information. However, the proposed model does not use the contextual information to score OSA, making it unable to distinguish the transition epochs. The other reason may be that the proposed model finds it difficult to score the artifact epochs. The ECG signals can be polluted by unwanted noise signals, including body movement. Figure 7 shows a polluted ECG epoch. Because the artifact epochs are few and varied, the proposed model was unable to learn the distributions of all the artifact epochs. Therefore, it is difficult for the proposed model to detect the OSA of artifact epochs. In this case, the usage of handcrafted features seems to be considerably robust.

TABLE 2: The parameters and TACs of the different models.

Name	$N$	Layer	Units	Size	Stride	TAC (%)
Model_1	3	Cn_1	24	$125 \times 1$	$1 \times 1$	94.832
		Cn_2	24	$15 \times 1$	$1 \times 1$	
		Cn_3	24	$5 \times 1$	$1 \times 1$	
Model_2	4	Cn_1	24	$125 \times 1$	$1 \times 1$	94.835
		Cn_2	20	$100 \times 1$	$1 \times 1$	
		Cn_3	24	$15 \times 1$	$1 \times 1$	
		Cn_4	24	$5 \times 1$	$1 \times 1$	
Model_3	4	Cn_1	24	$125 \times 1$	$1 \times 1$	93.92
		Cn_2	20	$50 \times 1$	$1 \times 1$	
		Cn_3	20	$15 \times 1$	$1 \times 1$	
		Cn_4	20	$5 \times 1$	$1 \times 1$	
Model_4	3	Cn_1	24	$100 \times 1$	$1 \times 1$	94.78
		Cn_2	24	$15 \times 1$	$1 \times 1$	
		Cn_3	24	$5 \times 1$	$1 \times 1$	
Model_5	2	Cn_1	30	$125 \times 1$	$1 \times 1$	90.4
		Cn_2	30	$15 \times 1$	$1 \times 1$	

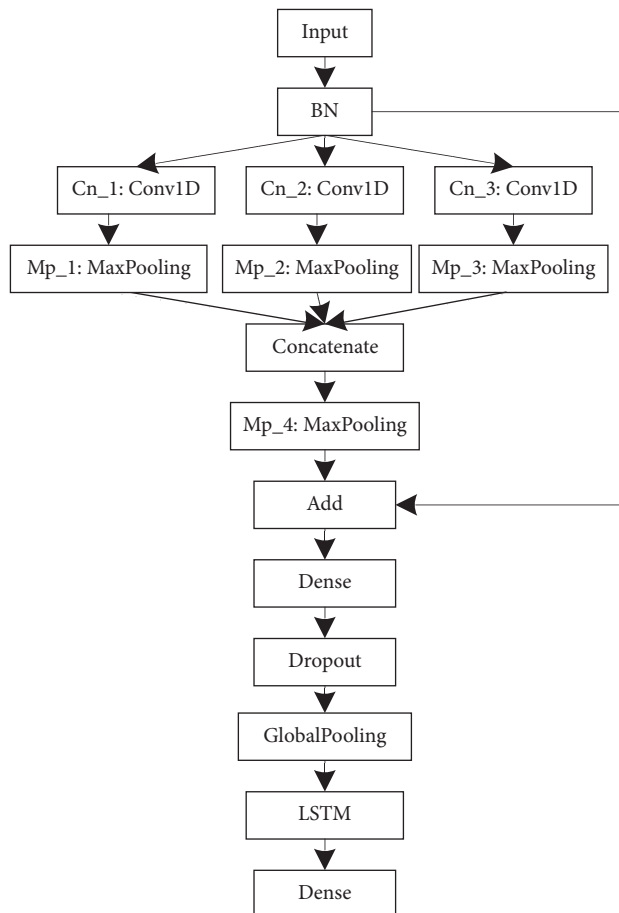


FIGURE 2: Architecture of the proposed model.

**4.4. Benefits of Long Short-Term Memory.** The major advantage associated with the usage of LSTM is that it can be trained to learn long-term dependencies, including the transition rules that are used by the physicians to identify the next possible OSA event(s) from a sequence of ECG epochs. To validate the usefulness of LSTM, we removed the LSTM layer from the model (Figure 2) and then reimplemented the

experiment. This test was named CNN\_1. Table 5 shows the comparison results, where we can see that the proposed model (CNN + LSTM) results in a gain of 1.3% over the TAC of CNN\_1. In addition, KP increased by 0.03 when LSTM was added, verifying our assumption.

Figure 8 shows an example of the NE ECG signal. When the proposed method (CNN + LSTM) is used, the epoch is classified as an NE. However, when CNN\_1 is used, this epoch is scored as an OSA event. The reason for OSA misclassification is that the heart rate is slow at the center of this epoch. According to a previously conducted study [11], the heart rate decelerates when OSA occurs. Therefore, CNN\_1 learned this feature. However, from Figure 8, we can observe that the heart rate changes very little. At the same time, the heart rates of previous epochs are similar to those of this epoch. However, because the LSTM learns long-term dependencies, the CNN + LSTM method accurately detects the epoch, which is the benefit associated with the usage of LSTM.

**4.5. OSA Detection.** As mentioned previously, long OSA is dangerous because it can lead to sudden death. To identify long OSA, the window overlapping method can be used to detect the start and end positions of an OSA event. In this way, long OSA can be detected. Figure 9 shows that the proposed model can detect complete OSA events from the ECG signals. From the nasal airflow signal, we can observe that the OSA events detected by our model have been accurately identified.

**4.6. Comparison of the Proposed Method with Existing Studies.** The comparison of various methods of automatic OSA detection is difficult because different datasets, feature sets, and classifiers are used in different studies. For ensuring a fair comparison with existing studies, Table 6 shows the classification performances of different methods based on single-lead ECG signals. From Table 6, we can observe that the proposed model achieved better performance when compared with those achieved in the previous studies. More

TABLE 3: The parameters of the proposed model.

Layer	Layer type	Units	Unit type	Size	Stride	Output size
Input						$1000 \times 1$
BN						$1000 \times 1$
Cn_1	Convolutional	24	ReLU	$125 \times 1$	$1 \times 1$	$876 \times 24$
Cn_2	Convolutional	24	ReLU	$15 \times 1$	$1 \times 1$	$986 \times 24$
Cn_3	Convolutional	24	ReLU	$5 \times 1$	$1 \times 1$	$996 \times 24$
Mp_1	Max pooling	24		$2 \times 1$	$1 \times 1$	$438 \times 24$
Mp_2	Max pooling	24		$2 \times 1$	$1 \times 1$	$493 \times 24$
Mp_3	Max pooling	24		$2 \times 1$	$1 \times 1$	$498 \times 24$
Concatenate		24				$1429 \times 24$
Mp_4	Max pooling	24		$3 \times 1$	$1 \times 1$	$476 \times 24$
Add	Add	24				$1000 \times 24$
Dense	Fully connected	48	LeakyReLU			$1000 \times 48$
Dropout	Dropout					$1000 \times 48$
Gp	Global pooling					$48 \times 1$
LSTM	LSTM					$64 \times 1$
Dense	Fully connected	2	Softmax			2

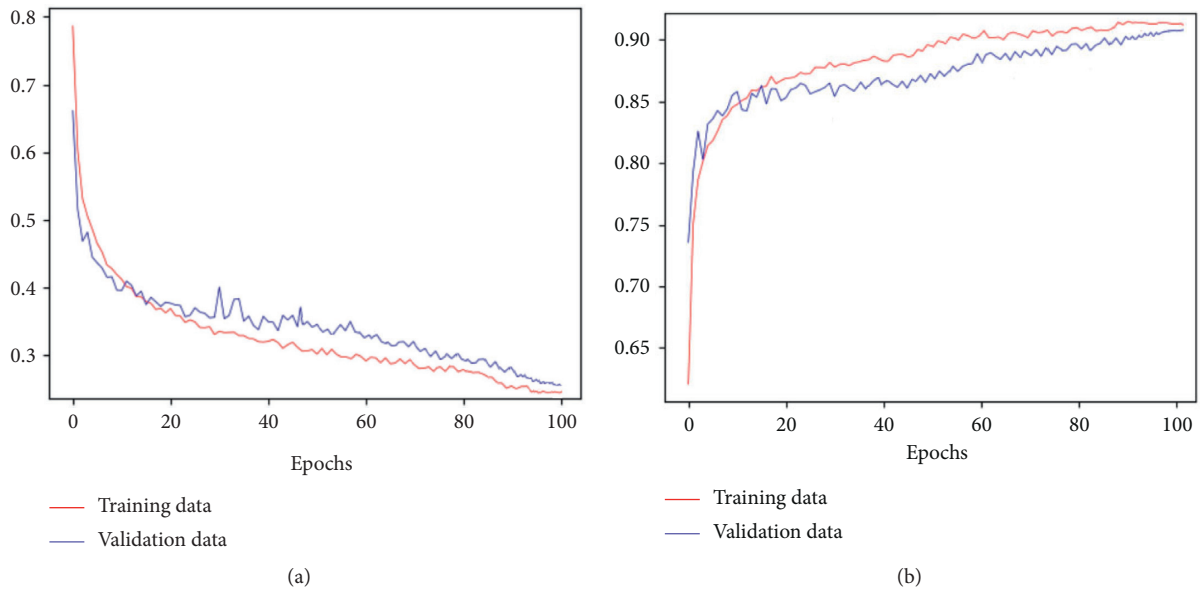


FIGURE 3: Accuracy and loss of the proposed model for automated OSA detection. (a) Loss curve. (b) Accuracy curve.

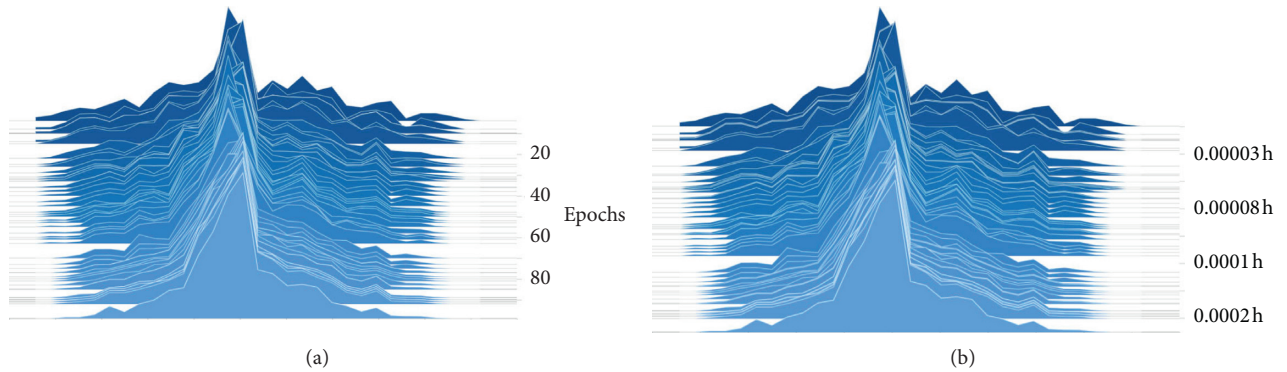


FIGURE 4: Filters morphology and training time with each epoch. (a) Filter morphology. (b) Training time.

TABLE 4: The performances of the proposed model for automated detection of OSA.

	NE	AE	KP	SE (%)	SP (%)	TAC (%)	PPV (%)	NPV (%)
NE	202460	4940	0.92	96.1	96.2	96.1	97.6	93.8
AE	8220	125110						

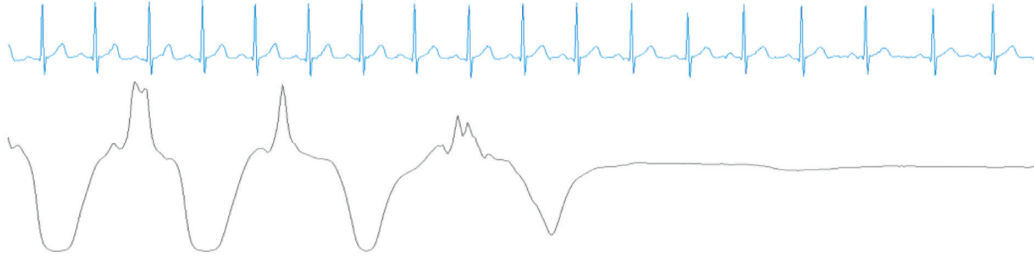


FIGURE 5: A transition epoch from an NE to an AE. Blue denotes the ECG signal, and black denotes the nasal airflow signal.

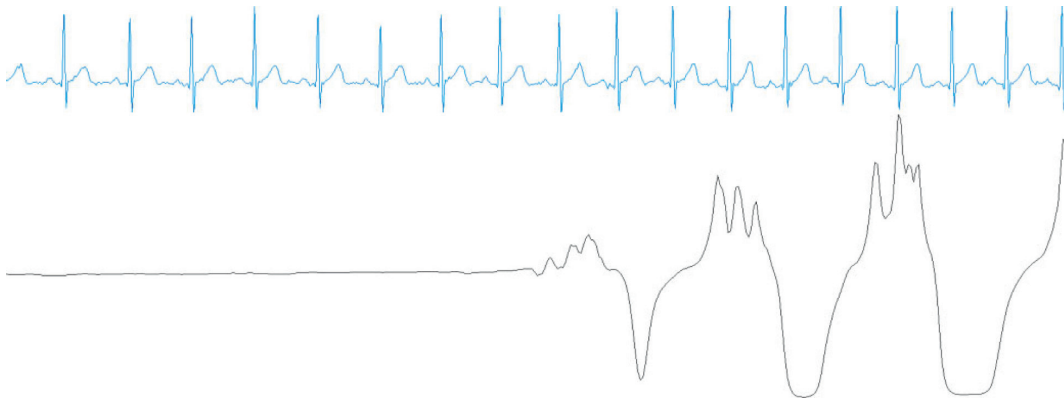


FIGURE 6: A transition epoch from an AE to an NE. Blue denotes the ECG signal, and black denotes the nasal airflow signal.

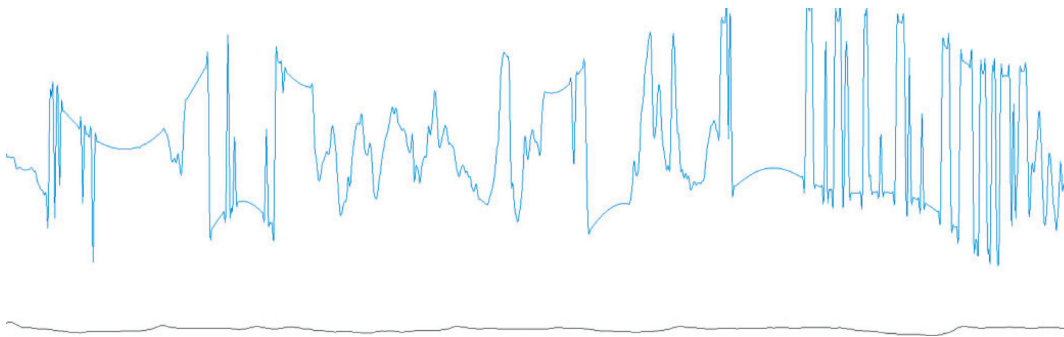


FIGURE 7: An ECG artifact epoch. Blue denotes the ECG signal, and black denotes the nasal airflow signal.

TABLE 5: Comparison of classification performances.

Model	KP	TAC (%)
CNN_1	0.89	94.8
CNN + LSTM	0.92	96.1

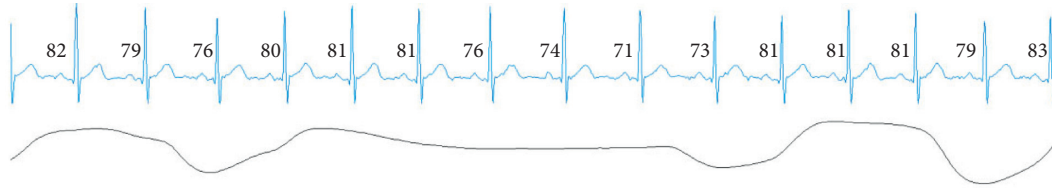


FIGURE 8: A normal ECG epoch.

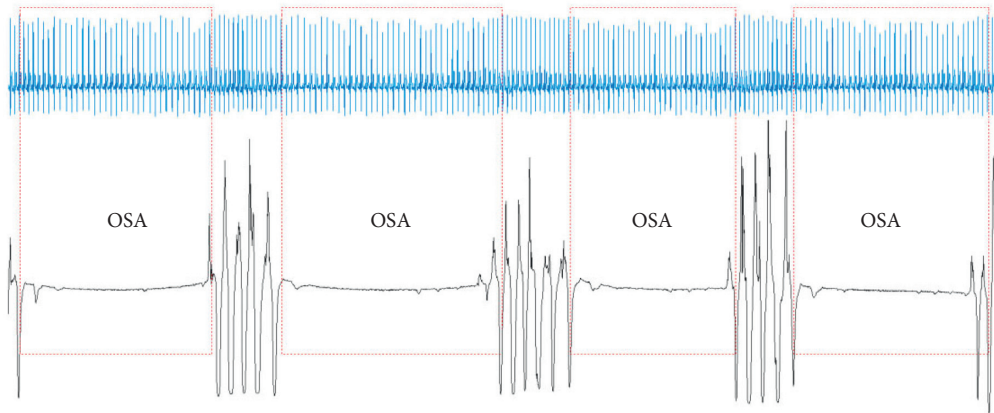


FIGURE 9: The start and end positions of multiple OSA events.

TABLE 6: Comparison of performances of different methods.

Input	Author	Method	TAC (%)	SE (%)	SP (%)
ECG	Jafari [43]	Handcrafted features, SVM	94.8	94.1	95.4
	Chen et al. [44]	Handcrafted features, SVM	82.1	83.2	80.2
	Urtnasan et al. [25]	CNN	96	96	96
	Banluesombatkul et al. [34]	CNN	79.45	77.6	80.1
	Zarei and Asl [12]	Handcrafted features, SVM	94.63	94.43	94.77
	Tripathy [45]	Handcrafted features, kernel extreme learning machine	76.37	78.02	74.64
	Hassan and Haque [46]	Handcrafted features, RUSboot	88.88	87.58	91.49
	Hassan [47]	Handcrafted features, AdaBoost	87.33	81.99	90.72
	Our method	CNN	96.1	96.1	96.2

importantly, our method can be used in conjunction with wearable medical devices, which is very important for home OSA monitoring.

## 5. Conclusions

In this study, we developed an automated OSA event detection method using a CNN, where the feature extraction and selection processes were not required. The proposed method detected the start and end positions of the OSA events based on the overlapping epochs in the ECG signal dataset. Our method automatically extracted the time-invariant features from raw ECG signals without utilizing any handcrafted features. The proposed approach is robust and completely automated, and the method can be easily adapted to other physiological

signal analyses and prediction problems. The TAC and KP of the proposed model applied to the single-channel ECG reached 96.1% and 0.92, respectively. The experimental results showed that the proposed method could accurately score the OSA events and that it achieved comparable performance with other state-of-the-art studies. More importantly, our method can prevent sudden death from OSA, which is important for the patients who are severely affected by OSA.

There are some limitations associated with our CNN method. First, the proposed model can only detect OSA and normal events but not hypopnea events. Although hypopnea is not as serious as OSA, it is still prevalent in sleep-disordered breathing patients. Second, it is difficult to score transition epochs using our method. In the future, we will improve the discrimination ability of our method for

AEs and NEs. In addition, the automated anomaly detection of ECG based on the CNN, which is important to rapidly assess the quality of the ECG data, will be studied.

## Data Availability

The Apnea-ECG dataset, downloaded from <https://www.physionet.org/content/apnea-ecg/1.0.0>, was used to evaluate our proposed approach.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the Henan Provincial Key Science and Technology Research Projects under Grant nos. 202102210127 and 212102210142, in part by the National Science Foundation of China under Grant no. 61973177, in part by the Henan Key Laboratory of Smart Lighting, in part by Henan International Joint Laboratory of Behavior Optimization Control for Smart Robots, in part by the Programme of Henan Innovative Research Team of Cooperative Control in Swarm-based Robotics, in part by the Award Plan for Tianzhong Scholars of Huanghuai University in 2019, in part by Zhumadian Artificial Intelligence & Medical Engineering Technical Research Centre, and in part by Zhumadian Industrial Innovation and Development Research Major Project under Grant no. 2020ZDA06.

## References

- [1] T. Young, L. Evans, L. Finn, and M. Palta, "Estimation of the clinically diagnosed proportion of sleep apnea syndrome in middle-aged men and women," *Sleep*, vol. 20, no. 9, pp. 705–706, 1997.
- [2] C. Iber, S. Ancoli-Israel, A. L. Chesson et al., *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*, pp. 17–48, American Academy of Sleep Medicine, Darien, IL, USA, 2007.
- [3] H. M. Engleman and N. J. Douglas, "Sleep-4: sleepiness, cognitive function, and quality of life in obstructive sleep apnoea/hypopnoea syndrome," *Thorax*, vol. 59, no. 7, pp. 618–622, 2004.
- [4] D. W. Jung, S. H. Hwang, Y. J. Lee, D.-U. Jeong, and K. S. Park, "Apnea-hypopnea index prediction using electrocardiogram acquired during the sleep-onset period," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 2, pp. 295–301, 2017.
- [5] J. B. Dixon, L. M. Schachter, and P. E. O'Brien, "Predicting sleep apnea and excessive day sleepiness in the severely obese," *Chest*, vol. 123, no. 4, pp. 1134–1141, 2003.
- [6] N. M. Ghahjaverestan, S. Masoudi, M. Shamsollahi et al., "Coupled hidden markov model-based method for apnea bradycardia detection," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 527–538, 2016.
- [7] W. W. Flemons, J. E. Remmers, and A. M. Gillis, "Sleep apnea and cardiac arrhythmias: is there a relationship?" *American Review of Respiratory Disease*, vol. 148, no. 3, pp. 618–621, 1993.
- [8] D. Wang, K. K. Wong, L. Rowsell, G. W. Don, B. J. Yee, and R. R. Grunstein, "Predicting response to oxygen therapy in obstructive sleep apnoea patients using a 10-minute daytime test," *European Respiratory Journal*, vol. 51, no. 1, p. 1701587, 2018.
- [9] Y.-Y. Lin, H.-T. Wu, C.-A. Hsu, P.-C. Huang, Y.-H. Huang, and Y.-L. Lo, "Sleep apnea detection based on thoracic and abdominal movement signals of wearable piezoelectric bands," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1533–1545, 2017.
- [10] C. S. Ho, Y. Heenam, K. H. Seok et al., "Real-time apnea-hypopnea event detection during sleep by convolutional neural networks," *Computers in Biology and Medicine*, vol. 100, pp. 123–131, 2018.
- [11] J. E. Mietus and C. K. Peng, "Detection of obstructive sleep apnea from cardiac interbeat interval time series," *Computers in Cardiology*, vol. 27, pp. 753–756, 2000.
- [12] A. Zarei and B. M. Asl, "Automatic detection of obstructive sleep apnea using wavelet transform and entropy-based features from single-lead ECG signal," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 3, pp. 1011–1021, 2019.
- [13] W. M. Faizal, N. N. N. Ghazali, I. A. Badruddin et al., "A review of fluid-structure interaction simulation for patients with sleep related breathing disorders with obstructive sleep," *Computer Methods and Programs in Biomedicine*, vol. 180, Article ID 105036, 2019.
- [14] P. Lakhan, A. Dittapron, N. Banluesombatkul et al., "Deep neural networks with weighted averaged overnight airflow features for sleep apnea-hypopnea severity classification," in *Proceedings of the 2018 IEEE Region 10 Conference*, pp. 0441–0445, Jeju Island, Korea, October 2018.
- [15] K. Rachel and G. Christian, "Obstructive sleep apnea syndrome," *Clinics in Chest Medicine*, vol. 31, no. 2, pp. 187–201, 2017.
- [16] U. Erdenebayar, J.-U. Park, P. Jeong, and K.-J. Lee, "Obstructive sleep apnea screening using a piezo-electric sensor," *Journal of Korean Medical Science*, vol. 32, no. 6, pp. 893–989, 2017.
- [17] J. Hui, L. Liang, and S. Lijuan, "Acoustic analysis of snoring in the diagnosis of obstructive sleep apnea syndrome: a call for more rigorous studies," *Journal of Clinical Sleep Medicine JCSM: Official Publication of the American Academy of Sleep Medicine*, vol. 11, no. 7, pp. 765–771, 2015.
- [18] S. Gutta and Q. Cheng, "Modeling of oxygen saturation and respiration for sleep apnea detection," in *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*, pp. 1636–1640, Pacific Grove, CA, USA, November 2016.
- [19] C. Cheng, C. Kan, and H. Yang, "Heterogeneous recurrence analysis of heartbeat dynamics for the identification of sleep apnea events," *Computers in Biology and Medicine*, vol. 75, pp. 10–18, 2016.
- [20] H. Sharma and K. K. Sharma, "An algorithm for sleep apnea detection from single-lead ECG using Hermite basis functions," *Computers in Biology and Medicine*, vol. 77, pp. 116–124, 2016.
- [21] C. S. S. Viswabargav, R. K. Tripathy, and U. R. Acharya, "Automated detection of sleep apnea using sparse residual entropy features with various dictionaries extracted from heart rate and EDR signals," *Computers in Biology and Medicine*, vol. 108, pp. 20–30, 2019.
- [22] C. Varon, A. Caicedo, D. Testelmans, B. Buyse, and S. Van Huffel, "A novel algorithm for the automatic detection



- of sleep apnea from single-lead ECG,” *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 9, pp. 2269–2278, 2015.
- [23] S. Manish, A. Shreyansh, and A. U. Rajendra, “Application of an optimal class of antisymmetric wavelet filter banks for obstructive sleep apnea diagnosis using ECG signals,” *Computers in Biology and Medicine*, vol. 100, pp. 100–113, 2018.
- [24] W. Lei, L. Youfang, and W. Jing, “A RR interval based automated apnea detection approach using residual network,” *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 93–104, 2019.
- [25] E. Urtnasan, J.-U. Park, E.-Y. Joo, and K.-J. Lee, “Automated detection of obstructive sleep apnea events from a single-lead electrocardiogram using a convolutional neural network,” *Journal of Medical Systems*, vol. 42, no. 6, 2018.
- [26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, pp. 1097–1105, The MIT Press, Cambridge, MA, USA, 2012.
- [27] O. Abdel-Hamid, A. R. Mohamed, H. Jiang et al., “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” in *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4277–4280, Kyoto, Japan, March 2012.
- [28] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in *Proceedings of the 21st International Conference on Pattern Recognition*, pp. 3304–3308, Tsukuba, Japan, November 2012.
- [29] J. Zhang and Y. Wu, “A new method for automatic sleep stage classification,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 5, pp. 1097–1110, 2017.
- [30] N. Banluesombatkul, P. Ouppaphan, P. Leelaarporn et al., “MetaSleepLearner: a pilot study on fast adaptation of bio-signals-based sleep stage classifier to new individual subject using meta-learning,” *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2020.
- [31] M. Piriyajitakonkij, P. Warin, P. Lakhan et al., “SleepPoseNet: multi-view learning for sleep postural transition recognition using UWB,” *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2020.
- [32] A. Ditthaporn, N. Banluesombatkul, S. Ketrat, E. Chuangsuwanich, and T. Wilaiprasitporn, “Universal joint feature extraction for P300 EEG classification using multi-task autoencoder,” *IEEE Access*, vol. 7, pp. 68415–68428, 2019.
- [33] T. Wilaiprasitporn, A. Ditthaporn, K. Matchaparn, T. Tongbuasirilai, N. Banluesombatkul, and E. Chuangsuwanich, “Affective EEG-based person identification using the deep learning approach,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 3, pp. 486–496, 2020.
- [34] N. Banluesombatkul, T. Rakthanmanon, and T. Wilaiprasitporn, “Single channel ECG for obstructive sleep apnea severity detection using a deep learning approach,” in *Proceedings of the 2018 IEEE Region 10 Conference*, pp. 2011–2016, Jeju Island, Korea, October 2018.
- [35] S. McCloskey, R. Haidar, I. Koprinska, and B. Jeffries, “Detecting hypopnea and obstructive apnea events using convolutional neural networks on wavelet spectrograms of nasal airflow,” in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 361–372, Melbourne, Australia, June 2018.
- [36] C. Guilleminault, R. Winkle, S. Connolly, K. Melvin, and A. Tilkian, “Cyclical variation of the heart rate in sleep apnoea syndrome,” *The Lancet*, vol. 323, no. 8369, pp. 126–131, 1984.
- [37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [38] M. Zhang, L. Li, D. Fowler et al., “Causes of sudden death in patients with obstructive sleep apnea,” *Journal of Forensic Sciences*, vol. 58, no. 5, pp. 1171–1174, 2013.
- [39] A. L. Goldberger, L. A. N. Amaral, L. Glass et al., “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2003.
- [40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [41] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.
- [42] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.
- [43] A. Jafari, “Sleep apnoea detection from ECG using features extracted from reconstructed phase space and frequency domain,” *Biomedical Signal Processing and Control*, vol. 8, no. 6, pp. 551–558, 2013.
- [44] L. Chen, X. Zhang, and C. Song, “An automatic screening approach for obstructive sleep apnea diagnosis based on single-lead electrocardiogram,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 106–115, 2015.
- [45] R. K. Tripathy, “Application of intrinsic band function technique for automated detection of sleep apnea using HRV and EDR signals,” *Biocybernetics and Biomedical Engineering/Polish Academy of Sciences. Institute of Biocybernetics and Biomedical Engineering*, vol. 38, pp. 136–144, 2018.
- [46] A. R. Hassan and M. A. Haque, “An expert system for automated identification of obstructive sleep apnea from single-lead ECG using random under sampling boosting,” *Neurocomputing*, vol. 235, pp. 122–130, 2017.
- [47] A. R. Hassan, “Computer-aided obstructive sleep apnea detection using normal inverse Gaussian parameters and adaptive boosting,” *Biomedical Signal Processing and Control*, vol. 29, pp. 22–30, 2016.