Wireless Communications and Mobile Computing

Deep Feature Learning for Big Data

Lead Guest Editor: Xiaojie Wang
Guest Editors: Amr Tolba, Jing Gao, and Zhaolong Ning

Deep Feature Learning for Big Data

Wireless Communications and Mobile Computing

Deep Feature Learning for Big Data

Lead Guest Editor: Xiaojie Wang
Guest Editors: Amr Tolba, Jing Gao, and Zhaolong Ning

Chief Editor

Associate Editors

Academic Editors

Contents

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5562995, 11 pages
https://doi.org/10.1155/2022/5562995

Research Article

Popularity-Guided Cost Optimization for Live Streaming in Mobile Edge Computing

Tao He, Kunxin Zhu, Zhipeng Chen, Ruomei Wang, and Fan Zhou

School of Computer Science and Engineering, Sun Yat-sen University, China

Correspondence should be addressed to Fan Zhou; isszf@mail.sysu.edu.cn

Received 8 January 2021; Revised 20 August 2021; Accepted 5 November 2021; Published 5 January 2022

Academic Editor: Xiaojie Wang

Live streaming service usually delivers the content in mobile edge computing (MEC) to reduce the network latency and save the backhaul capacity. Considering the limited resources, it is necessary that MEC servers collaborate with each other and form an overlay to realize more efficient delivery. The critical challenge is how to optimize the topology among the servers and allocate the link capacity so that the cost will be lower with delay constraints. Previous approaches rarely consider server collaborations for live streaming service, and the scheduling delay is usually ignored in MEC, leading to suboptimal performances. In this paper, we propose a popularity-guided overlay model which takes the scheduling delay into consideration and utilizes MEC collaboration to achieve efficient live streaming service. The links and servers are shared among all channel streams and each stream is pushed from cloud servers to MEC servers via the trees. Considering the optimization problem is NP-hard, we propose an effective optimization framework called cost optimization for live streaming (COLS) to predict the channel popularity by a LSTM model with multiscale input data. Finally, we compute topology graph by greedy scheme and allocate the capacity with convex programming. Experimental results show that the proposed approach achieves higher prediction accuracy, reducing the capacity cost by more than 40% with an acceptable delay compared with state-of-the-art schemes.

1. Introduction

With the rapid popularization of smart devices, the Internet traffic has ushered an explosive growth [1], and almost 82% of all network traffic comes from video traffic [2]. The increasing traffic puts amounts of pressure on the cloud data center, bringing more difficulties for the optimization of the servers [3], especially for some latency-sensitive services, e.g., live streaming. To address this, mobile edge computing (MEC) is brought in as a new technology for live streaming service to reduce the network latency and alleviate the backhaul capacity [4]. Internet service provider (ISP) places nearby MEC servers at the network edge so that users can visit these servers instead of remote cloud servers and get a better experience. Due to limited resources of a single MEC server, multiple servers are usually used to collaborate with each other and form an overlay to deliver the content [5]. The cost of deploying such an overlay mainly comes from the link capacity cost (If an overlay is firstly deployed, it has another cost called *Server Cost* to purchase servers' hardware. Compared with the link capacity cost, *Server Cost* is a one-time deployment cost, so we ignore it in this paper. We think that servers' hardware resources, computing capacities and upload capacities, are enough and will not be an optimization bottleneck). While the link capacity is higher, the source-to-end delay which is defined as the time elapsed from the cloud server to the MEC server is lower but the cost increases rapidly. A critical problem is how to construct the topological graph among these servers and allocate the bandwidth capacity to each link so that the cost will be minimum while the delay is still under a certain bound.

Failed to address the problem above, existing works mainly suffer from following deficiencies.

In MEC environment, most works pay attention to the optimization for static contents [6, 7], such as video on demand (VoD) streaming or image caching, which are delay insensitive. These approaches usually fail to address the rationality of MEC application providers or have different objective functions, leading to improper results in cost optimization for live streaming.

Some works optimize the resource allocation by predicting the popularity of contents [8–10], considering that a few popular videos usually contribute to most of the bandwidth consumption [11]. However, suffering from the similar reason that these models usually focus on the prediction of static contents, it is still hard to meet the real-time requirement of live streaming.

As a considerable impact factor in optimization strategy, server's scheduling delay is usually ignored or not modeled sufficiently in most of existing works [12–14], which renders these models' inaccuracy and inefficiency in reality.

To tackle above challenges, we construct a multisource multichannel overlay model which focuses on the popularity prediction, topology generation, and link bandwidth allocation. We combine the deep neural network and the mathematical model to optimize the overlay deployment cost, i.e., we first predict the popularity of live channel with LSTM model and identify which channel the MEC server should subscribe to. Then, we compute the topology graph and allocate the link capacity by mathematical optimization methods.

In the proposed approach, each channel stream has heterogeneous rate which is constant in transmission, and all packet lengths are equal (the channel rate and the packet length may vary in reality, but they are not variables in our model and do not affect the problem solving. Hence, we simplify them.). Without loss of generality, we suppose that a channel can only originate from one source, and a cloud server could be the source of multiple channels. A subscriber (defined as a MEC server which subscribes to channels) could receive a channel stream from a cloud server, another MEC server subscribing to the same channel or a helper (a helper denotes a MEC server which forwards unsubscribed channel streams.). Therefore, there are some nodes (helpers) that can be included or excluded in a channel transporting path, and all channel trees are combined into a mesh.

In practice, the link cost is usually charged by the maximum rented capacity. And the source-to-end delay actually consists of four type delays, e.g., link propagation delay, server transmission delay, server processing delay, and server queuing delay. More specifically, the link propagation delay is the time that a packet travels over a physical connection, usually reflected by the round-trip time. And the server processing delay can be omitted while the computing resource is sufficient as aforementioned. Then, we combine server transmission delay and server queuing delay into server scheduling delay, defined as the used time that a packet is kept on a server until it is completely transmitted out. In this way, the source-to-end delay consists of link propagation delay and server scheduling delay. As a result, a high scheduling delay will cause network congestion, which should be taken into consideration by all means.

As presented in Figure 1, it depicts an example of proposed overlay model where $S1$, $S2$ are cloud (source) servers and $S3$–$S6$ are MEC servers in different regions. Cost optimization is carried out by a central optimizer which continuously collects network parameters and sends control message flows. The workflow is shown in Figure 2. The optimizer predicts the popularity of live channels and decides the subscribed list of each MEC server. Based on obtained information and the delay requirements, the optimizer computes a topology graph with least deployment cost. Finally, the optimizer informs MEC servers of the optimized information, i.e., the subscribed list, the topology graph, and the link capacity. Once MEC servers receive the control message, they collaborate with each other and form an overlay to deliver live channel A to server $S3$ and $S5$, channel B to $S4$, $S5$, and $S6$. In this overlay, channel stream A is pushed from $S1$ to $S5$ with $S4$ forwarding. $S4$ does not subscribe to channel A but forward its data as a helper. The link between $S4$ and $S5$ transports two channels simultaneously.

In summary, we make the key contributions as follows:

(i) Aiming at live streaming service, we formulate the optimization problem and construct a multisource multichannel overlay model in which the MEC servers collaborate with each other. In addition, the scheduling delay is also taken into consideration to construct an optimized topology graph among MEC servers, achieving lower link capacity with delay constraints

(ii) Instead of using fixed live channel popularity for the optimization, we utilize state-of-the-art LSTM model to learn the features of historical streaming data for adaptive and accurate popularity prediction. Furthermore, we also takes the information of time and weekdays into consideration, employing multiscale input data to make a more accurate and robust prediction

(iii) Cost optimization for live streaming (COLS) is proposed as a complete and efficient cost optimizer framework, which considers the whole systematic flow of the optimization in a logical order, including accurate key parameters prediction, comprehensive overlay model formulation, optimal topology generation, and efficient capacity allocation. Finally, COLS is able to achieve a lower cost in polynomial time while meeting the delay constraint

The remainder of this paper is organized as follows. After reviewing related works in Section 2, we elaborate the popularity prediction of live channel in Section 3. Following the mathematical formulation in Section 4, we compute the topology in Section 5. Illustrative experiment results are presented in Section 6. Finally, we conclude in Section 7.

2. Relate Work

In this section, we review related works in the areas of MEC collaboration, popularity prediction, and scheduling delay model.

MEC collaboration. In MEC environment, many works [6, 7] realize collaboration mechanism among the servers to achieve higher efficiency. For example, the approach proposed in [6] utilizes the collaboration between the MEC servers to cache the static content in spare time, e.g., midnight, which is impractical for live streaming service. Since these proposed optimization methods aim to allocate

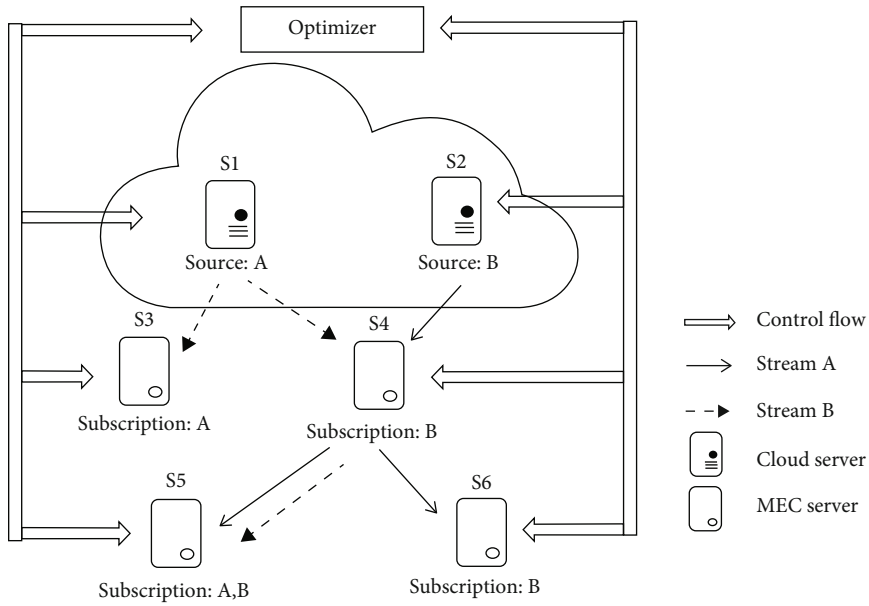


Figure 1: A multisource multichannel overlay model in MEC.

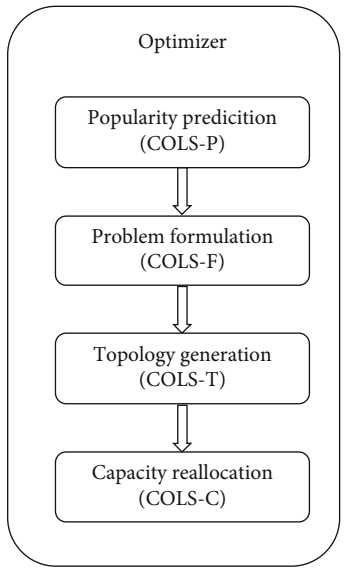


Figure 2: Workflow of optimizer.

the resource efficiently, they usually have different optimization objective in resource utilization and fail to be applied in live streaming services with real-time requirement.

Recently, some specially designed resource optimization methods [15–18] are proposed for live streaming services. For instance, CCAS [15] proposes an auction-based algorithm to optimize the backhaul capacity and the caching space so as to improve the live video quality. Zhang et al. [16] model the computational and wireless spectrum resource in edge-clouds networks. They propose a Markov decision process to decrease the latency of live streaming services. Nevertheless, these approaches usually focus on resource optimization for a single MEC server and rarely consider the collaboration among multiple servers. It potentially results in a low performance such as the network congestion while the user request is higher.

Popularity of video stream. As a key parameter, some works predict the popularity of videos by analyzing the image frame. For example, TLRMVR [8] proposes a novel low-rank multiview embedding learning method to predict the popularity of microvideo. MMVED [9] combines multiple features (image frame, acoustic, and textual info) and considers the randomness for the mapping from data to popularity. Although these approaches are able to achieve efficient prediction, they usually aim at the static complete file and need to parse entire image frame, which is impractical for live streaming. Inspired by success of deep learning techniques, Deepcache [19] predicts the popularity with LSTM Encoder-Decoder to cache contents smartly. And BSPP [10] presents a model for predicting the number of user requests based on Malcov model in MEC and further designs an offloading scheme based on this model. Although effective, these approaches utilize the data in single dimension to predict the popularity, which lack sufficient robustness while facing the random noise and outliers.

On the other hand, some approaches [20, 21] use both the popularity and retention rate of video streams to maximize video bitrate for efficient utilization of bandwidth. Distinguished from these approaches which emphasize bitrate adaptation, this paper focuses on the topology optimization for the cooperation of MEC servers while additionally considering the scheduling delay to achieve lower link capacity with delay constraints. Since our proposed approach and these methods have different focus, respectively, they can be combined to achieve better performances of live streaming services.

Scheduling delay model. For the mathematical model about overlay, most existing works [12–14] rarely formulate the relationship between the link capacity and the server scheduling delay. BSUM [12] considers the scheduling delay

and constructs the topology graph among MEC servers to optimize the resource. But it studies the scheduling delay insufficiently without considering the impact of different link capacities on the scheduling delay, which limits the effect of optimization consequently.

3. Popularity Prediction

In this section, we take advantage of state-of-the-art LSTM model to predict the popularity of live channels, which is the first step called COLS-P of the optimizer as shown in Figure 2.

For time series data, recurrent neural network (RNN) has been widely used to capture the temporal correlations and continuity constraints. As a variant of RNN, long short-term memory (LSTM) model solves the long-term dependence problem of general RNN and enables the network to learn the long-term dependence of time series by selectively memorizing the characteristic information of time series. At each timestep of input time series, LSTM applies the following operations:

$$\begin{aligned} f_t &= \sigma_g\left(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + b_f\right), \\ i_t &= \sigma_g(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + b_i), \\ o_t &= \sigma_g(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + b_o), \\ \tilde{c}_t &= \sigma_c(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + b_c), \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t, \\ \mathbf{h}_t &= o_t \circ \sigma_h(c_t), \\ \mathbf{y}_t &= \sigma_o\left(W_y \mathbf{h}_t + b_y\right), \end{aligned} \tag{1}$$

where the operator $\circ$ denotes the Hadamard product (element-wise product). The subscript t denotes the time step. $\mathbf{x}_t$ represents the input at time t and $\mathbf{h}_t$ represents hidden state. f, i, o represent *forget gate*, input and output *reset gates,* respectively, and c denotes *memory cell state*. W, U, and b are weight matrices and bias which need to be learned during training. And σ indicates the activation function.

The consecutive historical popularity data can be regarded as time series data. Intuitively, we employ state-of-the-art LSTM model to predict the live channel popularity based on historical data. Specifically, Figure 3 shows the popularity of one live channel on one MEC server in 300 consecutive hours and the second red box records (in 100-150 hours period) is further detailed in Figure 4, which presents the popularity changes in single day. We observe from these figures that the time data have an impact on the channel popularity:

(i) *Weekday impact.* The data in two red boxes of Figure 3 correspond to the channel popularity in Sunday and Wednesday, respectively. As presented in Figure 3, it is obvious that the popularity of Sunday is higher than that of Wednesday, so the message of weekday can be a valid supplementary information for accurate prediction

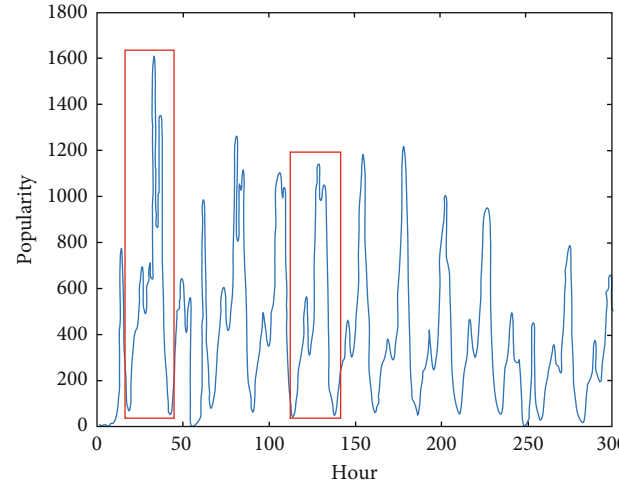


Figure 3: Live channel popularity records.

(ii) *Hour impact.* As shown in Figure 4, we can easily distinguish the popularity changes in the third black box from the first two via the trends. However, the first two is hard to be distinguished from each other just with the trends. Fortunately, as we can observed in Figure 4, the different period of time in the day (different hours) can be an efficient indicator to resolve this confusion

Therefore, different from previous works which only take the historical popularity data for training, we also consider the influence of time data (weekday and hour info impacts) and utilize multiscale time data for training to achieve more accurate prediction.

We train the LSTM network for each channel individually. The raw data consists of consecutive tuples, which contain hourly, weekday, and popularity records (i.e., user requests) per hours. The original data is processed in the form of sliding window, in which these tuples are treated as the input sequence. Then, the popularity in next hour is set as a label. As the popularity prediction in this paper is a regression task, the loss function of the network is set to mean square error (MSE) which is the most common and widely used loss function for regression task.

When the training is completed, for a channel n, we use the historical popularity records of one MEC server to predict its popularity $p^{(n)}$ in the next time period. Define $q^{(n)} = p^{(n)}/\tau^{(n)}$, where $\tau^{(n)}$ is the streaming rate of channel n. Then, we sort the live channel by $q^{(n)}$ and select the first k live channels as the subscribed list $\mathcal{N}_i$ of the MEC server i.

4. Overlay Formulation

After we learned from LSTM network which channels each MEC server should subscribe to (i.e., $\mathcal{N}_i$), we formulate the overlay model including topology model, cost model, delay model, and joint optimization for live streaming.

The major symbols used in this paper are presented in Table 1. We regard the overlay as a directed complete graph $G = (\mathcal{V}, \mathcal{E})$, where V is the set of all servers (cloud servers

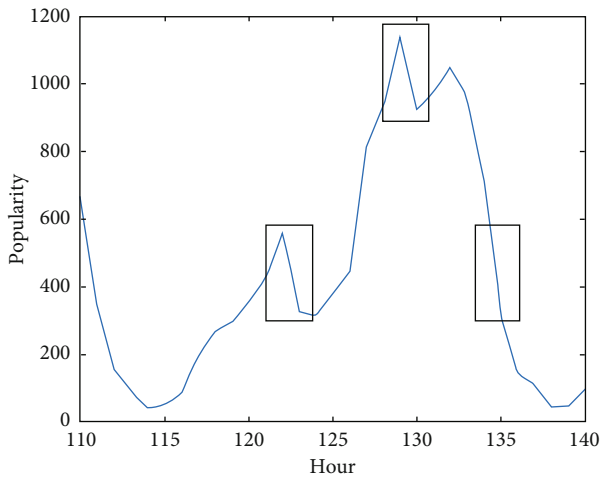


Figure 4: Popularity records in specific time period.

Table 1: Major notations.

Notation	Definition
$\mathcal{S}$	Set of sources (cloud servers)
$\mathcal{M}$	Set of subscribers (MEC servers)
$\mathcal{V}$	Set of servers where $\mathcal{V} = \mathcal{S} \cup \mathcal{M}$
$\mathcal{E}$	Set of links
$\mathcal{N}$	Set of channels
$\mathcal{N}_i$	Set of subscribing channels of server i
$\tau^{(n)}$	Streaming rate of channel n
$s^{(n)}$	Source server of channel n
$p^{(n)}$	Predicted popularity of channel n
$\mathcal{M}^{(n)}$	Set of subscribers of channel n
$T^{(n)}$	Deliver tree of channel n
$\langle i, j \rangle$	Link from server i to server j
b_{ij}	Link capacity of $\langle i, j \rangle$
c_{ij}	Capacity cost of $\langle i, j \rangle$
x_{ij}	Indicator indicating whether $\langle i, j \rangle$ is used
d^p_{ij}	Propagation delay of $\langle i, j \rangle$, $d^p_{ij} = d^p_{ji}$
d^s_i	Worst-case scheduling delay of server i
$D^{(n)}_i$	Source-to-end delay of server i in tree $T^{(n)}$
D	Total delay constraint
C	Total capacity cost

and MEC servers). Let $\mathcal{S}$ be the set of sources (cloud servers) and M be the set of subscribers (MEC servers). So, $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is the set of possible overlay connections and $\mathcal{V} = \mathcal{S} \cup \mathcal{M}$. $\mathcal{N}$ is the set of channels. Denote the rate of channel n as $\tau^{(n)}$. The streaming is delivered to subscribers in $\mathcal{M}^{(n)}$ via a tree. A subscriber receives a stream either from a cloud server, a MEC server which subscribes to the same channel or a helper. There are totally $|\mathcal{N}|$ trees, and we denote the tree of channel n as $T^{(n)}$.

4.1. Topology Model. Equation (2) shows a variable $x^{(n)}_{ij}$ which indicates whether link $\langle i, j \rangle$ is used in tree $T^{(n)}$. All $x^{(n)}_{ij}$ are combined into a vector solution $\vec{x}$.

$$x^{(n)}_{ij} = \begin{cases} 1, \text{if } \langle i, j \rangle \in T^{(n)}, \\ 0, \text{otherwise}, \end{cases} \tag{2}$$

$$\vec{x} = \left\{ x_{ij} \middle| x_{ij} = \max_n x^{(n)}_{ij}, n \in \mathcal{N} \right\}. \tag{3}$$

Equations (4) and (5) guarantee each channel tree is connected, and there is no isolated server. Also, there is no loop in each tree as shown in Equation (6). $|\mathcal{M}^{(n)}|$ is the number of MEC servers which subscribe to channel n. $s^{(n)}$ is the source server of channel n. Y is a subset of servers and $E(Y)$ denotes the set of links connecting servers in Y.

$$\left| \mathcal{M}^{(n)} \right| \le \sum_{\langle i,j \rangle \in \mathcal{E}} x^{(n)}_{ij} \le |\mathcal{M}|, \quad \forall n \in \mathcal{N}, \tag{4}$$

$$\sum_{\langle i,j \rangle \in \mathcal{E}} x^{(n)}_{ij} \ge 1, \quad \forall i \in \mathcal{M} \cup \left\{ s^{(n)} \right\}, \forall j \in \mathcal{M}^{(n)}, \forall n \in \mathcal{N}, \tag{5}$$

$$\sum_{\langle i,j \rangle \in \mathcal{E}(\mathcal{Y})} x^{(n)}_{ij} \le |\mathcal{Y}| - 1, \quad \forall \mathcal{Y} \subseteq \mathcal{M} \cup \left\{ s^{(n)} \right\}, \forall n \in \mathcal{N}. \tag{6}$$

4.2. Cost Model. Denote the capacity of link $\langle i, j \rangle$ as b_{ij}. Like x_{ij}, all b_{ij} are combined into a vector $\sim b = \{b_{12}, b_{13}, \cdots, b_{ij}\}$. The link cost is c_{ij}, which is a linear function of b_{ij}. Total cost C is the sum of all link capacity costs, i.e.,

$$C = \sum_{i \in \mathcal{V}, j \in \mathcal{M}} x_{ij} * c_{ij}(b_{ij}), \qquad c_{ij}(b_{ij}) = k_{ij} * b_{ij}, \tag{7}$$

where k_{ij} is a constant coefficient.

4.3. Delay Model. We employ a sequential scheduling model in which a parent node transmits packets into one link after another sequentially [22]. Denote the worst-case scheduling delay of server i as $d^s{}_i$, which is the maximum amount of time that a packet has to wait until it is transmitted out completely (according to a packet, its queuing delay is the sum of other packets' transmission delay.). To avoid the congestion, $d^s{}_i$ should be smaller than the time interval L/τ between two sequential packets, where L is the packet size [22]. Therefore, we set following congestion constraints:

$$d^s_i = \sum_n \sum_{k \in \Gamma_i} \frac{L \cdot x^{(n)}_{ik}}{b_{ik}}, \tag{8}$$

$$d^s_i \le \frac{L}{\tau^{\max}}, \tau^{\max} = \max_n \tau^{(n)}, \quad n \in \mathcal{N}^{\text{Out}}_i, \tag{9}$$

where Γ_i is the set of children (with repetition) of server i in

all channel trees, N_i^{Out} is the set of channels that server i is streaming out. $\tau^{\max}$ is the maximum streaming rate of these channels (N_i^{Out} may be different from N_i since server i can be the leaf of a tree (it will not stream out such channel) or it acts as a helper to relay an unsubscribed stream.).

We illustrate an example of the scheduling delay in Figure 5. Server i is the parent node while others are son node. b_1 and b_2 are capacities of link $\langle i, j_1\rangle$ and $\langle i, j_2\rangle$, respectively. Channel streams A and B are pushed from server i. Servers j_1 and j_2 subscribe to channel $\{A\}$ and $\{A, B\}$. Hence, child set Γ_i of server i is $\{j_1, j_2, j_2\}$ (j_2 is calculated repeatedly). In a scheduling period, server i transmits one packet into each edge $\langle i, j_1\rangle$, $\langle i, j_2\rangle$, and $\langle i, j_2\rangle$. Therefore, it transmits three packets totally. The worst-case scheduling delay is the maximum amount of time that the third packet has to wait until it is transmitted out completely, i.e., $d_i^s = L/b_1 + L/b_2 + L/b_2$. Denote the rate of channel A and B as $\tau^{(A)}$ and $\tau^{(B)}$, respectively, then we have $d_i^s \le L/\max(\tau^{(A)}, \tau^{(B)})$.

$$D_j^{(n)} = D_i^{(n)} + d_i^s + d_{ij}^p \le D, \quad \forall j \in \mathcal{M}, \forall n \in \mathcal{N}. \tag{10}$$

Equation (10) shows the source-to-end delay constraint. Denote the propagation delay of link $\langle i, j\rangle$ as d_{ij}^p and the source-to-end delay of server j in tree $T^{(n)}$ as $D_j^{(n)}$. $D_j^{(n)}$ is the sum of the source-to-end delay of its parent i in tree $T^{(n)}$, the scheduling delay of i and the propagation delay of link $\langle i, j\rangle$. To ensure quality of service, the source-to-end delay of each node is bounded by a constant value.

4.4. Joint Optimization Model. Combining the above model, we formulate our cost optimization problem as follows:

$$\begin{aligned} &\text{Objective}: \min_{\vec{x}, \vec{b}} (13), \\ &\text{subject to}: (9), (10), (11), (12), (15) \text{and} (16). \end{aligned} \tag{11}$$

Our goal is to find overlay trees among MEC servers for each live channel (optimize $\vec{x}$) and allocate the capacity (optimize $\vec{b}$) to minimize the cost while the delay is still under a boundary. However, it is a mixed integer nonlinear programming which is NP-hard [23]. Besides, from Equations (8) and (10) and Figure 5, we find that the source-to-end delay of server j_1 can be affected by link capacity b_2, even though link $\langle i, j_2\rangle$ does not connect j_1. It means there are correlations among different link capacities. When a tree is constructed completely, the scheduling delay is not determined and is affected by other trees which are constructed later. These factors bring difficulties in solving problems. So, we divide the original problem into two subproblems and solve them sequentially in Section 5.

5. Algorithm Design

To simplify the problem, we divide the original problem into two subproblems: topology generation (COLS-T) and capacity allocation (COLS-C). In COLS-T, we ignore the scheduling delay and congestion constraints to construct an overlay that meets the delay bound. In COLS-C, we reassign the capacity to each link so as to reach a lower cost based on the aforementioned topology.

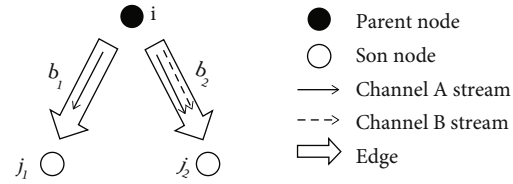


Figure 5: Scheduling delay example.

5.1. Topology Generation (COLS-T). In this section, we ignore the scheduling delay and constraint as present in Equation (9), focusing on the propagation delay to construct the tree. Hence, the problem is transformed into how to find a series of Steiner minimum trees [24] under the hop-constraint. We use a greedy scheme to solve it in polynomial time.

There are totally $|\mathcal{N}|$ channel trees. Each tree is initialized and has only a source cloud server. The initial capacity of a tree is its channel rate (In COLS-C, we will reallocate the capacity). We expand the tree from the source server to subscribing servers by adding a server into a partially constructed tree in each iteration. We define a metric ($\Delta C_{ij}^{(n)}$ or $\Delta C_{ihj}^{(n)}$) called the **M**arginal **U**nit **C**ost (*MUC*) to determine which server is added. A server has two ways to join in the tree, and their *MUC* is given by:

(i) Server j is directly connected via server i and *MUC* is:

$$\Delta C_{ij}^{(n)} = \begin{cases} \dfrac{c_{ij}\left(t_{ij} + \tau^{(n)}\right) - c_{ij}\left(t_{ij}\right)}{\tau^{(n)}}, D_i^{(n)} + d_{ij}^p \le D \\ +\infty, D_i^{(n)} + d_{ij}^p > D \end{cases}. \tag{12}$$

(ii) Server j is connected through a potential helper h via server i. *MUC* is given by:

$$\Delta C_{ihj}^{(n)} = \begin{cases} \Delta C_{ih}^{(n)} + \Delta C_{hj}^{(n)}, D_i^{(n)} + d_{ih}^p + d_{hj}^p \le D \\ +\infty, D_i^{(n)} + d_{ih}^p + d_{hj}^p > D, \end{cases} \tag{13}$$

where t_{ij} is the concurrent throughput of link $\langle i, j\rangle$.

COLS-T is outlined in Algorithm 1. In each iteration, we select link $\langle i, j\rangle$ which incurs the smallest *MUC* and connect the corresponding server j into tree $T^{(n)}$. Then, we update overlay parameters and continue a new iteration until all servers are connected. Finally, we combine all Steiner trees $\{T^{(n)}\}$ into a mesh.

$T^{(n)} \longleftarrow s^{(n)}, J^{(n)} \longleftarrow \mathscr{P}^{(n)}, \mathscr{H}^{(n)} \longleftarrow \mathscr{P} - \mathscr{P}^{(n)}$;
while $\exists n \in \mathscr{N}, J^{(n)} \notin \varnothing$ **do**
  foreach $n \in \mathscr{N}$, $i \in T^{(n)}$, $j \in J^{(n)}$ **do**
    if $\Delta C_{ij}^{(n)} < \Delta C_{ihj}^{(n)}, h \in \mathscr{H}^{(n)}$ **then**
      $TC_{ij}^{(n)} \longleftarrow \Delta C_{ij}^{(n)}$
      $h_{ij}^{(n)} \longleftarrow \varnothing$;
    else
      $TC_{ij}^{(n)} \longleftarrow \Delta C_{ihj}^{(n)}$;
      $h_{ij}^{(n)} \longleftarrow h$
    end
  end
  $i, j, n \longleftarrow \arg\min_{i,j,n} TC_{ij}^{(n)}$;
    if $h_{ij}^{(n)} \notin \varnothing$ **then**
      Add helper h, node j into $T^{(n)}$ via node i;
      $\mathscr{H}^{(n)} \longleftarrow \mathscr{H}^{(n)} - h$;
      $t_{ih} += \tau^{(n)}$;
      $t_{hj} += \tau^{(n)}$;
    else
      Add node j into $T^{(n)}$ via node i;
      $t_{ij} += \tau^{(n)}$;
    end
    $J^{(n)} \longleftarrow J^{(n)} - j$;
end

Algorithm 1: COLS-T

5.2. Capacity Allocation (COLS-C). To achieve efficient capacity allocation, we take the scheduling delay into consideration based on the aforementioned overlay $\{T^{(n)}\}$ given by COLS-T. So we reallocate the capacity of each link to make most of the limited capacity and achieve lower cost.

In order to achieve optimal allocation, we first prove the capacity allocation is a convex problem. Then, we take advantage of classical optimization algorithm, sequential least squares quadratic programming (SLSQP) which is widely used to solve the convex optimization problem.

Considering that the overlay topology has been constructed, $\vec{x}$ is constant, i.e., Equations (3), (4), (5), and (6) are always satisfied so we can omit them. In this way, our objective is to find a vector $\vec{b}$ so that (7) get a minimum value subject to constraints (9) and (10).

We prove COLS-C is a convex problem as follows:

(i) Object function (7) is the sum of convex functions. Obviously, (7) is a convex function

(ii) Scheduling delay constraint (9) can be rewritten as:

$$f\left(\vec{b}\right) = \sum_{n}\sum_{k\in\Gamma_i} \frac{L \cdot x_{ik}^{(n)}}{b_{ik}} - \frac{L}{\tau^{\max}} \le 0, \tag{14}$$

where Γ_i is the children set of server i. $L/\tau^{\max}$ is a constant. The second-order derivative of $f(\vec{b})$ fulfills the condition of being convex:

$$\nabla^2 f\left(\vec{b}\right) \succcurlyeq 0. \tag{15}$$

Hence, $f(\vec{b})$ is a convex function.

(iii) Total delay constraint (10) is similar as (9). It can be rewritten as:

$$g\left(\vec{b}\right) = D_j^{(n)} - D = d_{s^{(n)}j}^{p} + \sum_{i\in\mathscr{A}_j^{(n)}}\sum_{k\in\Gamma_i} \frac{L}{b_{ik}} - D \le 0, \tag{16}$$

where $s^{(n)}$ is the source of channel n; $d_{s^{(n)}j}^{p}$ is a constant means the aggregated source-to-end propagation delay of server j; $A_j^{(n)}$ is the ancestor set of server j; D means the delay upper bound. $g(\vec{b})$ is also a convex function since $\nabla^2 g(\vec{b}) \succcurlyeq 0$.

As demonstrated above, object function (7) and constraints (9) and (10) are all convex, so we have proved that the capacity allocation is a convex problem. Then, we utilize *SLSQP* algorithm (*SLSQP* can be called directly from the library *SciPy*) to seek the minimum solution by iterating over the objective function (7) which denotes the sum of all link capacity costs, while satisfying the delay constraints (9) and (10).

5.3. Computational Complexity Analysis. The complexity of COLS is $O(|\mathscr{N}|^2|\mathscr{P}|^3|\mathscr{V}| + |\mathscr{E}|^3)$. In COLS-T step, one server is included into one tree at each round, and there are totally $O(|\mathscr{N}||\mathscr{P}|)$ rounds. In each round, there are $O(|\mathscr{N}||\mathscr{P}||\mathscr{V}|)$ links to be calculated. Each link costs $O(|\mathscr{P}|)$ time to select the helper, hence, adding a node in each round costs $O(|\mathscr{N}||\mathscr{P}|^2|\mathscr{V}|)$ and all rounds cost $O(|\mathscr{N}|^2|\mathscr{P}|^3|\mathscr{V}|)$.

In COLS-C step, we use *SLSQP* to solve the convex problem. It uses the Han-Powell quasi-Newton method with a BFGS update of the B-matrix and L1-test function in the step-length algorithm. It has $O(Q^3)$ overall time complexity where Q is the number of variable [25–27]. There are $|\mathscr{E}|$ variables, so the time complexity is $|\mathscr{E}|^3$. In summary, the overall complexity of COLS-{T + C} is $O(|\mathscr{N}|^2|\mathscr{P}|^3|\mathscr{V}| + |\mathscr{E}|^3)$.

6. Illustrative Experiment Results

To evaluate the performance of COLS, we have conducted extensive experiments in two aspects: popularity prediction and mathematical optimization. We present detailed experiment settings and comparison schemes in Section 4.1. Then, we illustrate results in Section 6.2.

6.1. Simulation Setup. We compare COLS with following state-of-the-art schemes:

(i) DEEPCACHE [19] builds a LSTM Encoder-Decoder model to predict the popularity of content.

In this network, the dimension of training data is single, and it only has historical playback records

(ii) CCAS [15]. Each MEC server connects the cloud server directly, and there is no collaboration. The MEC server gives up delivering some channels to save the link capacity. The link capacity is a constant in CCAS, and the scheduling delay is not considered. To meet the delay constraint, we adapt CCAS by adding a capacity allocation step. We increase the capacity iteratively with a certain proportion until it meets the constraint

(iii) BSUM [12] constructs an overlay with MEC collaboration. It considers the scheduling delay insufficiently, which ignores the correlation between different link capacities, and hence, the real scheduling delay is higher. Besides, just like CCAS, the link capacity is constant. BSUM only optimizes the topology and does not optimize the capacity. We also add a capacity allocation step which is the same as the step in CCAS

The dataset used for experimental evaluation comes from real scenes, which is provided by the telecom operator. Considering that the size of the source raw data is more than 2 TB including 931964 files (live streaming playback records) [28], we randomly choose a certain number of the records in several consecutive periods of time to construct the dataset used for evaluation. And the detailed baseline parameters are shown in Table 2.

Specifically, experiments are carried out in two parts:

(i) *Popularity prediction*. We compare COLS with DEEPCACHE. We randomly selected 10 sequences as input of designed model and output the corresponding predicted popularity. Mean square error is used to evaluate the prediction performance

(ii) *Cost optimization*. To compare COLS with CCAS and BSUM, we randomly select some MEC servers from the raw data and generate round trip time (RTT, which denotes the propagation delay) matrix among servers. For the selected servers, we use the aforementioned LSTM network to predict the popularity and get the subscribed list of each MEC servers. In order to ensure the accuracy of results, the channel that the MEC server subscribed to in CCAS and BSUM remains the same as COLS. Each scheme is evaluated 20 times and gets an average result. To evaluate the performance of each scheme, we focus on the cost metric, which is the sum of all link capacity costs

6.2. Simulation Results. To evaluate the performance of proposed approach, we have conducted extensive experiments. Figures 6 and 7 demonstrate COLS's advantages (MEC collaboration, scheduling delay consideration and capacity convex programming), which make it outperforms other schemes. Figure 6 illustrates the component proportion of maximum server delay before capacity allocation. CCAS does not consider the scheduling delay and BSUM considers insufficiently. Both of them have some missing scheduling delays which are not calculated. All their theoretical delays are lower than the bound but real delays are opposite. To meet the delay constraint (500 ms in this paper), they need to allocate more capacities to reduce the scheduling delay, which bring the cost increases. Just as mentioned above, CCAS makes MEC servers connect the cloud server directly, leading to a higher scheduling delay and a lower propagation delay. On the contrary, COLS and BSUM have collaborations, and thus, their scheduling delays are lower.

Figure 7 depicts maximum server delays after capacity allocation. It is obvious that CCAS reduces most scheduling delay and causes a highest capacity cost. BSUM uses a simple allocation algorithm, and the link capacity is redundant after

Table 2: Baseline parameters.

Parameters	Value
Number of servers	$\lvert\mathcal{V}\rvert = 50$
Number of all channels	$\lvert\mathcal{N}\rvert = 10$
Length of sliding window in COLS-P	$W = 6$
Selected parameter in COLS-P	$k = 3$
Segment size	$L = 100$kb
Dalay constraint	$D = 500$ms

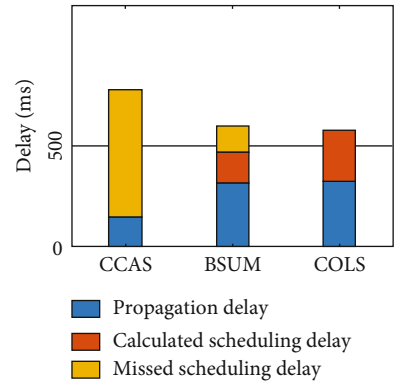


Figure 6: Max delay before reallocation.

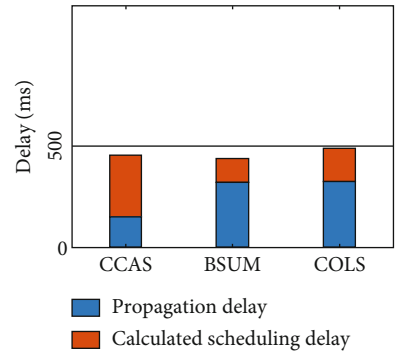


Figure 7: Max delay after reallocation.

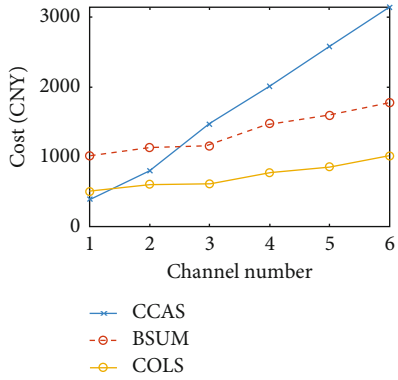


Figure 8: Cost vs. channel number.

the allocation. Hence, the scheduling delay is lower than that of COLS as shown in Figure 7. Compared with other schemes, COLS uses convex programming to allocate capacities, which is more efficiently. It makes capacities less redundant and minimizes the cost while the delay is still under the bound (500 ms).

Figure 8 shows the cost versus the channel number. The results demonstrate that COLS outperforms the other two competing schemes, and the cost of COLS is only around half of BSUM/one-third of CCAS when the channel number reaches 6. CCAS gives up delivering some live channels, i.e., users receive the live stream from the cloud server directly, which increases the cloud server scheduling delay and saves the link capacity. When the channel number is low, the overlay topology is simple, and fewer links connect the cloud server, which provides more growth space for the scheduling delay. However, as the channel number increases, the topology becomes complicated, and more links connect the cloud server. CCAS needs more capacities to reduce the scheduling delay. These capacity costs are higher than costs saved by giving up live channels. Inversely, in this case, COLS has the lowest cost because of MEC collaboration, scheduling delay consideration, and convex programming. It is more suitable for multichannel overlay, which is more common in the reality, i.e., COLS is more practicable than other schemes.

Figure 9 presents the difference results between COLS and DEEPCACHE. The mean prediction error of our method is lower than that of DEEPCACHE. This is because COLS adds more information (hours and weekday) to predict the popularity, making it a more accurate.

Figure 10 illustrates the cost versus the server number. It shows that COLS cost is lowest (outperforms others at least 40%) and increases more slowly than that of competing schemes. The reason is that COLS considers both MEC collaboration, scheduling delay, and efficient capacity allocation. In CCAS, all MEC servers connect the cloud server directly. Section 5 infers that the cloud server connects too much links, leading to a high scheduling delay. To reduce the scheduling delay, CCAS has to increase the link capacity and gets a highest cost. BSUM considers the scheduling delay insufficiently, and the real delay is beyond the bound. It has to allocate more capacity to meet the constraint. The allocate algorithm causes redundant capacities and a higher cost.

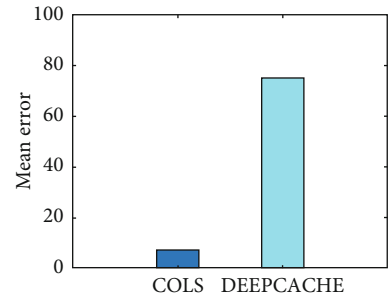


Figure 9: Mean prediction error of popularity.

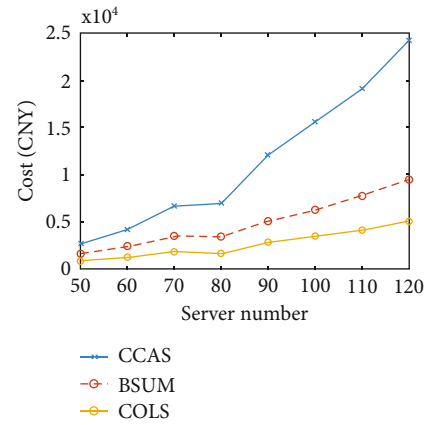


Figure 10: Cost vs. server number.

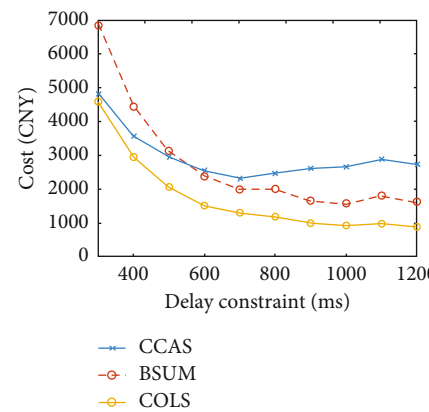


Figure 11: Cost vs. delay constraint.

We plot in Figure 11 the cost versus the delay constraint D with 50 servers. When D decreases, topology graphs constructed by COLS and BSUM gradually become similar as the graph constructed by CCAS, i.e., MEC servers connect the cloud server directly, and there is no MEC collaboration. The reason is that the propagation delay will accumulate and beyond small constraint if there are collaborations. Therefore, CCAS could give up some live channels to get a lower cost. When D increases, all costs of three schemes decrease. COLS has collaborations and convex programming to reduce the cost more quickly than others. In other words,

high-cost reductions are achieved by sacrificing a small amount of delay in COLS.

7. Conclusion

In this paper, we study the cost optimization of an overlay MEC network for live streaming. Proposed network has a more realistic delaying and topological model, where MEC servers collaborate with each other to delivery streaming. We formulate the problem and propose a framework called COLS, which predicts the popularity by LSTM model and solves optimization problem by greedy scheme and convex programming. Simulation results show that COLS has higher prediction accuracy, reduces the capacity cost by at least 40% compared with state-of-the-art schemes.

Data Availability

The source data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 6 months after publication of this article, will be considered by the corresponding author.

Disclosure

The abstract was presented at the 8th International Conference on Digital Home (ICDH) 2020.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Guangxi Innovation Driven Development Special Fund Project (AA18118039).

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] R. Pepper, *Cisco Visual Networking Index (VNI) Global Mobile Data Traffic Forecast Update*, Cisco, Tech. Rep., 2013.

[3] H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 102, pp. 847–861, 2020.

[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[5] K. Bilal and A. Erbad, "Edge computing for interactive media and video streaming," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 68–73, Valencia, Spain, 2017.

[6] P. Yuan, Y. Cai, X. Huang, S. Tang, and X. Zhao, "Collaboration improves the capacity of mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10610–10619, 2019.

[7] A. Ndikumana, S. Ullah, T. LeAnh, N. H. Tran, and C. S. Hong, "Collaborative cache allocation and computation offloading in mobile edge computing," in *2017 19th AsiaPacific Network Operations and Management Symposium (APNOMS)*, pp. 366–369, Seoul, Republic of Korea, September 2017.

[8] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, and M. Wang, "Low-rank multi-view embedding learning for micro-video popularity prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1519–1532, 2017.

[9] J. Xie, Y. Zhu, Z. Zhang et al., "A multimodal variational encoder-decoder framework for micro-video popularity prediction," *Proceedings of The Web Conference*, vol. 2020, pp. 2542–2548, 2020.

[10] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 926–937, 2019.

[11] L. Tang, Q. Huang, A. Puntambekar, Y. Vigfusson, W. Lloyd, and K. Li, "Popularity prediction of facebook videos for higher quality streaming," in *2017 {USENIX} Annual Technical Conference ({USENIX}{ATC} 17)*, pp. 111–123, Santa Clara, CA, USA, 2017.

[12] A. Ndikumana, N. H. Tran, T. M. Ho et al., "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1359–1374, 2020.

[13] D. Zhang, L. Tan, J. Ren et al., "Near-optimal and truthful online auction for computation offloading in green edge-computing systems," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 880–893, 2020.

[14] Z. Zheng, L. Song, Z. Han, G. Y. Li, and H. V. Poor, "A stackelberg game approach to proactive caching in large-scale mobile edge networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5198–5211, 2018.

[15] Y. Hung, C. Wang, and R. Hwang, "Optimizing social welfare of live video streaming services in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 922–934, 2020.

[16] Z. Zhang, R. Wang, F. R. Yu, F. Fu, and Q. Yan, "QoS aware transcoding for live streaming in edge-clouds aided hetnets: an enhanced actor-critic approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11295–11308, 2019.

[17] Y. Hung, C. Wang, and R. Hwang, "Combinatorial clock auction for live video streaming in mobile edge computing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 196–201, Honolulu, HI, USA, 2018.

[18] W. Chen, P. Chou, C. Wang, R. Hwang, and W. Chen, "Live video streaming with joint user association and caching placement in mobile edge computing," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, pp. 796–801, Big Island, HI, USA, 2020.

[19] A. Narayanan, S. Verma, E. Ramadan, P. Babaie, and Z.-L. Zhang, "Deepcache: a deep learning based framework for content caching," in *Proceedings of the 2018 Workshop on Network Meets AI & ML*, pp. 48–53, Budapest, Hungary, 2018.

[20] A.-T. Tran, N.-N. Dao, and S. Cho, "Bitrate adaptation for video streaming services in edge caching systems," *IEEE Access*, vol. 8, pp. 135844–135852, 2020.

[21] N.-N. Dao, D. T. Ngo, N.-T. Dinh et al., "Hit ratio and content quality tradeoff for adaptive bitrate streaming in edge caching systems," *IEEE Systems Journal*, vol. 15, no. 4, pp. 1–4, 2020.

[22] J. Dai, Z. Chang, and S.-H. G. Chan, "Delay optimization for multi-source multi-channel overlay live streaming," in *2015 IEEE international conference on communications (ICC)*, pp. 6959–6964, London, UK, 2015.

[23] J. Hartmanis, "Computers and intractability: a guide to the theory of np-completeness (Michael R. Garey and David S. Johnson)," *SIAM Review*, vol. 24, no. 1, pp. 90-91, 1982.

[24] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees," *SIAM Journal on Applied Mathematics*, vol. 16, no. 1, pp. 1–29, 1968.

[25] D. Kraft, "Algorithm 733: TOMP–Fortran modules for optimal control calculations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 20, no. 3, pp. 262–281, 1994.

[26] M. M. Alves, R. Monteiro, and B. F. Svaiter, *Primaldual Regularized SQP and SQCQP Type Methods for Convex Programming and Their Complexity Analysis*, Preprint, 2014.

[27] D. Kraft, *A Software Package for Sequential Quadratic Programming*, Open Grey, 1988.

[28] N. Liu, H. Cui, S.-H. G. Chan, Z. Chen, and Y. Zhuang, "Dissecting user behaviors for a simultaneous live and vod iptv system," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 10, no. 3, pp. 1–16, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5579090, 13 pages
https://doi.org/10.1155/2021/5579090

Research Article

Cascading and Residual Connected Network for Single Image Superresolution

Kai Huang[1], **Wenhao Wang**[1], **Cheng Pang**[1], **Rushi Lan**[1,2], **Ji Li**[1,3] and **Xiaonan Luo**[2,3]

[1]*Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*National Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*Guilin Huigu Institute of Artificial Intelligence Industrial Technology, Guilin 541004, China*

Correspondence should be addressed to Cheng Pang; pangcheng3@guet.edu.cn and Rushi Lan; rslan2016@163.com

Received 7 January 2021; Revised 25 July 2021; Accepted 3 August 2021; Published 21 October 2021

Academic Editor: Xiaojie Wang

Convolution neural networks facilitate the significant process of single image super-resolution (SISR). However, most of the existing CNN-based models suffer from numerous parameters and excessively deeper structures. Moreover, these models relying on in-depth features commonly ignore the hints of low-level features, resulting in poor performance. This paper demonstrates an intriguing network for SISR with cascading and residual connections (CASR), which alleviates these problems by extracting features in a small net called head module via the strategies based on the depthwise separable convolution and deformable convolution. Moreover, we also include a cascading residual block (CAS-Block) for the upsampling process, which benefits the gradient propagation and feature learning while easing the model training. Extensive experiments conducted on four benchmark datasets demonstrate that the proposed method is superior to the latest SISR methods in terms of quantitative indicators and realistic visual effects.

1. Introduction

Superresolution (SR) image reconstruction is widely used in various applications, such as military surveillance, medical diagnostics [1, 2], remote sensing [3], and video streaming [4, 5]. Single image superresolution (SISR) is aimed at reconstructing a high-resolution (HR) image from its counterpart low-resolution (LR) input, which is an essential and classic task in computer vision. Recently, high-resolution (HR) demand images have boosted. However, physical constraints limit the conduction of high-resolution pictures. A series of successful works brought attention to the research community.

The task of recovering HR images I^{SR} from its counterpart (LR) version I^{LR} is ill-posed. Researchers have made many efforts to this task and invented numerous algorithms, including interpolation-based, reconstruction-based, and learning-based methods [1], respectively.

The traditional SISR algorithms, for instance, bicubic interpolation [6], are high-speed while suffering from poor accuracy. It is easy to fail in practice. To limit possible solving space, researchers present more advanced methods, reconstruction-based algorithms [7, 8], by introducing available prior knowledge. These algorithms may restore clear details (i.e., texture details), but extensive experiments show that they degrade sharply when the scale factors increase; subsequently, the algorithms with learnable parameters [9] are proposed to analyze relationships between the I^{LR} image and their counterpart I^{HR} image by training concrete instances [10, 11]. Although such learning-based methods perform very well, the time-consuming optimization problems they involve are very tricky.

In recent years, CNNs have been introduced to facilitate the progress of the SISR field because of their excellent feature representation ability. Dong et al. [12] were the first to propose a three-stage convolutional network to solve the

SISR problem, which has become a milestone in this field. Since then, the research community has set out to design more complex networks to improve performance. EDSR, a very large network with residual blocks, was presented by Lim et al. [13] and achieved satisfactory performance in both PNSR and SSIM [14]. However, these state-of-the-art methods still have some limitations:

(1) The state-of-the-art (SotA) models [13] mainly improve the performance by considerably growing the depth and width of the proposed methods. Therefore, massive parameters and increasing resource-consuming problems are inevitable

(2) Many progressive models do not fully take advantage of the hierarchical information from the primary LR images, which are essential for improving visual performances

To address these shortcomings, we present a model named CASR, exploring two separate strategies to functionally extract features for precise SISR. Figure 1 shows the ×4 SR results of our proposed model on dataset DIV2K [15]. First, we propose a small but functional depthwise separable convolution network named head module aimed at more systematic feature extraction.

Second, we present another cascaded residual network (CAS-Block) for better feature and gradient propagation. Our proposed method combines features from excessive layers at both the regional and global levels with such architecture. Moreover, a stacking broader local residual connection is applied to exploit the feature of the I^{LR} and let the vast low-level mappings be transmitted. This schema unites nonlocal actions to capture remote spatial features from former inputs.

As the crucial integrant of the presented method, the CAS-Block includes six subtrunks, each of which consists of two convolutional layers and a nonlinear activation pReLU. Because using the activation function in bottlenecks does affect the performance, we take advantage of channels before the pReLU layer to construct the inverse residual block, resulting in performance improvement.

The three main contributions of this article are summarized as follows.

(1) We propose a head module applied with a series of depthwise separable convolution operations for feature extraction. In addition, we replace all existing conventional convolution operations with deformable convolution layers in the module. At the same time, in order to effectively retain the features, we extend the low-dimensional representation to high-dimensional before passing the activation function. This maintains a balance between a large number of parameters and excellent performance

(2) In order to effectually raise feature fusion and gradient propagation, we introduce a cascaded block called CAS-Block. This mechanism allows our network to combine features from diverse layers. Furthermore, such a structure is also used to construct the network and promote its functional expression

(3) We utilize the L_1 with the addition of total variance loss $\mathscr{L}_{TV}$ instead of the traditional sole L_1 loss function, which significantly improves the quality of the reconstruction image I^{SR}. Meanwhile, to obtain better optimization weights, we explored various parameter settings

2. Related Works

2.1. SISR Using Deep CNNs. In the field of superresolution, compared to conventional image restoration methods, CNN-based models have a stronger feature expression ability and have achieved great success. Dong et al. [12] first proposed an algorithm named SRCNN, which is an end-to-end algorithm based on CNN. It consists of three convolutional layers, and its performance is impressive compared to traditional methods (i.e., sparse coding [7] and bicubic interpolation [6]). Later, the research community designed more intricate CNN architectures and developed more profound networks. For example, in order to grow the depth of the network, VDSR [16] introduced residual learning, and the verification experiment proved that this strategy heightens the SR image qualities and promotes convergence. DRCN [17] uses deep recursion to construct a neural network and uses the same convolution kernel 16 times in the reference network, effectually dropping the number of parameters. He et al. [18], inspired by the ordinary differential equation (ODE), propose an intriguing network named OISR, which provides a new understanding of network designs. It is worth noting that most of these latest methods use interpolated images for input, which will not only cause the details to be too smooth but also boost additional computational cost and time consumption.

2.2. Skip Connection. ResNet [19] was the first to adopt the concept of skip connection, and then, the idea was extended to various computer vision tasks, such as image restoration [20] and semantic segmentation [21]. Since it is difficult for ordinary SR networks to construct extremely deep networks, employing various skip connections avoids the gradient vanishing trap and boosts performance. The strategy is roughly divided into two categories, namely, global or local residual connections and dense connections.

2.2.1. Global or Local Residual Connections. In image restoration tasks, LR images are closely related to HR ones. Obtaining the residual feature maps among the image's pixels can learn the absent high-frequency detailed information. VDSR is the first residual model for superresolution. Extensive experiments have proved that residual learning can enhance reconstruction performance and promote convergence speed. Therefore, this method has been widely used in various computer vision tasks [22].

2.2.2. Dense Connections. A dense connection allows the current layer to connect with all former layers, and the architecture provides more intriguing effects on restoring high-

(a) (b)

Figure 1: ×4 superresolution results of our proposed SR model on dataset DIV2K.

resolution patterns. DenseNet [23] first presents the dense connection in the SR field, starting from the features, achieving better results with fewer parameters through the extreme use of the features.

Traditional neural networks are basically unidirectional propagations, and the signals received in the later layers are very weak. To solve this problem, MemNet [20] stacks memory blocks and adds dense connections among each block, which is called the long-term memory model. Such architecture reduces the weight of the entire network, facilitates convergence, and deepens the network.

RDN [24] uses a similar architecture, but MemNet does not take all the intermediate feature information, while RDN applies global residual learning to use all of them.

Jiang et al. [25] proposed a hierarchical dense network (HDRN) in 2019, which can effectively establish realistic mapping relationships between the LR and HR image, promoting information interaction and representation.

Different from the above models, CARN also uses a cascade mechanism at the local and global levels to integrate features from multiple layers, which can reflect input representations at different levels for receiving more information [26]. Haris et al. [27] proposed D-DBPN, which connects the features of the up- and downsampling stages and improves the SR result.

2.3. Depthwise Separable Convolution. The cross-channel correlation and spatial correlation of the convolutional layer can be decoupled, and they can be mapped separately to achieve better results. Some lightweight networks, such as MobileNet [28], apply depthwise separable convolution, which is a combination of depthwise (DW) and pointwise (PW) to extract feature maps. Compared with the conventional convolution operation, the number of parameters and operation cost are relatively small. In Figure 2, we use the separable convolution operation in the depth direction in the head module.

2.4. Multiscale Learning. So as to utilize computing resources more efficiently and extract more features under the same amount of calculation, Szegedy et al. [29] present the inception module. There are two main contributions of the inception structure: one is to use 1×1 convolution to perform dimensionality reduction; the other is to simultaneously perform convolution and reaggregation on multiple sizes. Inspired by [29] and [30], MSRB [31] was proposed. Multiscale feature fusion and local residual learning can be applied to adaptively detect images of different scales with different sizes of convolution kernel features. The results show that performing different kernel operations can provide better extraction capabilities. However, this method cannot expand more receptive fields and cannot generate more detailed structural information.

2.5. Deformable Convolution. Conventional convolution kernels are usually of fixed size (for example, $3 \times 3, 5 \times 5$, and 7×7.). The biggest problem with this convolution kernel is that it has poor adaptability to unknown changes and weak generalization ability. In order to solve the object space deformation problem, deformable convolution [32] is proposed to heighten the transformation modeling ability of CNN. Deformable convolution is based on traditional convolution, adding the direction vector of the adjustment convolution kernel to make the shape of the convolution kernel closer to the feature.

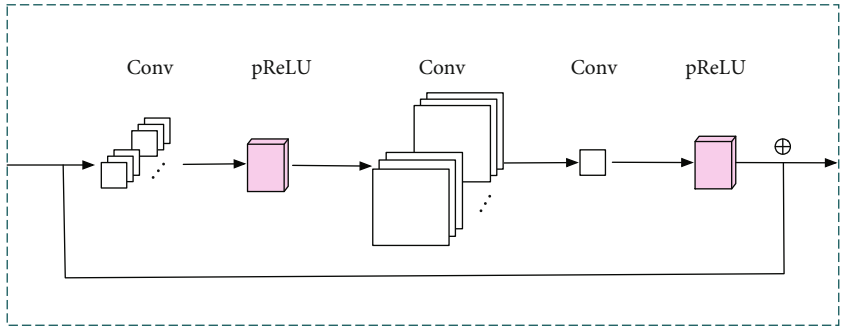


Figure 2: The architecture of head module is abundantly applied with depthwise separable convolution and deformable convolution operations. ⊕ stands for the element-wise additional operation.

2.6. Real-World Image Superresolution. In real-world image restoration scenarios, lacking corresponding high-quality references usually conduct poor experimental results. We additionally introduce the naturalness image quality evaluator (NIQE) [33] and Perceptual Index (PI) [22] to perform the evaluation. In fact, these indicators can sensitively reflect content sharpness, detail contrast, and texture diversity. These evaluation indexes have a high consistency with the subjective quality and can effectively reflect the visual quality of images without reference. In particular, the smaller values of NIQE/PI indicate better perceptual quality and clearer content. We intend to apply it to the DRIVE [34] dataset to estimate the restoration capability of the proposed method.

3. Proposed Approach

3.1. Network Architectures. SISR's algorithm, such as ESPCN [35] and FSRCNN [36], does not take full advantage of low-level feature information. With a deeper structure, there are more parameters. As shown in Figure 3, the proposed CASR consists of three components: (1) head module, (2) cascading block, and (3) upsample module. All we want is the balance between the performance and the cost.

To better explore the mentioned issues, we adopt two different strategies: (1) original feature extraction and (2) cascading connection structure.

3.1.1. Original Feature Extraction. We depict I^{LR} and I^{SR} as the input and output of our models, respectively. Figure 2 illustrates how the head module extracts the original information from LR images:

$$F_{\mathrm{ext}} = H_{\mathrm{ext}}\left(I^{\mathrm{LR}}\right), \tag{1}$$

where $H_{\mathrm{ext}}(\cdot)$ means a series of convolution operations. In the head module, we first replace the conventional one with a depthwise convolution layer for reducing parameters. Through an activation layer, the feature maps are sent to another specific convolution layer, deformable convolution. As we discussed in Section 2, deformable convolution adds an offset to each convolution sampling point, thus achieving free deformation of the sampling grid. Then, after passing through a specific convolutional layer with 1×1 kernel and another pReLU activation function, F_{ext} is sent to the next stage for a higher-level abstraction.

3.1.2. Cascading Connection Structure. Now, we present the CAS-Block. The cascade connection allows information to spread across multiple paths in the network, which greatly enhances feature fusion. It [10] has been widely applied in various computer vision tasks. In Figure 4, the mapping process of our cascade network includes C CAS-Blocks, each with a skip connection:

$$H_{\mathrm{map}_0} = C_0(F_{\mathrm{ext}}), \tag{2}$$

$$H_{\mathrm{map}_i} = C_i\left(H_{\mathrm{map}_0}\right), \tag{3}$$

where H_{map_i} presents the output of the C_ith CAS-Block. Each CAS-Block contains one group convolution layer (with 3×3 or 1×3 kernel), one traditional convolution layer for adjusting the number of channels, and a pReLU layer. We prefer stacking several kernels with smaller sizes (such as 1×3 and 3×3) to directly applying larger kernels (such as 5×5 and 7×7) for enlarging the receptive field of the feature extraction module and decreasing the number of learnable parameters:

$$H_{\mathrm{middle}} = C_{m+1}\left(H_{\mathrm{cas}}\left(\left(\left(H_{\mathrm{map}_m} + H_{\mathrm{map}_{m-1}}\right), \left(H_{\mathrm{map}_m} + H_{\mathrm{map}_{m-2}}\right), \cdot \left(H_{\mathrm{map}_{m-1}} + H_{\mathrm{map}_{m-2}}\right)\right)\right)\right), \tag{4}$$

where $H_{\mathrm{map}_m}, H_{\mathrm{map}_{m-1}}, H_{\mathrm{map}_{m-2}}$ means all the outputs of the middle three CAS-Blocks. H_{cas} denotes the cascading operation:

$$H_{\mathrm{map}} = C_{m+2}(C_1 + H_{\mathrm{middle}}), \tag{5}$$

where $H_{\mathrm{map}}(\cdot)$ demotes our proposed mapping function. Finally, we use a common upsampling module to fuse the hierarchical structural features and amplify the image size:

$$F_{\mathrm{up}} = H_{\mathrm{up}}\left(H_{\mathrm{map}}\right), \tag{6}$$

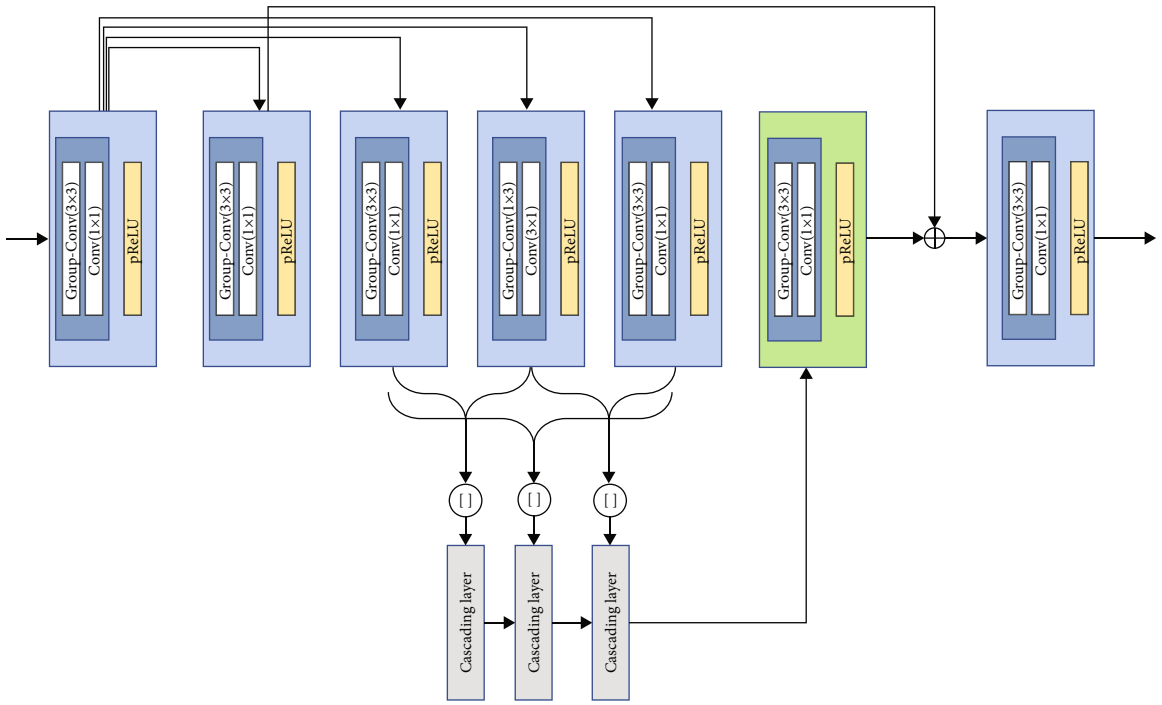


FIGURE 3: Network design for our model. The green cuboid means the head module; the yellow one represents the cascading block, and the last blue rectangle depicts the upsampling process.

where $H_{\text{up}}(\cdot)$ indicates an upscale module. In recent years, many upsampling methods have been proposed, such as [12, 27, 36]. We adopt the postupsampling method, which has been proven effectively outstanding. The process of our model roughly includes three steps. First, taking the low-resolution I^{LR} image as the original input, the feature extraction module obtains the initial features from the low-quality image. Then, these features are delivered to a higher abstraction layer. Finally, we adopt a simple upsampling block, including a convolutional layer, and a pixel-shuffle layer to enlarge the SR image.

3.2. Total Variation Loss. Aly and Dubois bring the total variation (TV) [37] loss to the SR field in order to suppress noise in generated images, and for imposing spatial smoothness, Yuan et al. also select this TV loss:

$$\mathscr{L}_{\text{TV}}\left(\widehat{I}\right) = \frac{1}{hwc}\sum_{i,j,k}\sqrt{\left(I\wedge_{i,j+1,k} - I\wedge_{i,j,k}\right)^2 + \left(I\wedge_{i+1,j,k} - I\wedge_{i,j,k}\right)^2}, \tag{7}$$

where $\widehat{I}$ depicts the reconstructed HR image, h, w, represent the dimensions of the corresponding feature maps, and c symbolizes the number of channels. On the other hand, although mean square error (MSE) is available, previous work [38] proved that it is not a good choice. Thus, the second loss function is defined as follows:

$$\mathscr{L} = \mathscr{L}_1 + \lambda\mathscr{L}_{\text{TV}}. \tag{8}$$

We applied these loss functions in the training process of our presented model. From the experiment, we found that adopting the $\mathscr{L}$ loss compared with the simple $\mathscr{L}_1$ loss, the model achieves better performance, and set $\lambda = 1e^{-4}$ works well. As shown in Figure 5, the loss $\mathscr{L}$ enables the network to generate smoother recovery images, and Figure 6 comparatively illustrates that the combined loss function may produce sharper SR results.

3.3. Comparison with Other CNN-Based Methods

3.3.1. Comparison with MSRN. Compared with MSRN, our CASR is different as follows. First, the basic module design is distinct. In MSRN, the multiscale residual block (MSRB) incorporates parallel convolution with multiple feature channels. The output of each multiscale residual block is cascaded together through a hierarchical feature fusion to produce the final result, which leads to a lot of calculations. However, our multiscale modules are branch-based, using regional skip connections and cascades extensively, scaling down parameters. Second, it is the difference in the activation function. MSRN utilizes the ReLU function, while we employ PReLU as the activation function. According to the comparison in Figure 7, PReLU optimizes and improves ReLU. Under the premise of almost no increase in the amount of calculation, the PReLU function effectually improves the overfitting problem of the model, accelerates the convergence, and lowers the error. Therefore, our proposed multiscale module owns more effective representation capabilities.

3.3.2. Comparison with MemNet. We summarize the main differences between MemNet [20] and our CASR. The

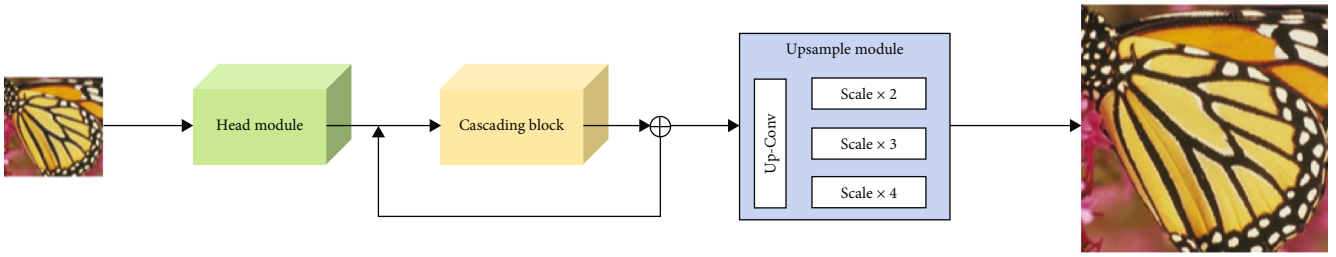


FIGURE 4: The architecture of cascading block (CAS-Block). ⊕ operator denotes the element-wise additive operation and [ ] means cascading operation.

FIGURE 5: ×3 SR's loss function comparison. In the first row, it is the *comic* images in the Set14 dataset. The image processed with $L_1 + L_{TV}$ has precise details in the area around the eyes. The bottom line is the "img020" images in the Urban100 benchmark dataset. This method applies the $L_1 + L_{TV}$ method to reconstruct the clear details, such as the windows.

former employs stacking memory blocks and massive shortcuts, while our method avoids extensive dense connections for lowering the number of parameters. What is more, Lim et al. trained their network with the L_2 loss, but we prefer L_1 loss to L_2 loss function. Besides, MemNet regards the interpolated images as input. Contrastively, our proposed method directly extracts hierarchical features from the original LR images upsampled at the end of the process for computational efficiency and SR performance improvement.

4. Experimental Results

4.1. Training Details. We set depthwise separable convolution operations in head module shown in Figure 2, which were first illustrated in the Inception net in the proposed model, and were able to reduce the size of the network parameters effectively. Figure 4 graphically illustrates the cascading process occurring. The medial layers' outcomes are cascaded into the posterior layers and finally assemble in a convolutional trunk consisting of a depthwise separable convolutional operation with three times the input and output features and then thorough a pReLU activation function.

We prefer $L_1 + \mathscr{L}_{TV}$ to L_2 loss as the loss function, though the latter has been generally applied in computer vision tasks because of its intimate relation with PSNR's calculation. However, the research community recently indicates that L_1 loss provides better accuracy and faster convergence; TV loss ($\mathscr{L}_{TV}$) imposes spatial smoothness on reconstructed images. Specifically, we set training patches with a size of 128×128, and batch size = 16. We employ

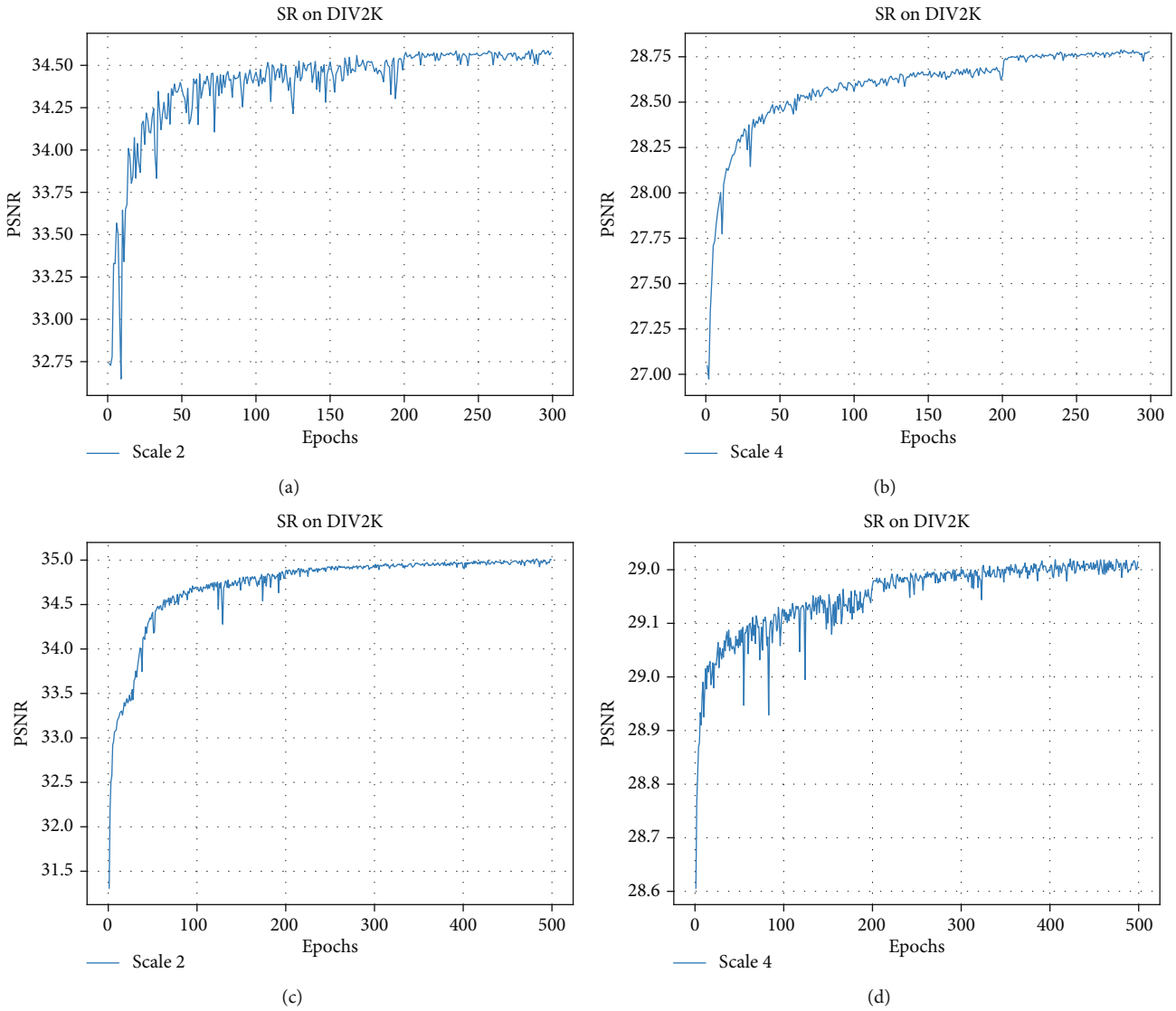


Figure 6: PSNR (dB) of training process on scales ×2 and ×4 with L_1 and $L_1 + \mathscr{L}_{TV}$ loss, respectively.

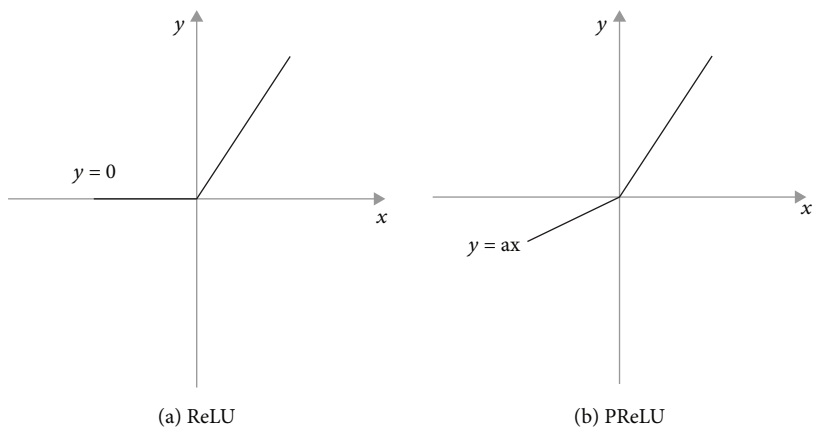


Figure 7: Comparison: ReLU versus PReLU. PReLU optimizes and improves ReLU. Under the premise of almost no increase in the amount of calculation, the overfitting problem of the model is effectively improved. The convergence is faster, and the error is lower.

Table 1: Public benchmark test results (PSNR (dB)/SSIM). Bold illustrates the best result, and the underline indicates the second-best ones.

Method	Scale	Params	Multi-adds	Set5 [40]	Set14 [41]	B100 [42]	Urban100 [43]
SRCNN [12]	×2	57K	52.7G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
VDSR [16]	×2	665K	612G	37.52/0.9586	33.01/0.9123	31.89/0.8960	30.76/0.9140
LapSRN [44]	×2	813K	29.9G	37.51/0.9590	33.07/0.9129	31.89/0.8959	30.40/0.9100
DRCN [17]	×2	1774K	17974G	37.43/0.9587	33.23/0.9137	31.29/0.8911	30.22/0.9023
DRRN [8]	×2	297K	6796.9G	37.74/0.9523	32.79/0.9121	31.67/0.8900	30.14/0.9112
MemNet [20]	×2	677K	2662.4G	37.83/0.9623	33.15/0.9192	31.99/0.9000	30.51/0.9177
RDN [24]	×2	22.12M	5096.2G	38.30/0.9616	**34.10/0.9218**	**32.40/0.9022**	**33.09/0.9368**
HDRN [25]	×2	—	—	37.75/0.9590	33.49/0.9150	32.03/0.8980	31.87/0.9250
OISR [18]	×2	41.91M	9656.5G	37.98/0.9604	33.58/0.9172	32.18/0.8996	32.09/0.9281
IDN [45]	×2	590K	174.1G	38.10/0.9601	33.91/0.9194	32.31/0.9012	32.92/0.9269
CASR (ours)	×2	501K	113G	**38.49/0.9607**	33.97/0.9204	32.33/0.9017	32.90/0.9244
SRCNN [12]	×3	57K	52.7G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7963
VDSR [16]	×3	665K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
LapSRN [44]	×3	813K	29.9G	33.82/0.9207	29.89/0.8304	28.82/0.7950	27.07/0.8298
DRCN [17]	×3	1.77M	17974G	33.84/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
DRRN [8]	×3	297K	6796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
MemNet [20]	×3	677K	2662.4G	34.09/0.9590	30.07/0.8429	29.89/0.8110	28.41/0.8610
RDN [24]	×3	22.31M	2281.2G	**34.78/0.9300**	**30.67/0.8482**	**29.33/0.8105**	**29.00/0.8683**
HDRN [25]	×3	—	—	34.24/0.9240	30.23/0.8400	28.96/0.8040	27.93/0.8490
OISR [18]	×3	44.86M	4590.1G	34.43/0.9273	30.33/0.8420	29.10/0.8053	28.20/0.8534
IDN [45]	×3	590K	105.6G	34.46/0.9282	30.52/0.8462	29.25/0.8093	28.80/0.8653
CASR (ours)	×3	597K	52G	34.49/0.9297	30.57/0.8450	29.29/0.8099	28.90/0.8704
SRCNN [12]	×4	57K	52.7G	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221
VDSR [16]	×4	665K	612.6G	31.35/0.8830	28.02/0.7680	27.29/0.7226	25.18/0.7540
LapSRN [44]	×4	813K	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560
DRCN [17]	×4	1774K	17974G	31.53/0.8846	28.02/0.7670	27.23/0.7233	25.14/0.7510
DRRN [8]	×4	297K	6796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
RDN [24]	×4	22.27M	1309.2G	32.61/0.9003	28.92/0.7893	**27.80/0.7434**	**26.82/0.8069**
HDRN [25]	×4	—	—	32.23/0.8960	28.58/0.7810	27.53/0.7370	26.09/0.7870
OISR [18]	×4	44.27M	2962.5G	32.21/0.8950	28.63/0.7822	27.58/0.7364	26.14/0.7874
MemNet [20]	×4	677K	2662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7286	25.50/0.7630
IDN [45]	×4	590K	81.8G	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033
CASR (ours)	×4	597K	51G	**32.67/0.9006**	**28.96/0.7899**	27.77/0.7428	26.67/0.8057

16000 iterations of back-propagation per epoch; we adopt the ADAM [39] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$ for optimization. We set 850 training epochs and the learning rate of all layers to 1×10^{-4} initially, which will be reduced to half for every 50 epochs. All experiments are run under the PyTorch framework and deployed on NVIDIA RTX 2080Ti GPU.

4.2. Datasets. DIV2K [15] is a high-definition dataset containing various image contents. It has 800 training images, 100 verification images, and 100 test images. We employ 800 training images to train the proposed model and randomly select ten validation images as evaluation. In the testing process, we adopt the following benchmark datasets as test datasets: Set5 [40], Set14 [41], B100 [42], and Urban100 [43]. They contain various scenes in real life, such as landscapes, buildings, and people, while the Digital Retinal Images for Vessel Extraction (DRIVE [34]) dataset is a dataset for retinal vessel segmentation. It consists of a total of JPEG 40 color fundus images, including 7 abnormal pathology cases.

4.3. Experimental Analyses

4.3.1. Results on Benchmark Datasets. Our proposed method will be compared with the state-of-the-art SR model on two commonly adopted image quality metrics (i.e., PSNR and SSIM). We analyze our methods with several progressive networks: (1) bicubic, (2) SRCNN [12], (3) VDSR [16], (4) LapSRN [44], (5) DRCN [17], (6) DRRN [8], (7) MemNet [20], (8) RDN [24], (9) HDRN [25], (10) OISR [18], and (11) IDN [45]. As described in the technical literature, these methods were evaluated on four aforementioned datasets. Table 1 lists the performance of all mentioned algorithms.

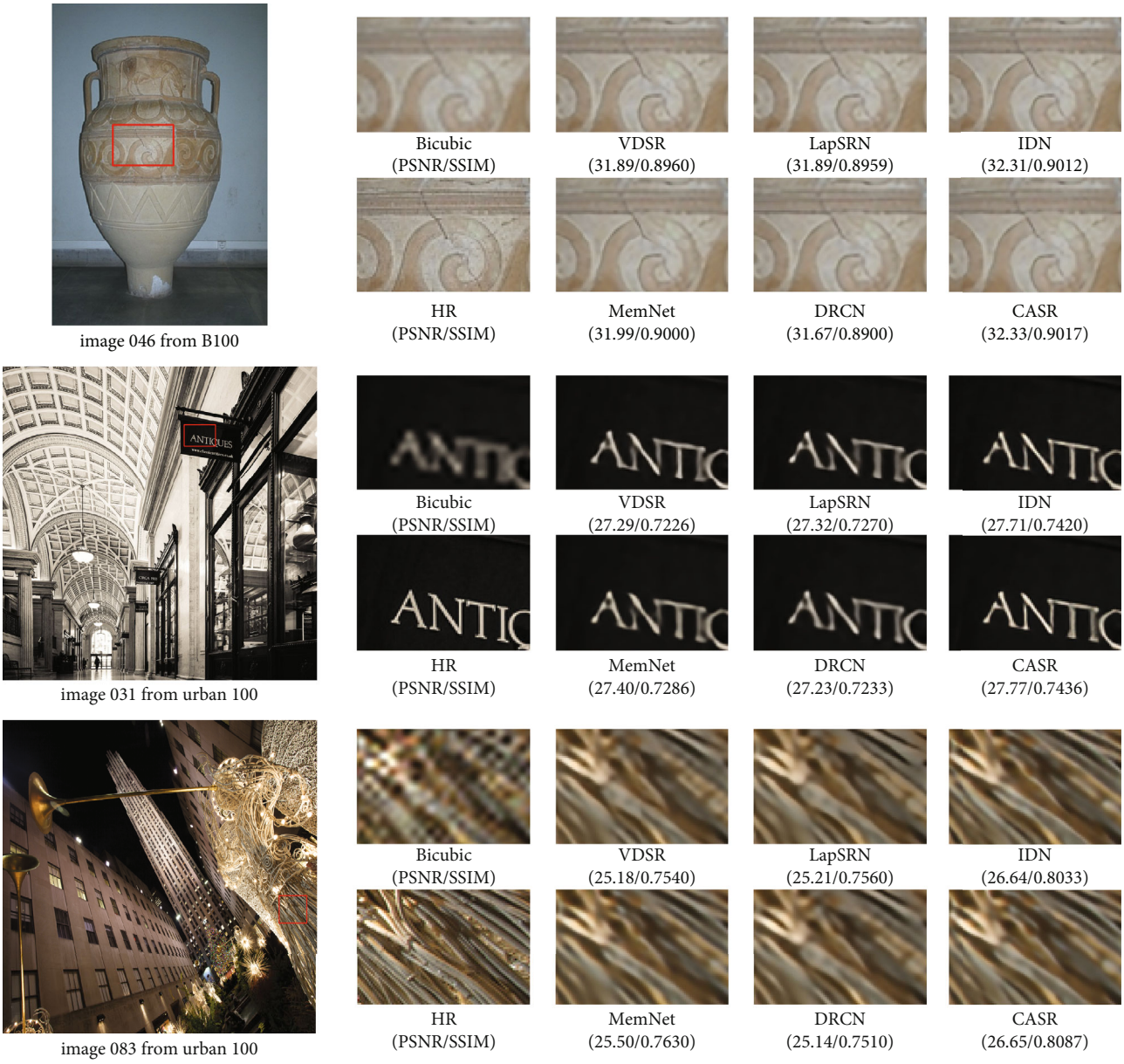


Figure 8: Comparison of reconstructed image details on the benchmark dataset. From top to bottom, the ×2 superresolution experiment on B100, the ×3 and ×4 superresolution experiment on Urban100. Our presented method reconstructs SR images with richer details.

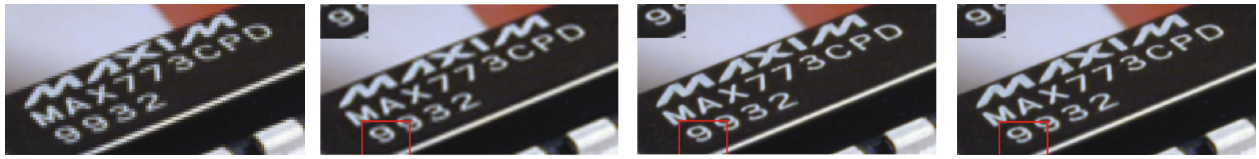
Table 2: Time consumption and experimental performance in dataset Set14 with the scale 4.

Method	Scale	Params	Multi-adds	Runtimes (s)	PSNR/SSIM
VDSR [16]	×4	665K	612.6G	0.4523	28.02/0.7680
RDN [24]	×4	22.27M	1309.2G	0.2114	28.92/0.7893
CASR (ours)	×4	597K	51G	0.1017	28.96/0.7899

Our networks are much better than the comparison model in variant scale factors except for RDN. On some datasets, the performance of CASR is completely close to RDN, while the consumption of RDN is much larger in the meantime. We will particularly discuss it later.

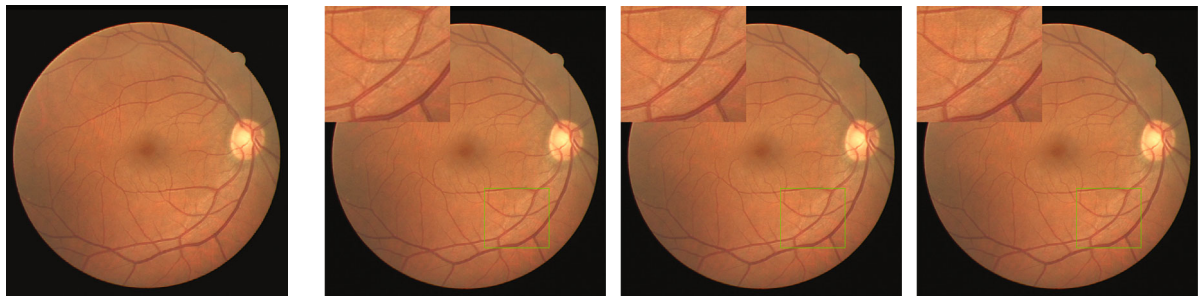
Table 3: Average NIQE and PI values on public dataset Set14 with scale factor ×4.

Metrics	SRCNN [12]	VDSR [16]	IDN [45]	CASR (ours)
NIQE [33]	6.624	5.744	6.472	4.506
PI [22]	5.929	5.121	5.448	3.859



(a) LR Image (b) Bicubic (c) IDN [50] (d) CASR (Ours)

Figure 9: Visual image comparisons with different SR methods on real-world image *chip* with scale ×2.



(a) LR Image (b) Bicubic (c) RDN [18] (d) CASR (Ours)

Figure 10: The qualitative comparisons between different SR methods on the digital retinal image for vessel extraction (DRIVE) dataset with scale ×4.

Compared with other methods, on dataset Set5, our proposed method performs better at all scales. Especially on the ×2 superresolution, the reconstructed image contains very clear texture details.

Our method performs well in image superresolution restoration tasks of various scales on the dataset Set14. Specifically, the best performance of the benchmark SR model is RDN [24] and IDN [45], which reach 30.67/0.8482 and 30.52/0.8462 on PSNR/SSIM metrics, respectively, at the ×3 scale; our model is 0.1 db lower than the former and 0.05 db higher than the latter.

As mentioned earlier, the dataset B100 contains many real-world images. As seen in Figure 8, the vase image recovered by our method has clearer edges, reaching 32.33 db. Figure 8 demonstrates visual comparisons on dataset B100 and Urban100 with scales ×2 and ×4, respectively.

The Urban100 dataset consists of 100 pictures of various buildings, which usually contain clear edges and rich textures. So, according to [24], RDN is expected to perform well on superresolution tasks, reaching 33.09 db on ×2. Our method acts well at ×3 and ×4 superresolution tasks, reaching 28.90 db and 26.67 db, respectively, which is approximately 0.1 db and 0.15 db lower than RDN, while CASR costs much less than the competitors.

4.3.2. Comparison Results on Time Complexity. Besides, we have provided a comparison of the model's efficiency in terms of time complexity on public dataset Set14 (taking ×4 as an instance), as tabulated in Table 2. The table intuitively shows that the CASR model achieves a similar competitive experiment result compared to VDSR [16] and RDN [24], reaching 28.96/0.7899 on PSNR/SSIM metrics, while spending less time (0.1017 s on a single image) and costing the least resource on processing image restoration. We may conclude that our proposed CASR model takes the least time consumption and adopts acceptable parameters compared with VDSR and RDN.

4.3.3. Superresolution on Real-World Images. Table 3 indicates that our proposed CASR method is highly competitive, achieving the lowest average values of NIQE/PI on benchmark dataset Set14 with scale factor ×4. Figure 9 illustrates the visual image restoration comparison with several SR methods on the real-world image chip. Results visually show that our method, compared to others, achieves better restorative performance. It not only achieves competitive PI and NIQE values but also improves more pleasant visual quality in terms of image, edge, texture, color, and feature-rich regions. Besides, as shown in Figure 10, the restorative performance on the larger scale, e.g., ×4, is also acceptable. The vessel in the retinal image is more clear than the competitors, and the edge of the retina is also sharp as we expected. Considering that the whole experiment was designed and conducted on dataset DIV2K, a supervised public dataset with ground truth images, which is acceptable and compromised, we believe that could provide a further research direction, exploring a more realistic oriented image SR process with a better degradation kernel on a real-world image dataset.

4.4. Ablation Study. In order to further explore the details of the experiment, we design 2 ablation experiments: one is to investigate the influence of different dilation factors on deformable convolution, and the other is the experiment of different loss functions' effects.

4.4.1. Study of the Deformable Convolution. Figure 11 illustrates two training processes with variant dilation scales. We examine whether the dilation scale of deformable

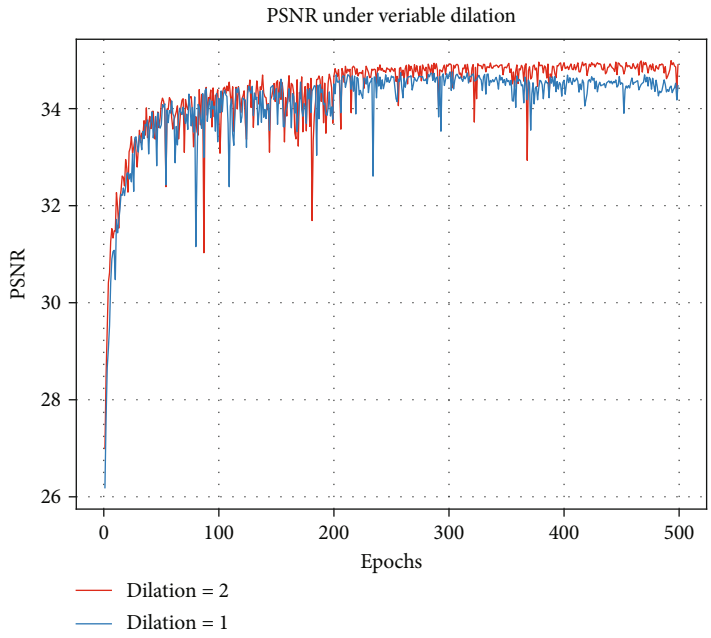


Figure 11: Comparable training results (PSNR) with scale factor ×2 under variable dilations; blue result illustrates the training process without dilation while red one shows deformable convolution with dilation 2.

Table 4: The results of ablation studies on deformable convolutions with distinct dilation factors on dataset Set5 and B100.

Scale	DeformConv	Dilation	Set5 (PSNR)	B100 (PSNR)
2	×	—	37.34	31.83
	✓	1	37.54	30.99
	✓	2	33.97	31.90
4	×	—	31.95	26.76
	✓	1	32.32	26.64
	✓	2	32.53	26.44

Table 5: Effect of the variant loss functions on the Set14 and Urban100 benchmark datasets, with a scale factor of ×2 and ×4, respectively.

Scale	Loss function	Set14	Urban100
2	L_1	33.38	31.62
	$L_1 + L_{TV}$	33.97	31.90
4	L_1	31.95	25.76
	$L_1 + L_{TV}$	28.97	26.65

convolution would affect recovery performance or not. As is shown in Figure 11, with epochs rising, both training results grow as well, while the model with dilation two would achieve better performance but cause a worse fluctuation. We also compare the effect of different scale factors on the experimental performance, as shown in Table 4. It can be learned that with the same scale factor ×2, our proposed method, which replaces the convenient convolution with deformable convolution, would achieve better results on both datasets Set5 and B100. With the dilation factor enlarging, the performances go better. This result mainly occurs since the operation may effectively and dynamically expand the receptive field. Because different input feature maps may correspond to objects with different deformation scales, for some tasks, it is essential to adaptively determine the ratio or receptive field size.

4.4.2. Study of the Loss Function. To examine the effect of the mentioned loss functions, we design an ablation experiment to explore it. Expressed formally, let the first model be "L_1" and the other one be $L_1 + L_{TV}$ (i.e., using the enhanced loss function $L_1 + L_{TV}$). We tried different combinations of scale factor and loss function to examine which one would achieve better performance on dataset DIV2K, as demonstrated in Figure 6 and Table 5. Afterward, the validation process on dataset Set14 and Urban100 proves that the enhanced loss function actually results in a clearer image with more details in Figure 5.

5. Conclusion

This paper presents two novel CNN architectures, namely, head module and CAS-Block, to improve the SISR performance. Compared with the state-of-the-art (SotA) CNN-based algorithms, our presented head module considers low-level feature expression by applying depthwise separable convolution and deformable convolution, which is demonstrated to not only effectively extract the patterns but also reduce the parameter size. At the same time, the CAS-Block employs a global residual connection and abundantly utilizes cascading connections to capture remote spatial features from former inputs. Extensive experiments have

illustrated that our presented model has effectively improved both the quality of the reconstructed images and the processing speed compared with the SotA methods in terms of quantitative indicators and realistic visual effects.

Data Availability

The image data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. U1701267 and 61962014), Guangxi Science and Technology Project (AD18216004 and AD18281079), Guangxi Bagui Scholars Special Project (2019GXNSFFA245014, AA17202024, Ji Li, 2018), Guangxi Key Laboratory of Image and Graphic Intelligent Processing (GIIP202001), and Innovation Project of GUET Graduate Education (No. 2020YCXS053).

References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[2] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020.

[3] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced gan for remote sensing image superresolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5799–5812, 2019.

[4] Z. Wang, P. Yi, K. Jiang et al., "Multi-memory convolutional neural network for video super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2530–2544, 2019.

[5] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multitemporal ultra dense memory network for video super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2503–2516, 2020.

[6] R. Keys, "Cubic convolution interpolation for digital image processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 1, pp. 1153–1160, 1982.

[7] A. Marquina and S. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *Journal of Scientific Computing*, vol. 37, no. 3, pp. 367–382, 2008.

[8] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[9] C. L. P. Chen, L. Liu, L. Chen, Y. Y. Tang, and Y. Zhou, "Weighted couple sparse representation with classified regularization for impulse noise removal," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4014–4026, 2015.

[10] R. Lan, L. Sun, Z. Liu et al., "Cascading and enhanced residual networks for accurate single-image super-resolution," *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 115–125, 2021.

[11] 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.

[12] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image superresolution," in *European Conference on Computer Vision (ECCV)*, pp. 184–199, Springer, 2014.

[13] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1132–1140, 2017.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[15] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[16] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, 2016.

[17] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[18] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng, "Ode-inspired network design for single image super-resolution," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1732–1741, 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[20] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: a persistent memory network for image restoration," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4549–4557, 2017.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 2016no. 6.

[22] X. Wang, K. Yu, S. Wu et al., "ESRGAN: enhanced superresolution generative adversarial networks," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taix'e and S. Roth, Eds., pp. 63–79, Springer International Publishing, 2018.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[24] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, 2018.

[25] K. Jiang, Z. Wang, P. Yi, and J. Jiang, "Hierarchical dense recursive network for image super-resolution," *Pattern Recognition*, vol. 107, p. 107475, 2020.

[26] Y. Li, E. Agustsson, S. Gu, R. Timofte, and L. Gool, *CARN: Convolutional Anchored Regression Network for Fast and Accurate Single Image Super-Resolution*, ECCV Workshops, 2018.

[27] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1664–1673, 2018.

[28] 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.

[29] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, pp. 4278–4284, AAAI Press, 2017.

[31] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[32] J. Dai, H. Qi, Y. Xiong et al., "Deformable convolutional networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017.

[33] L. Liu, B. Liu, H. Huang, and A. C. Bovik, "Noreference image quality assessment based on spatial and spectral entropies," *Signal processing: Image communication*, vol. 29, no. 8, pp. 856–863.

[34] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[35] W. Shi, J. Caballero, F. Husz'ar et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, 2016.

[36] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision (ECCV)*, pp. 391–407, Springer, 2016.

[37] H. A. Aly and E. Dubois, "Image up-sampling using total-variation regularization with a new observation model," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647–1659, 2005.

[38] R. Lan, L. Sun, Z. Liu et al., "Cascading and enhanced residual networks for accurate single-image super-resolution," *IEEE Transactions on Cybernetics*, pp. 1–11, 2020.

[39] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *International Conference on Learning Representations*, vol. 12, 2014.

[40] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. A. Morel, "Low-complexity single-image superresolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference*, pp. 135.1–135.10, BMVA Press, 2012.

[41] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," *Proceedings of the 7th International Conference on Curves and Surfaces*, , pp. 711–730, Springer-Verlag, Berlin, Heidelberg, 2010.

[42] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[43] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, 2015.

[44] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 624–632, 2017.

[45] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 723–731, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6645803, 10 pages
https://doi.org/10.1155/2021/6645803

Research Article

Intelligent Channel Allocation for Age of Information Optimization in Internet of Medical Things

Kefeng Wei,[1,2] Lincong Zhang,[3] and Shupeng Wang[4]

[1]*School of Computer Science and Engineering, Northeastern University, Shenyang, China*
[2]*Shen Kan Engineering and Technology Corporation, MCC., Shenyang, China*
[3]*School of Information Science and Engineering, Shenyang Ligong University, Shenyang, China*
[4]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Lincong Zhang; lincongz@foxmail.com and Shupeng Wang; wangshupeng@iie.ac.cn

Received 28 December 2020; Accepted 16 August 2021; Published 31 August 2021

Academic Editor: Xiaojie Wang

Along with the development of realtime applications, the freshness of information becomes significant because the overdue information is worthless and useless and even harmful to the right judgement of system. Therefore, The Age of Information (AoI) used for marking the freshness of information is proposed. In Internet of Medical Things (IoMT), which is derived from the requirement of Internet of Thins (IoT) in medicine, high freshness of medical information should be guaranteed. In this paper, we introduce the AoI of medical information when allocating channels for users in IoMT. Due to the advantages of Deep Q-learning Network (DQN) applied in resource management in wireless network, we propose a novel DQN-based Channel Allocation (DQCA) algorithm to provide the strategy for channel allocation under the optimization of the system cost considering the AoI and energy consumption of coordinator nodes. Unlike the traditional centralized channel allocation methods, the DQCA algorithm is distributed as each user performs the DQN process separately. The simulation results show that our proposed DQCA algorithm is superior to Greedy algorithm and Q-learning algorithm in terms of the average AoI, average energy consumption and system cost.

1. Introduction

Corona Virus Disease 2019 (COVID-19) has caused more than 2.32 million deaths worldwide by February 8th, 2021 [1]. People are forced to stay at home, reduce the trip proportion, and avoid to go to crowded places. In this case, both the government, medical staff, or the general public hope to monitor virus infections like COVID-19 and isolate them in time to avoid the spread of the virus on a large scale. Besides, people are more concerned about their health than ever before. More and more chronic patients and even healthy people hope to have long-term effective monitoring of their bodies and obtain important information about their health as soon as possible. The emergence of the Internet of Medical Things (IoMT) has provided the possibility to solve these problems, and its intelligent monitoring function has gained massive demand around the world [2].

For the COVID-19 virus, Swati Swayamsiddha et al. proposed a Cognitive Internet of Medical Things (CIoMT), which is a particular case of the IoMT, enabling real-time tracking, remote monitoring of patients, rapid diagnosis, contact tracing and clustering, screening and monitoring, etc., thus reducing the workload of medical staff and preventing and controlling the spread of the virus [3]. RaviPratap Singh et al. discussed the feasibility of using the IoMT to track, monitor, analyze data, and provide treatment plans for orthopedic patients in an environment ravaged by COVID-19 [4]. For COVID-19 management, M.A. Mujawar et al. also proposed a health monitoring system based on wearable devices and artificial intelligence, which continuously monitors the patient's heartbeat, body temperature, and other parameters through medical sensors and transmits them to cloud storage through WSN. At the same time, these parameters are used to update the user's health status in real time and then the status will be sent to the medical staff [5].

The IoMT is a vast network system with diverse technologies. This paper only studies the channel allocation problems in the monitoring and transmitting human physiological data

in the IoMT. During the monitoring and transmission, too old data may cause erroneous analysis and evaluation, reduce the accuracy and reliability of system decision-making, and even threaten the safety of users. Therefore, the freshness of information is crucial, and it also occupies an essential position in the design of 6G systems applied to body area networks [6–10]. To effectively describe the freshness of information, this paper introduces the Age of Information (AoI) [11], and studies the channel allocation problem of IoMT with AoI as the target.

In recent years, artificial intelligence has become an effective method to solve the resource allocation problem with many data processing [12]. As the main solution of artificial intelligence, machine learning has also received tremendous attention in recent years. Machine learning uses algorithms to analyze and learn from data to make decisions and predictions about real-world events. Among them, deep learning is the most popular machine learning method at present, which has been well applied in automatic detection [13, 14], case recognition [15–17], environmental monitoring [18], and epidemic prediction [19], etc. In terms of channel allocation, with the rapid growth of network size and data volume, deep learning can significantly improve the processing speed for a large number of nodes [20–23].

The research content of this paper is the problem of channel allocation among users oriented to the optimization of the AoI. The AoI of each controller on each user's body at the gateway is the number of slots experienced by the latest update received from this controller at the end of each slot. In each time slot, the system needs to pay for the AoI. our requirement of timely updating the content received by the gateway is reflected in the minimum payment cost of the whole system. At the same time, this paper adopts a deep learning method to solve the proposed optimization problem. The main contributions of this paper are as follows:

(i) In view of the channel allocation problem of the IoMT, we focus on the timeliness of the information, and at the same time, considering the mobility of nodes. To measure the cost that the system pays for the lack of new information on gateway, we propose a system cost function based on the AoI and the current energy consumption rate of the nodes.

(ii) Based on the cost function, we constructed a mathematical model of the optimization problem that minimizes the average cost for the channel allocation of the IoMT.

(iii) For the problems raised, we propose a Deep Q-Learning Network (DQN) based channel allocation algorithm, named DQCA, which provides channel allocation scheme to minimize the cost on the basis of meeting the requirements of node SNR and residual energy.

The rest of the paper is organized as follows. Section 2 provides a comprehensive overview about the AoI. Section 3 describes the system model and optimization model of channel allocation problem in IoMT. The proposed DQCA algorithm is illustrated in Section 4. The simulation and performance evaluation is performed in Section 5. Finally, we conclude the paper in Section 6.

2. Related Works

With the increasingly developed Internet of Things (IoT), real-time applications are gradually increasing, such as driverless cars, which make decisions and control based on road information detected by sensors, adjust the travel mode of vehicles, avoid collisions, and ensure the safe driving of driverless cars. This type of application requires high timeliness and freshness of data, and outdated data will lead to wrong judgments and decisions. The longer the time, the less important and effective the data will be. In order to measure the freshness and effectiveness of data, scholars put forward the indicator of the AoI in 2011 to quantify the freshness of information on a remote system state [11]. The AoI refers to the time elapsed between the creation of the newly successfully received information and its successful reception. The AoI is different from the transmission delay of information. In a system with multiple source nodes and one destination node, each source node collects information and sends it to the destination node regularly. At the destination node, the AoI of each source node can be calculated [24]. Since the source node is constantly sending information to the destination node, the AoI of each source node refers to the AoI of the latest information received by the destination node from that source node. In other words, the AoI of each source node is not fixed and depends on the sending rate of the source node and the receiving rate of the destination nodes for source node's information. If the destination node has not received the latest information from a certain source node, then the AoI of the source node will show a linear increase until it gets the newest information from the source node and changes to the AoI of the latest information.

As shown in Figure 1, $t_i(i = 0, 1, 2 \cdots)$ is the time that data packet i is generated by node j, t_i' is the time that data packet i is received by the destination. When $t = 0$, the destination node receives a data packet 0 from node j, then $A_j(0) = A_0 = t_0$. Then $A_j(t)$ increases linearly until the destination node receives a latest data packet 1 at t_1'. At this time, $A_j(t)$ is updated as $A_j(t_1') = A_1 = t_1' - t_1$. Like this, we can deduce that $A_j(t_2') = A_2 = t_2' - t_2$ when the destination node receives a latest data packet 2 at t_2', and so forth.

The Swedish scholar Antzela Kosta et al. published a review paper on the AoI in 2017, introducing the concept of AoI in detail and summarizing the early researches [25]. Jhunjhunwala P R et al. proposes an AoI-aware channel scheduling algorithm for a sensor network with a monitoring station and multiple source nodes. The algorithm proposes that the cost function is a non-declining function, but it does not provide a completely function and optimization model [24].

There have been some researches on the AoI in the IoT. Abbas Q et al. studied the importance and optimization of

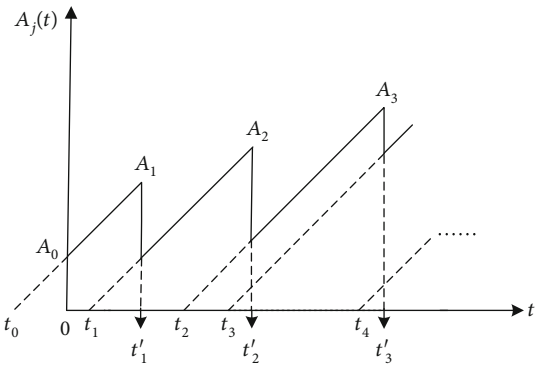


Figure 1: Age of Information.

the AoI and energy efficiency in the IoT [26]. Gu Y et al. studied the average peak AoI under two schemes of overlay and underlay in a cognitive radio-based IoT network [27]. Li J et al. studied the average peak AoI of time-limited multicast transmission in the IoT. The author first describes the evolution of the instantaneous AoI and then derives the service time distribution of all possible reception results on IoT devices, and obtains the closed expressions of the average AoI and the average peak AoI [28]. Azarhava H et al. proposed a new protocol based on non-orthogonal multiple-access (NOMA) in a wireless IoT network with energy harvesting sensors and limited battery cells. A closed-form equation of the AoI for the entire network is obtained and the AoI is optimized by power scheduling parameters [29].

3. System Model and Optimization Model

3.1. System Model. Figure 2 illustrates the topology of the IoMT which is born out of the IoT and wearable devices. Therefore, the core of the IoMT are the users equipped with several wearable devices involving wireless sensors. These wearable devices on user's body can detect the physiological information (such as the blood pressure, the pulse, the temperature, and the electrocardiogram (ECG), etc.) and mobility information (such as location, move speed and move direction, etc.). In addition, there is a coordinator on user's body used to collect the information from all the wearable devices on the same body and communicate with the gateway. The physiological information of all users is sent to the gateway and then transmitted to the nurse, doctor or ambulance on demand through the Internet. In this paper, each user selects a channel from a gateway in each time slot. In order to describe the problem more conveniently, we first illustrate the notations.

The AoI of each mobile node is defined as the elapsed time when the latest data of this node is received by the gateway, as shown in Eq. (1).

$$l_j^i(q+1) = \begin{cases} t_q - t_{gen_{cur}}\,; \textit{user j sends data} \text{ to } \textit{gateway i} \text{ in } \textit{time slot q}\,; \\ l_j^i(q+1) + t_s\,; \textit{user j fails} \text{ to } \textit{send data} \text{ in } \textit{time slot q}\,; \end{cases} \tag{1}$$

t_gen_cur is the generation time of the currently received data frames, t_s is the length of each time slot. Here, we represent the AoI by using the specific time other than the time slot, which is more precise. At each time slot, the system pays the cost for AoI, and the cost $C(t)$ is defined as a function of the AoI of all mobile nodes. Since $C(t)$ is the cost paid by the system for lack of fresh information from the source node, it is a non-descending function, as shown in Eq. (2).

$$C(t) = f(\boldsymbol{L}(t)) = f(l_1(t)) + f(l_2(t)) + \cdots f\left(l_j(t)\right) + \cdots f(l_M(t)) \tag{2}$$

Where $f(l_j(t))$ is defined as the cost function of the AoI of node *j*,

$$f\left(l_j(t)\right) = w_j l_j(t) \tag{3}$$

$w_j = E_j/E_0$ is the weight coefficient, it is determined by the ratio of the consumed energy of node *j* to the initial energy, Among them, E_j is the energy consumed by the node, $\varepsilon_{fs} d_{i,j}^2$ is the energy consumption of free space transmission.

$$E_j = b\left(e_t + \varepsilon_{fs} d_{i,j}^2\right), \text{if } d_{i,j} \le d_{th} \tag{4}$$

The mobile node communication complies with the 802.11 standards and adopts OFDM technology. The signal-to-noise ratio of the mobile node is defined as follows:

$$\gamma_{j,k}^i(t) = \frac{p_{i,j}(t) h_{i,j}^2 \beta_{j,k}^i(t)}{\sigma^2} \tag{5}$$

3.2. Optimization Model.

$$\text{objective} \min_{\alpha_{j,k}^i(t)} \sum_{j\in\mathfrak{M}} \sum_{i\in\mathfrak{N}} \lim_{t\longrightarrow\infty} \frac{1}{T} \sum_{t=1}^{T} C(t) \tag{6}$$

s.t.

$$\sum_{j=1}^{M} \sum_{i=1}^{N} \alpha_{j,k}^i(t) = 1, \forall k \in \mathfrak{C} \tag{7}$$

$$\sum_{i=1}^{M} \alpha_{j,k}^i(t) = 1, \forall k \in \mathfrak{C}, j \in \mathfrak{N} \tag{8}$$

$$\alpha_{j,k}^i(t) \in \{0, 1\}, \forall k \in \mathfrak{C}, i \in \mathfrak{N}, j \in \mathfrak{M} \tag{9}$$

$$\sum_{j=1}^{M} \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_{j,k}^i(t) \le N \tag{10}$$

$$\sum_{j=1}^{M} \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_{j,k}^i(t) \le K \tag{11}$$

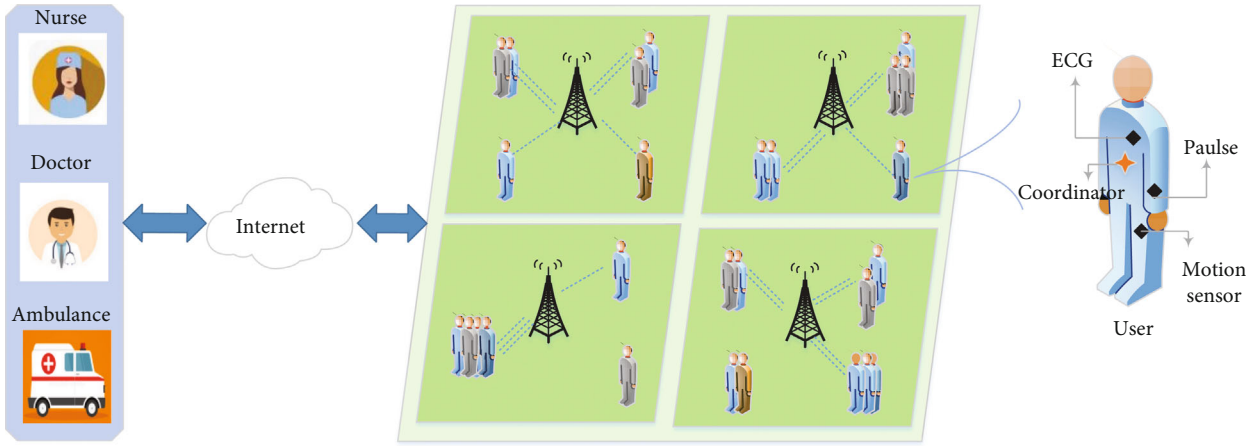


Figure 2: The topology of the IoMT.

$$\sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{k=1}^{K}\alpha_{j,k}^{i}(t)b_k \le B \tag{12}$$

$$\gamma_{j,k}^{i}(t) \ge \gamma_0 \tag{13}$$

The formula (7) indicates that in any time slot t, one channel k can only be allocated to one node j. The formula (8) indicates that in the time slot t, a node can only communicate with one gateway. The formula (9) indicates whether the channel k of the gateway i is allocated to the user j in the time slot t, 1 means yes, and 0 means no. Equation Formula (10) indicates that the number of occupied gateways cannot exceed the number of available gateways. Equation Formula (11) indicates that the number of occupied channels cannot exceed the number of available channels. Equation Formula (12) indicates that the occupied channel bandwidth cannot exceed the total channel bandwidth. Equation Equation (13) indicates that the signal-to-noise ratio of a node must be higher than the threshold so as to ensure the transmission rate.

For the network with small scale and small total number of channels, the enumeration method is available to calculate the cost of users choosing a subchannel of a gateway, and then find the subchannel with the lowest cost. However, if there are 1000 users, 5 gateways and 64 subchannel in the network, the amount of calculation of payment for AoI by enumeration method is at least 320000 times. Thus, for larger networks, the computational complexity is quite high. It is considerably significant to design a low-complexity algorithm to solve the proposed problem.

4. DQCA Algorithm Design

We assume that each user selecting the channel is a Markov decision process (MDP) and the policy decision and the AoI just depend on the selection in last time slot. In this network, there are a large number of users and they move randomly. The optimization model mentioned above is difficult to obtain an optimal analytical solution because the result of optimization depends largely on the built model and the computing process rate of the computer. Reinforcement learning is suitable for the channel allocation problem of the network. On the one hand, it can adjust actions through the interaction between the user and the environment and rewards, which can solve the optimization problem that is difficult to obtain analytical solutions; on the other hand, it can be well adapted to a highly dynamic environment and the frequently changing channel. Q-learning and DQN are two typical reinforcement learning algorithms. The algorithm flow diagrams are shown in Figures 3 and 4, respectively.

In Q-learning, the agent chooses an action under each state, builds a Q-table and record the Q-value for each pair of state and action. The Q-value is updated by the reward produced by the selected action. However, since all the possible states and actions are enumerated in Q-table, Q-learning is only suitable for the MDP problem with small state space and small action space. When the space becomes large, the storage space of the Q-table will become very large, and the Q-table cannot hold the memory. Meanwhile, the convergence speed of Q-learning will come down.

Compared with Q-learning algorithm, DQN uses the artificial neural network (ANN) to approximate the value function, uses target Q network to update the target value and use experience replay to train the learning process of reinforcement learning. DQN just updates the parameter θ of the artificial neural network rather than update the whole Q-table. Therefore, it shortens the convergence time and is more suitable for the problem with large state and action space. Considering a large number of users and channels, we abandon the Q-learning algorithm based on Q-table and choose the DQN to train the network to obtain an approximate optimal solution. Our proposed DQCA algorithm is a channel allocation algorithm based on DQN.

Agent: We define the controller node on mobile user as an agent. As an agent, it trains the neural network according to the network status (number of users, user location, moving speed and direction of users, etc.) to obtain reasonable actions.

System state: Denoted by $s(t)$, including channel environment and node behavior. The behavior of the node mainly refers to the current position of the node (the mobility of node follows the random walk model [24]), and the nearest

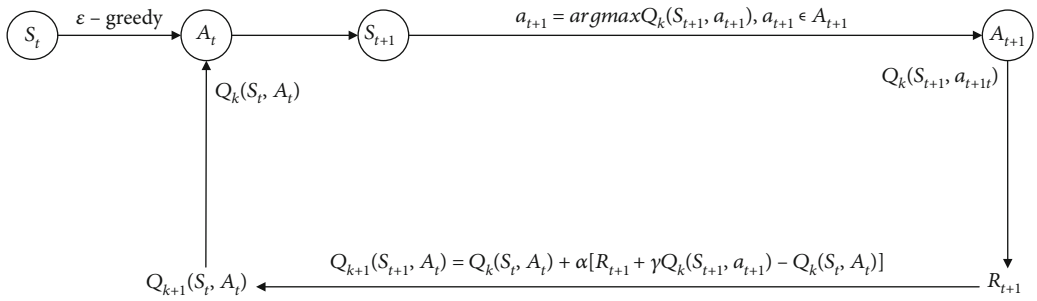


Figure 3: The flow chart for Q-learning.

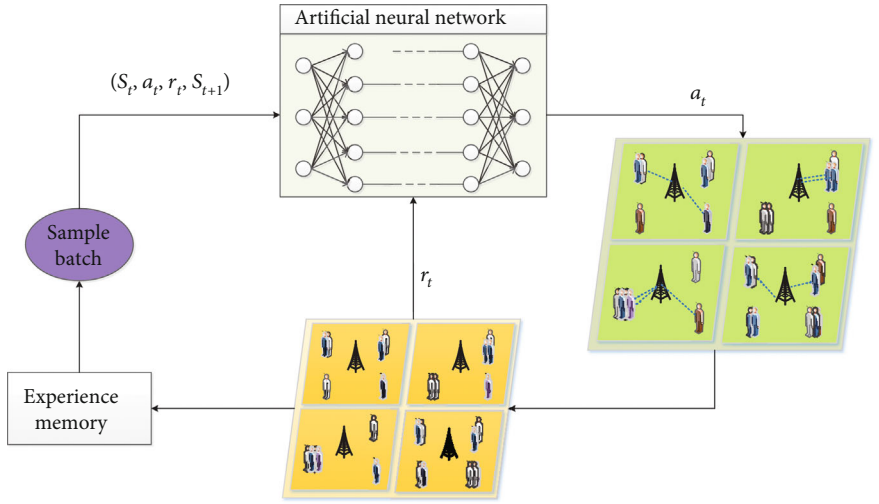


Figure 4: The flow chart for DQN.

gateway is selected for access according to the position of the node. The channel environment can be characterized by the signal-to-noise ratio of the node. If node j selects gateway i for data transmission in time slot t, the signal-to-noise ratio of node i in time slot t is $\gamma_{j,k}^{i}(t)$, if $\gamma_{j,k}^{i}(t) \geq \gamma_0$, then $s_j(t) = 1$; otherwise $s_j(t) = 0$. That is, $s_j(t) = \{0, 1\}$.

$$s(t) = \{s_1(t), s_2(t) \cdots s_j(t) \cdots s_M(t)\} \tag{14}$$

System action: After the node selects the gateway i, the system action is defined as which channel k of the gateway i is selected by the node j.

$$a_j(t) = \left\{a_{j,1}^{i}(t), \cdots..a_{j,k}^{i}(t) \cdots .a_{j,K}^{i}(t)\right\} \tag{15}$$

Reward: User j uses the immediate reward produced by $a_j(t)$ at the system state $s_j(t)$, which is defined as Eq. (16). This revenue function can ensure that the cost of AoI is minimized while meeting the channel ratio constraint.

$$r_j(t) = \begin{cases} -f\left(l_j(t)\right), \gamma_{j,k}^{i}(t) \geq \gamma_0 \\ -\infty, otherwise \end{cases} \tag{16}$$

For each user j, we define the Q function as $Q(s_t, a_t)$ when take action a_t at state s_t, as shown by Eq. (17).

$$Q(s_t, a_t) = r_j(s_t, a_t) + \delta P(s_{t+1} \mid s_t, a_t) \max_{a_t \in \mathfrak{A}} Q(s_{t+1}, a_{t+1}) \tag{17}$$

Where $P(s_{t+1} \mid s_t, a_t)$ is the transition function from state s_t to state s_{t+1}. δ is a discount factor used to balance the immediate reward and long-term reward. $\mathfrak{A}$ is the set of feasible actions.

Q function and optimal policy: Then the optimal value of Q function and the optimal policy π^* can be represented as Eq. (18) and Eq. (19), respectively.

$$Q^*(s_t, a_t) = \max_{a_{t+1} \in \mathfrak{A}} Q(s_t, a_{t+1}) \tag{18}$$

$$\pi^* = \operatorname*{argmax}_{a_{t+1} \in \mathfrak{A}} Q(s_t, a_{t+1}) \tag{19}$$

Target value: To avoid overestimation brought by only one parameter θ in neural network, we use parameter θ and θ' to illustrate the predict network and target network, respectively. Then the Q-function can be given by Eq. (20).

Input: Node list, gateway list
Initialization:
1. Initialize cost c and energy e to 0.
2. Initialize step to 1.
For episode =1 to maximum iteration time T do
Count =1;
Obtain state s_t based on the input.
1. **Repeat:**
(1) Select action $a_t = \arg\max_a Q(s_t, a \mid \theta)$
(2) Output the next state s_{t+1}, reward r_t, cost c and energy e according to the count and action a_t
(3) Store transition (s_t, a_t, r_t, s_{t+1}) in the replay memory
2. **If** step >200 and step % 5 == 0:
Sample random minibatch of transitions (s_t, a_t, r_t, s_{t+1}) from the replay memory pool.
Else:
Continue
3. **Update**:
$s_t \longleftarrow s_{t+1}$;
c += c;
e += e;
Step + =1;
Count + =1;
For each node in node list
If packet size <= 0:
Break
4. Update target value $y = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1} \mid \theta')$.
5. Perform a gradient descent step on $(y - Q(s_t, a_t \mid \theta))^2$.
6. Reset $\theta' = \theta$ for every Z steps.
End for

Algorithm 1: Channel allocation algorithm for user j based on DQN

$$y = r_t + \gamma \max_{a_{t+1}} Q\left(s_{t+1}, a_{t+1} \mid \theta'\right). \tag{20}$$

Loss function: To approximate the Q-function, we also define the loss function as Eq. (20) to train the weights θ and θ' of ANN.

$$\mathfrak{L}(\theta) = (y - Q(s_t, a_t \mid \theta))^2 \tag{21}$$

In DQCA, we first get the locations of all nodes and gateways and select a gateway for each node according to the shortest distance. And then we perform the channel allocation algorithm by Algorithm 1.

5. Simulation and Performance Evaluation

In this section, we first introduce the simulation setup, then show the simulation results and analyze the performance of the proposed algorithm.

5.1. Simulation Setup. To testify the effectiveness of our proposed algorithm, the Q-learning algorithm and greedy algorithm are also simulated with the DQCA algorithm for comparison. Q-learning algorithm builds Q-table for each node and finds the maximum Q-value for each node from all available actions. The main idea of the greedy algorithm is to allocate the channel in each time slot with the minimum growth of the cost function in the next slot as the optimization objective [24]. To prove the effectiveness of the proposed algorithm, this paper compares the three

Table 1: Simulation parameters.

Simulation parameter	
Simulation area	$100 \times 100m^2$
Gateway number	1
Subchannel number	10
Gateway bandwidth	20 MHz
Node power	0.1 W
Time slot	0.1 s
AWGN delta	-110 dBm/Hz
SNR channel gain weight	-2
Node intial energy	2 J
Transmitter power	8×10^{-8} J/bit
Freedom space weight	6×10^{-8} J/bit
Distance threshold	0.4 m
SNR threshold	0.5
DQN iterations	2000
DQN steps	20000
Penalty reward	-200
Stay reward	-150

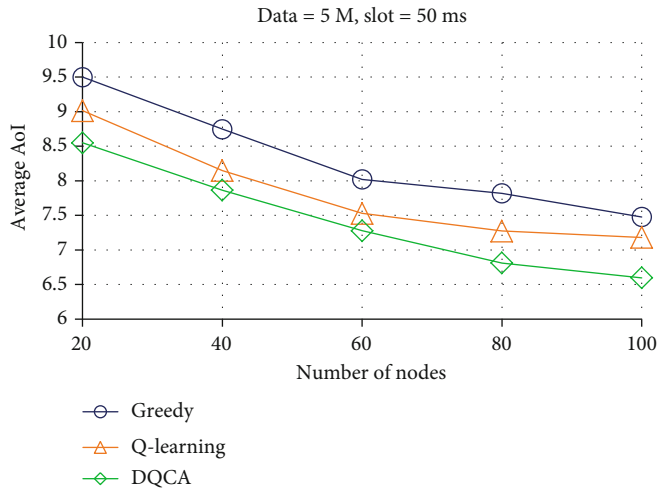


Figure 5: Average AoI with different number of nodes (Data = 5 M, slot = 50 ms).

algorithms from three aspects: cost, AoI, and energy consumption. Among them, cost refers to the overall cost of the network, calculated according to formula (6), and the average AoI of all nodes is calculated as follows:

$$\text{AoI} = \frac{\sum_{j=1}^{M} l_j(t)}{M} \tag{22}$$

Energy consumption is the average energy consumption of all nodes, defined as

$$\frac{\sum_{j=1}^{M} w_j(t)}{M} \tag{23}$$

5.2. Performance Evaluation. To verify the effectiveness and feasibility of the proposed DQCA algorithm, this paper uses three different scenarios. The first one: the average size of data packet is 5 M, and the data packet arrival interval is 50 ms, the number of nodes in the network changes; the second one: the number of nodes is 20, the data packet arrival interval is 50 ms, and the average size of data packet changes; the third one: the number of nodes is 20, the average size of data packet is 5 M, and the data packet arrival interval changes. The simulation program runs on a computer with an Intel Core i7-3520M with 2.90GHz frequency CPU and 8G RAM. The parameters used in the simulation are shown in Table 1.

Figures 5–7 study the impact of changes in the number of nodes on network performance when the length of the data packet and time slot is fixed as defined in the first scenario. It can be seen from Figures 5 and 6 that the average AoI and average energy consumption of the three algorithms continuously reduce as the number of nodes increases. This is because the AoI and energy consumption increase more slowly than the number of nodes, resulting in a decrease in the average value. At the same time, due to the large state space, Q-learning needs to consume more time and computing resources, and it is necessarily inferior to DQCA in terms of AoI and energy consumption. Especially for energy con-

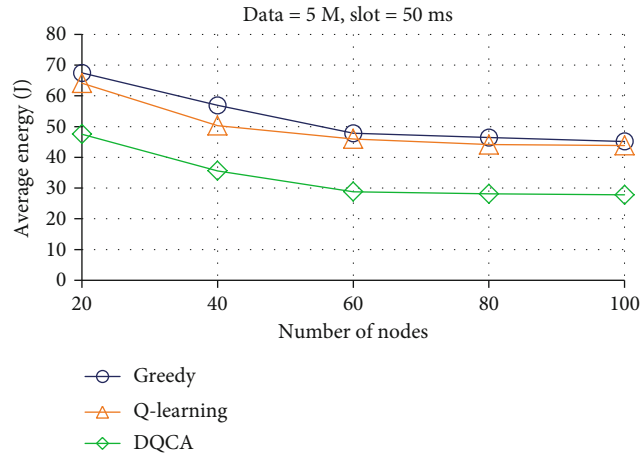


Figure 6: Energy with different number of nodes (Data = 5 M, slot = 50 ms).

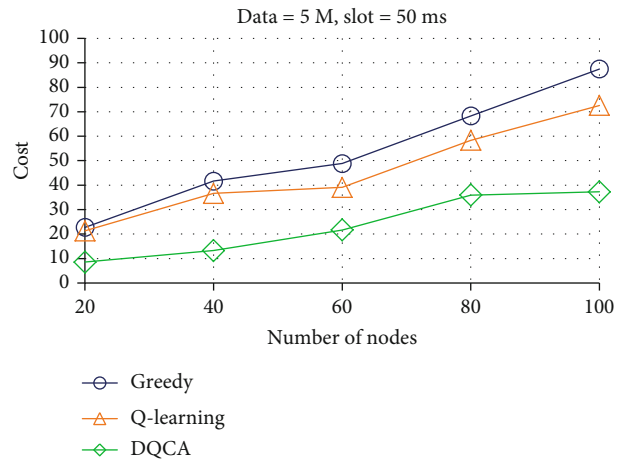


Figure 7: Cost with different number of nodes (Data = 5 M, slot = 50 ms).

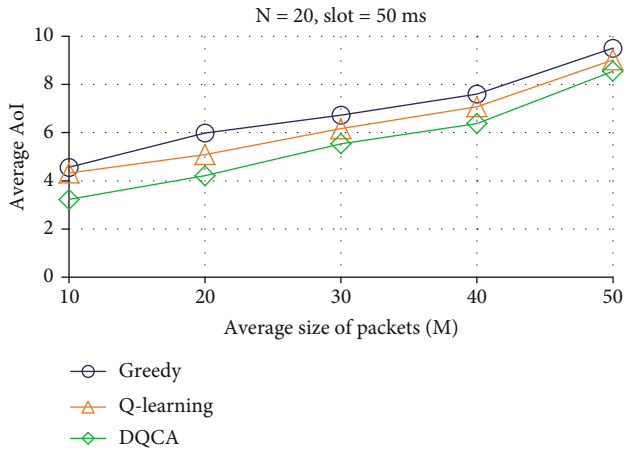


Figure 8: Cost with different average size of packets (N = 20, slot = 50 ms).

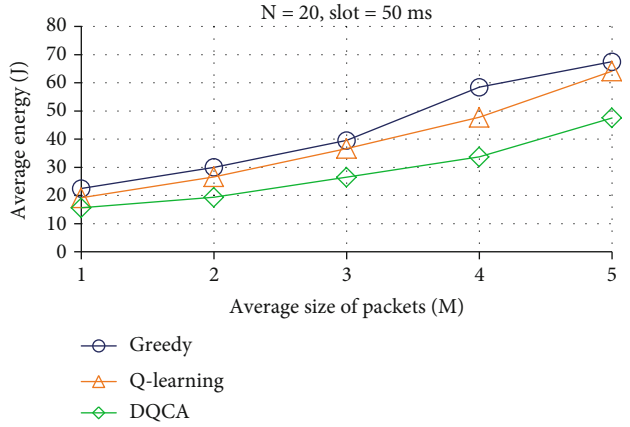


Figure 9: Cost with different average size of packets (N = 20, slot = 50 ms).

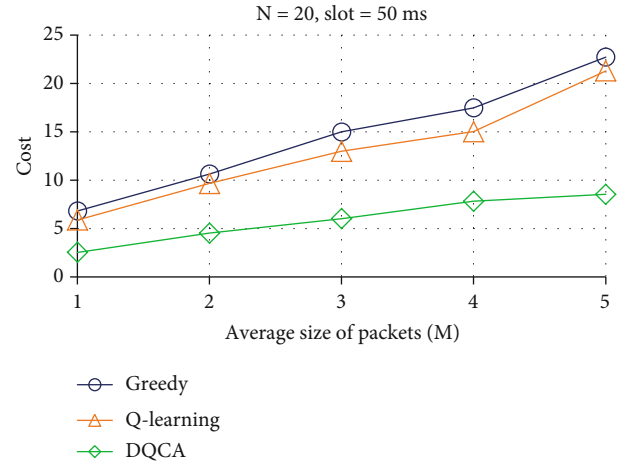


Figure 10: Cost with different average size of packets (N = 20, slot = 50 ms).

sumption, compared with the Q-learning algorithm, the average energy consumption of all nodes of the DQCA algorithm is reduced by about 38.56%. It can be seen from

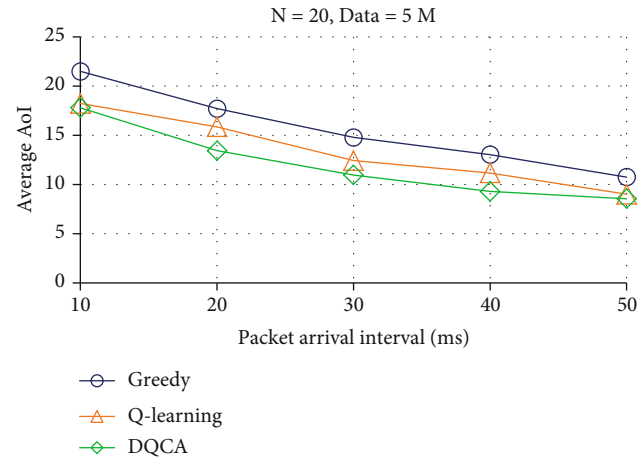


Figure 11: Cost with different packet arrival intervals (N = 20, Data = 5 M).

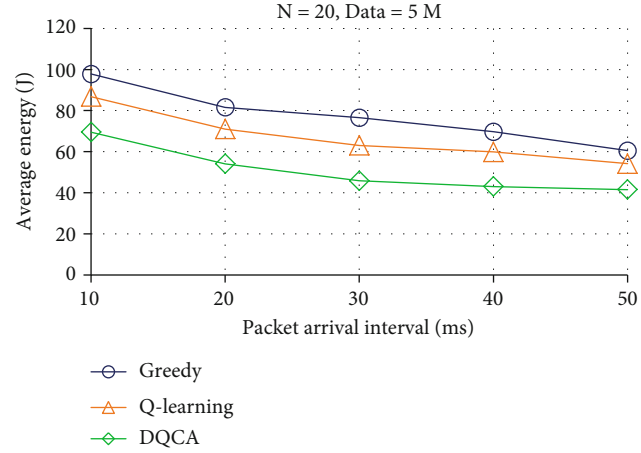


Figure 12: Cost with different packet arrival intervals (N = 20, Data = 5 M).

Figure 7 that the costs of the three algorithms all increase with the increase of the number of nodes, among which the DQCA algorithm increases slowly and the increment is small. The cost takes into account the AoI and energy consumption of the nodes. The DQCA algorithm has more advantages in these two aspects than the other two algorithms. Therefore, the total cost is significantly lower than the Greedy and Q-learning algorithms and can be reduced by up to 57.3% compared to the Greedy algorithm.

Figures 8–10 shows the fixed number of nodes and packet arrival interval in the second scenario, to study the change of network performance with the size of data packet. It can be seen that as the size of the data packet continues to increase, the AoI and energy consumption of the node is also increasing, so the cost increases accordingly. This is because after the data packet increases, the processing and transmission time of the data packet increases, and it takes longer time for the gateway node to wait for the latest update of the node, and the energy consumption of the transmitter and receiver of the nodes will increase accordingly.

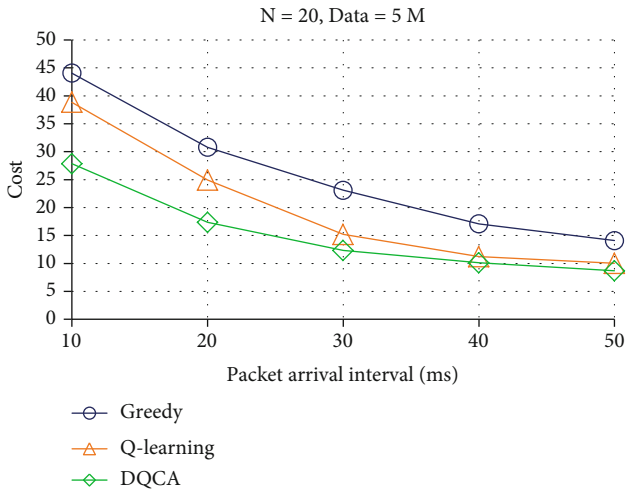


FIGURE 13: Cost with different packet arrival intervals (N = 20, Data = 5 M).

Compared with the greedy algorithm and Q-learning algorithm, the DQCA algorithm can reduce the cost by about 62% and 60%.

Figures 11–13 show the fixed number of nodes and the size of data packet in the third scenario, to study the change of network performance with the packet arrival interval. It can be seen that when the data packet arrival interval increases, the number of data packets in the network decreases, and the data packets sent and received by the node decrease, so the energy consumption of the node is reduced. Due to the increase of the data packet arrival interval, the probability of the node being allocated to the channel at the gateway node also increases, that is, the waiting time for an assigned channel for the node is shortened. As can be seen from Figure 11, overall, the AoI of the node is reduced. From the simulation results in Figure 13, as the packet arrival interval continues to increase, the impact on the average energy consumption and cost of the node gradually decreases, and the curves in Figures 12 and 13 tend to be stable. This is because the packet arrival interval increases to a certain extent, the basic energy consumption of the node accounts for a larger proportion of the total energy consumption, and the energy consumption of the node is less affected by the sending and receiving of data packets.

The greedy algorithm only considers the optimal value of the current function, does not consider the previous choice, nor the consequences of the current choice. But in fact, this method often does not have the best results. Therefore, in Figures 5–13, the greedy algorithm has the worst performance compared to Q-learning and DQCA.

6. Conclusion

Focused on the freshness of information in IoMT, this paper studied the channel allocation problem oriented to the AoI. In this paper, system cost is defined as a non-descending function about the AoI and energy consumption of nodes. Since the system cost optimization problem is difficult to solve due to the large amount of users and the mobility of users, we adopt a DQN-based method named DQCA algorithm. The simulation compared the proposed DQCA algorithm with Greedy algorithm and Q-learning algorithm in three different cases. The simulation results show the superiority of DQCA algorithm from the aspects of average AoI and average energy consumption of nodes and system cost.

Notations

N: the total number of gateway nodes
M: the total number of nodes
K: the total number of sub-channels
i: the gateway index, $1 \le i \le N$
j: the node index, $1 \le j \le M$
k: the channel index, $1 \le k \le K$
$\mathfrak{C}, \mathfrak{N}, \mathfrak{M}$: are the set of channels, the set of gateways, and the set of nodes, respectively
q: the time slot sequence number
t_q: the time when the data frame is received in the q_{th} time slot
$l_j^i(q)$: the information age of the node j communicating with the gateway i in the time slot q
b: the length of the frame sent
e_t: the basic energy consumption of the transmitter
ε_{fs}: energy consumption parameter for free space transmission
$d_{i,j}$: the distance between mobile node j and gateway i
$\gamma_{j,k}^i(t)$: the signal-to-noise ratio of the mobile node
$p_{i,j}(t)$: is the transmission power of node j to gateway i
$h_{i,j}^2$: the channel gain
σ: Gaussian noise
Z: a number can be set on demand.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The author(s) declare(s) that they have no conflicts of interest.

Acknowledgments

National Natural Science Foundation of China, Grant/Award Number: 61501308; Basic research project of Liaoning Provincial Department of Education, Grant/Award Number: LG202027; Postdoctoral Research Station project of Shenyang Ligong University.

References

[1] https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_aladin_banner#tab4.

[2] Z. Ning, P. Dong, X. Wang et al., "Mobile Edge Computing Enabled 5G Health Monitoring for Internet of Medical Things: A Decentralized Game Theoretic Approach," *IEEE Journal on*

Selected Areas in Communications, vol. 39, no. 2, pp. 463–478, 2021.

[3] S. Swayamsiddha and C. Mohanty, "Application of cognitive Internet of Medical Things for COVID-19 pandemic," *Diabetes and Metabolic Syndrome Clinical Research and Reviews*, vol. 14, no. 5, pp. 911–915, 2020.

[4] R. Pratap Singh, M. Javaid, A. Haleem, R. Vaishya, and S. Ali, "Internet of medical things (IoMT) for orthopaedic in COVID-19 pandemic: roles, challenges, and applications," *Journal of Clinical Orthopaedics and Trauma*, vol. 11, no. 4, pp. 713–717, 2020.

[5] M. A. Mujawar, H. Gohel, S. K. Bhardwaj, S. Srinivasan, N. Hickman, and A. Kaushik, "Nano-enabled biosensing systems for intelligent healthcare: towards COVID-19 management," *Materials Today Chemistry*, vol. 17, p. 100306, 2020.

[6] L. Barbierato, A. Estebsari, E. Pons et al., "A distributed IoT infrastructure to test and deploy real-time demand response in smart grids," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1136–1146, 2019.

[7] Z. Ning, P. Dong, X. Wang et al., "Partial Computation Offloading and Adaptive Task Scheduling for 5G-enabled Vehicular Networks," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[8] S. H. Shao, A. Khreishah, and I. Khalil, "Enabling real-time indoor tracking of IoT devices through visible light retroreflection," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 836–851, 2020.

[9] Z. Ning, P. Dong, X. Wang et al., "Distributed and Dynamic Service Placement in Pervasive Edge Computing Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 6, pp. 1277–1292, 2021.

[10] H. Viswanathan and P. Mogensen, "Communications in the 6G era," *IEEE Access*, vol. 99, pp. 1–1, 2020.

[11] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *8th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks (SECON)*, pp. 350–358, Salt Lake City, UT, USA, 2011.

[12] X. Wang, Z. Ning, and S. Guo, "Minimizing the Age-of-Critical-Information: An Imitation Learning-Based Scheduling Approach under Partial Observations," *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[13] F. James and M. Priya, "Deep Learning Radial Basis Function Neural Networks Based Automatic Detection of Diabetic Retinopathy," *SSRN Electronic Journal*, 2020.

[14] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation Learning Enabled Task Scheduling for Online Vehicular Edge Computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[15] S. Abbasi, S. Saberi, M. Zarvani, P. Amiri, and R. Azmi, "Deep Learning Classification Schemes for the Identification of COVID-19 Infected Patients Using Large Chest X-Ray Image Dataset," in *42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'20)*, Montreal, QC, Canada, 2020.

[16] T. Dong, C. Yang, B. Cui et al., "Development and validation of a deep learning Radiomics model predicting lymph node status in operable cervical Cancer," *Frontiers in Oncology*, vol. 10, 2020.

[17] Z. Ning, K. Zhang, X. Wang et al., "Intelligent Edge Computing in Internet of Vehicles: A Joint Computation Offloading and Caching Solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2212–2225, 2021.

[18] W. Qiao, W. Tian, Y. Tian, Q. Yang, Y. Wang, and J. Zhang, "The forecasting of PM2.5 using a hybrid model based on wavelet transform and an improved deep learning algorithm," *IEEE Access*, vol. 7, pp. 142814–142825, 2019.

[19] J. Sadefo Kamdem, R. Bandolo Essomba, and J. Njong Berinyuy, "Deep learning models for forecasting and analyzing the implications of COVID-19 spread on some commodities markets volatilities," *Chaos, Solitons & Fractals*, vol. 140, p. 110215, 2020.

[20] B. Zhao, J. Liu, Z. Wei, and I. You, "A deep reinforcement learning based approach for Energy-Efficient Channel allocation in satellite internet of things," *IEEE Access*, vol. 8, pp. 62197–62206, 2020.

[21] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2021.

[22] Z. Gao, M. Eisen, and A. Ribeiro, "Resource Allocation Via Model-Free Deep Learning in Free Space Optical Networks," 2020, https://arxiv.org/abs/2007.13709v1.

[23] Z. Ning, S. Sun, X. Wang et al., "Intelligent Resource Allocation in Mobile Blockchain for Privacy and Security Transactions: A Deep Reinforcement Learning Based Approach," *Science China Information Sciences*, vol. 64, no. 6, 2021.

[24] P. R. Jhunjhunwala, "Age-of-Information Aware Scheduling," in *2018 International Conference on Signal Processing and Communications (SPCOM)*, Bangalore, India, 2018.

[25] A. Kosta, N. Pappas, and V. Angelakis, "Age of information: a new concept, metric, and tool," *Foundations and Trends in Networking*, vol. 12, no. 3, pp. 162–259, 2017.

[26] Q. Abbas, S. Zeb, S. A. Hassan, R. Mumtaz, and S. A. R. Zaidi, "Joint Optimization of Age of Information and Energy Efficiency in IoT Networks," in *IEEE VTC Spring 2020*, Antwerp, Belgium, 2020.

[27] Y. Gu, H. Chen, C. Zhai, Y. Li, and B. Vucetic, "Minimizing age of information in cognitive radio-based IoT Systems: underlay or Overlay?," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10273–10288, 2019.

[28] J. Li, Y. Zhou, and H. Chen, "Age of Information for Multicast Transmission with Fixed and Random Deadlines in IoT Systems," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8178–8191, 2020.

[29] H. Azarhava, M. Pourmohammad Abdollahi, and J. Musevi Niya, "Age of information in wireless powered IoT networks: NOMA vs. TDMA," *Ad Hoc Networks*, vol. 104, article 102179, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5577735, 14 pages
https://doi.org/10.1155/2021/5577735

Research Article

A New Algorithm for Sketch-Based Fashion Image Retrieval Based on Cross-Domain Transformation

Haopeng Lei,[1] Simin Chen,[1] Mingwen Wang,[1] Xiangjian He,[2] Wenjing Jia,[2] and Sibo Li[1]

[1]*School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China*
[2]*School of Electrical and Data Engineering, University of Technology Sydney, Sydney NSW 2007, Australia*

Correspondence should be addressed to Simin Chen; simin_chen@jxnu.edu.cn

Received 6 January 2021; Revised 4 April 2021; Accepted 24 April 2021; Published 25 May 2021

Academic Editor: Amr Tolba

Due to the rise of e-commerce platforms, online shopping has become a trend. However, the current mainstream retrieval methods are still limited to using text or exemplar images as input. For huge commodity databases, it remains a long-standing unsolved problem for users to find the interested products quickly. Different from the traditional text-based and exemplar-based image retrieval techniques, sketch-based image retrieval (SBIR) provides a more intuitive and natural way for users to specify their search need. Due to the large cross-domain discrepancy between the free-hand sketch and fashion images, retrieving fashion images by sketches is a significantly challenging task. In this work, we propose a new algorithm for sketch-based fashion image retrieval based on cross-domain transformation. In our approach, the sketch and photo are first transformed into the same domain. Then, the sketch domain similarity and the photo domain similarity are calculated, respectively, and fused to improve the retrieval accuracy of fashion images. Moreover, the existing fashion image datasets mostly contain photos only and rarely contain the sketch-photo pairs. Thus, we contribute a fine-grained sketch-based fashion image retrieval dataset, which includes 36,074 sketch-photo pairs. Specifically, when retrieving on our Fashion Image dataset, the accuracy of our model ranks the correct match at the top-1 which is 96.6%, 92.1%, 91.0%, and 90.5% for clothes, pants, skirts, and shoes, respectively. Extensive experiments conducted on our dataset and two fine-grained instance-level datasets, i.e., QMUL-shoes and QMUL-chairs, show that our model has achieved a better performance than other existing methods.

1. Introduction

In recent years, the issue of fashion image retrieval has attracted increasing attention. Many research works have been reported on the tasks of clothing recognition [1, 2], clothing classification [3], and clothing retrieval [4, 5] due to their huge potential value to all walks of life. When consumers search for fashion images in online stores, mainstream retrieval methods are constrained by using text or example images as input. Due to the limited key words provided by online shopping platforms, it is difficult for consumers to retrieve the interested fashion image from the massive commodities by using text-based fashion image retrieval methods, while research on exemplar-based retrieval, where users provide an example image as the query, has recently received lots of interest in the community. However, the example images uploaded by users often suffer some problems during the actual retrieval process, such as poor light, posture changes, different shooting angles, and other factors. It is impractical to require users provide ideal example images as query input, which makes the fashion image retrieval even more challenging. A fast and effective fashion image retrieval method is currently the most urgent need for users.

Meanwhile, the way of human-computer interaction has changed dramatically due to the popularity of electronic devices. The way that humans use to retrieve fashion images is no longer restricted to using text and example images. Instead, people can use fashion sketches drawn on a touchscreen as input. For a long time, sketch is a general form of communication. Using sketch as input for retrieval has the following four advantages: (1) Fashion sketch contains more content than text does; (2) sketch is highly illustrative; (3) sketch is easy to express the styles of fashion image without

any ambiguity; (4) compared with example image, fashion sketch is easier to obtain; etc. Recently, the research related to sketch has flourished. Up to now, many problems have been studied, including sketch recognition [6, 7], sketch-based image retrieval (SBIR) [8, 9], and sketch-based 3D model retrieval [10], just to name a few. What is more, sketch-based fashion image retrieval is still relatively new. As the result, the urgent needs of users and the advantages of sketch-based retrieval provide us with a strong motivation to propose a more effective sketch-based image retrieval method, which uses sketch images as query input for fashion image retrieval.

With the above strong motivation, using sketch as input for fashion image retrieval faces, these problems to be solved in this paper. (1) Fashion sketches and fashion photos belong to two different domains. Compared with photos, sketches are composed of black lines on white background and look more abstract and lack information such as patterns, materials, and colours. This unique characteristic of the sketches increases the difficulty of fine-grained fashion image retrieval. (2) Most of the existing fashion image retrieval methods are based on example images input query. Images having similar visual content will be returned to the users by calculating the similarity between query image and database images. However, the input example images often have problems such as poor light, posture changes, different shooting angles, and complex background, which make it difficult to retrieve specific styles of fashion image for the users. (3) It is very difficult to collect fashion sketches. To the best of our knowledge, there is no large-scale dataset available for researchers to develop advanced solutions. In addition, we will need thousands of pairs of matching fashion sketches and images for our cross-domain deep learning. Therefore, it is challenging to create such this database covering different fashion image categories.

In this work, aiming to solve the problem of sketch-based fashion image retrieval, i.e., given a sketch of a fashion product, match it with the fashion photo in the dataset, and return the true-match fashion photo, we propose an efficient and reliable framework for fine-grained sketch-based fashion image retrieval to address these challenges. The framework of our method consists of three modules, including a cross-domain transformation module, a cross-domain feature extraction module, and a cross-domain similarity measurement module. We first use the cross-domain transformation module to transform sketches and photos into the same domain, and then, we adopt cross-domain feature extraction module to extract deep features of the query fashion sketch and the fashion photos in the retrieval dataset from the sketch domain and the photo domain, respectively. Next, we calculate the similarity between the transformed photo and fashion photos in photo domain and the similarity between the query sketch and transformed sketches in sketch domain. Finally, we fuse the two similarities in the different domains to achieve the final retrieval results.

The main contributions of this work are threefold:

(1) We propose a new algorithm for sketch-based fashion image retrieval based on cross-domain transformation, which transforms the fashion sketch and the fashion photo into the same domain before retrieval. Our proposed approach eliminates the requirement of rich annotation for the dataset and solves the heterogeneous problem of fashion sketches and fashion photos. In particular, the approach can effectively improve the retrieval accuracy of fashion image

(2) Most of the existing fashion image retrieval methods are based on example images input query. Images having similar visual content will be returned to the users by calculating the similarity between query image and database images. This method only calculates the similarity of the photo domain once. While we are doing cross-domain fashion image retrieval on two domains, we first transform the query fashion sketch into a fashion photo, use the transformed fashion photo to retrieve the fashion image dataset, and perform a similarity calculation of the photo domain. And then, we transformed all the fashion photos in the dataset into fashion sketches, use the query fashion sketch to retrieve the transformed fashion sketch dataset, and perform a similarity calculation of the sketch domain. Finally, we fuse the two similarities of the photo domain and the sketch domain to calculate the final similarity to obtain a more accurate retrieval result

(3) We contribute a new fine-grained sketch-based fashion image retrieval dataset, which contains 36,074 sketch-photo pairs covering 26 fashion types. As far as we know, it is the first comprehensive sketch-based fashion image retrieval dataset.

2. Related Work

2.1. Category-Level SBIR and Fine-Grained SBIR. Category-level sketch-based image retrieval (category-level SBIR) is conventional sketch-based image retrieval. It mainly focuses on retrieving images of the same category rather than the differences of intracategory. In recent years, the problem of category-level sketch for image retrieval [11–14] has been well studied. Most of the existing methods [11–13] first learn the common feature space of the sketch and the original image, perform similarity calculation and matching, and then retrieve the object that matches the target and return the object. However, with this method of learning, the common feature space between the sketch and the image may cause the model to collapse and therefore cannot achieve the expected results.

Fine-grained sketch-based image retrieval (fine-grained SBIR) is a new concept [8, 15–17]. The first attempt to solve the fine-grained SBIR was made by Li et al. [15], which mainly applied the deformable part-based model (DPM) to SBIR. Their definition of fine-grained emphasizes the viewpoint and observation of the object depicted by the sketch. As its consequence, an ideal recall image is the one that has a posture or perspective similar to the query sketch, regardless of whether the recalled image contains the same object.

However, it is very different from ours. Our definition of fine-grained is the same as that described in [8, 18], which emphasizes the details of the object depicted in the sketch. That is to say, for a retrieved image to match the query sketch, it must contain the same object instances. In recent years, with the development of artificial intelligence technology, CNN has significantly improved the performance in various computer vision tasks, such as image classification [19], image annotation [20], image retrieval [21–23], and medical image analysis [24, 25]. Khanday and Sofi [26] reviewed the state-of-the-art technology in computer vision by highlighting the contributions, challenges, and applications. In addition, the CNN-based feature extraction also demonstrates the excellent performance in sketch-based image retrieval, i.e., in 2015, Yu et al. [27] first abandoned the traditional feature extraction method of using convolutional neural networks and proposed a sketch-a-net network structure specially designed for free-hand sketch, which performed better than that proposed by Li et al. [18]. For example, when users search for a skirt, category-level SBIR can return a series of pictures of skirts for users, which is more complex than the way users input the text "skirt" instead of drawing the appearance of the skirt, whereas fine-grained SBIR can return the skirt corresponding to these details according to the sketch details entered by the user.

2.2. Fashion Image Datasets. Since the collection of sketches is not as easy as collecting photos, a significant obstacle to the research of sketch-based fashion image retrieval is the lack of benchmark datasets. As summarized in Table 1, the existing fashion image datasets all have different shapes and sizes and can be grouped according to single vs. multimodal. The single-modal datasets only consist of fashion photos, which are mainly used for fashion image recognition and retrieval from photo to photo. Moreover, most of the fashion photos contained in these single-modal datasets have complex backgrounds. Multimodal datasets support cross-domain tasks by providing sketches and photos. For example, the QMUL-shoes dataset [8] contains 419 sketch-photo pairs of shoes. The dataset contains simple images, but the only category is shoes. So, for the fashion category, the dataset is incomplete, and the size is small. Instead, our dataset has 36,074 fashion sketch-photo pairs, including clothes, pants, skirts, and shoes, covering almost all fashion categories. Compared with the QMUL-shoes dataset, it has more sketch-photo pairs and more comprehensive coverage of fashion image categories. Some example images in different datasets are shown in Figure 1. As it shows, photos in our dataset are as simple as those in QMUL-shoes.

2.3. Generative Adversarial Networks. Generative adversarial networks (GANs) [29] have made remarkable achievements in computer vision. A GAN model typically consists of two modules, i.e., the generator G and the discriminator D. In order to fool the discriminator, the generator should learn to generate false images that are indistinguishable from real images; meanwhile, the discriminator should learn to distinguish between real images and false images generated by the generator. The learning of GAN is a zero-sum game. The final result of the game is that, under ideal conditions, it is difficult for the discriminator to judge whether the image generated by the generator is real or false, that is, $D(G(z)) = 0.5$, where z is random noise.

Table 1: The comparison of our dataset with existing datasets.

Single-modal datasets	Number of images
DDAN [28]	341,021 photos
WTBI [4]	78,958 photos
DeepFashion [2]	Over 800,000 photos
Multi-modal datasets	**Number of images**
QMUL-shoe [8]	419 sketches, 419 photos
Our Fashion Image dataset	36,074 sketches, 36,074 photos

Since sketch and photo are heterogeneous, in order to overcome this challenge, GANs are used to eliminate the domain gap. The standard GAN is a one-way generation model that requires paired training data, i.e., all sketches in the sketch domain are converted to the same photo in the natural photo domain. To eliminate this requirement, Zhu et al. [30] proposed a cycle-consistency loss and CycleGAN. CycleGAN is a bidirectional generation model that can transform the sketch into the photo domain, and then back to the sketch domain, and can work in the absence of paired examples. Inspired by this approach, in this paper, we propose to transform images' domain by enforcing the cycle-consistency constraint. The backbone framework of our proposed model is based on UNIT [31] and VGG-16 [32]. We utilize the UNIT model to transform images' domain, where the UNIT model implies the cycle-consistency constraint, which can achieve perfect conversion between the images in different domains. Then, we use the VGG-16 network till the last convolutional layer to obtain the feature vectors, and then, we measure the similarity of feature vectors. Finally, the most similar photo is returned.

3. Proposed Method

3.1. Overview. In this section, we mainly describe the collection process of the Fashion Image dataset and the retrieval process of our proposed method. The framework of our method consists of three modules, including cross-domain transformation module, cross-domain feature extraction module, and cross-domain similarity measurement module. An overview of our proposed sketch-based fashion image retrieval model based on cross-domain transformation is illustrated in Figure 2.

Given a query fashion sketch s_q and the photos $p_n (n = 1, 2, \cdots, N)$ of the dataset, where N is the total number of fashion photos in the dataset, the aim is to retrieve the true-match fashion photo of the query sketch from the dataset. The retrieval procedure of our proposed method is divided into two streams: the sketch-based fashion photo retrieval stream and the sketch-based fashion sketch retrieval stream.

3.1.1. Sketch-Based Fashion Photo Retrieval Stream. First, in order to bridge the domain gap between the sketch and the

Figure 1: Examples of the images in different datasets showing their different styles.

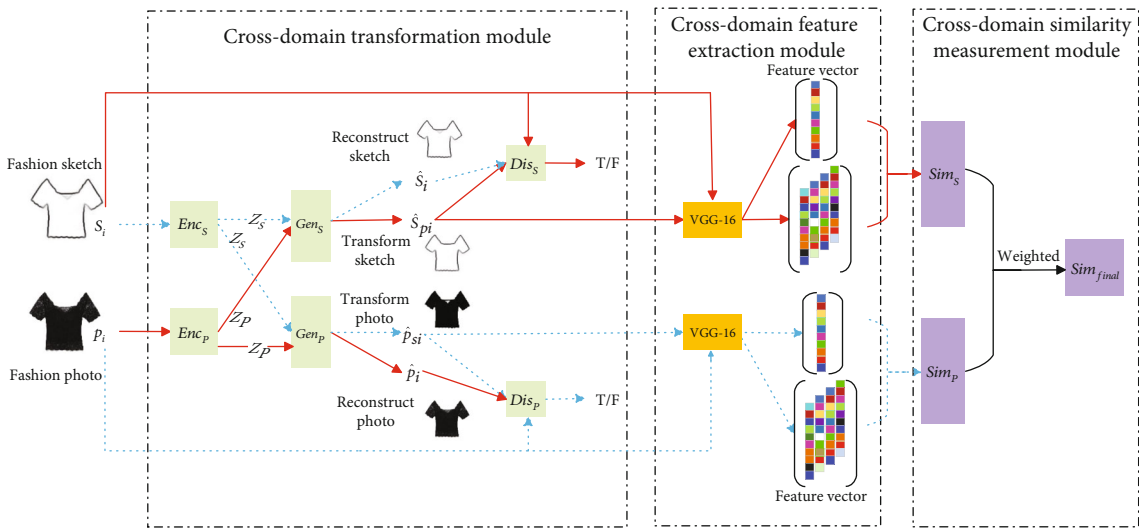


Figure 2: An overview of our proposed sketch-based fashion image retrieval model based on cross-domain transformation. Enc_S and Enc_P are encoders. Gen_S and Gen_P are generators. Dis_S and Dis_P are discriminator. The red solid arrows represent the sketch-based fashion sketch retrieval stream, and the blue dotted arrows represent the sketch-based fashion photo retrieval stream.

photo, the query sketch s_q first needs to be transformed into a fashion photo $\hat{p}_{s_q}$. Second, we extract the deep features of the transformed photo $\hat{p}_{s_q}$ and the fashion photos $p_n(n = 1, 2, \cdots, N)$ in the dataset through the cross-domain feature extraction module, respectively. Third, according to the obtained deep features, we calculate the similarity Sim_P between the transformed photo $\hat{p}_{s_q}$ and the fashion photos $p_n(n = 1, 2, \cdots, N)$. All the processes are shown by the dotted arrows in Figure 2.

3.1.2. Sketch-Based Fashion Sketch Retrieval Stream. Similar to the sketch-based fashion photo retrieval stream, we first map the fashion photos $p_n\,(n = 1, 2, \cdots, N)$ to the corresponding sketches $\hat{s}_{p_n}$. Second, we use the cross-domain

feature extraction module to extract the deep features of query sketch s_q and transformed sketches $\hat{s}_{p_n}(n = 1, 2, \cdots, N)$. Third, we calculate the similarity Sim_S between s_q and $\hat{s}_{p_n}(n = 1, 2, \cdots, N)$. All the processes are shown by the solid arrows in Figure 2.

After performing these two streams, for the query sketch s_q, we achieve the similarity Sim_P between the transformed photo $\hat{p}_{s_q}$ and the fashion photos $p_n\,(n = 1, 2, \cdots, N)$. Moreover, we also achieve the similarity Sim_S between the query sketch s_q and the transformed sketches $\hat{s}_{p_n}(n = 1, 2, \cdots, N)$. Then, we combine the sketch-based fashion photo retrieval stream and the sketch-based fashion sketch retrieval stream to improve the retrieval accuracy. Thus, we assign weights to these two similarities and add them to calculate the final similarity $\text{Sim}_{\text{final}}$. We rank the similarity $\text{Sim}_{\text{final}}$ to obtain an index table of the similarity between the query sketch s_q and the fashion photos $p_n\,(n = 1, 2, \cdots, N)$ in the dataset. Finally, according to the index table, the fashion photo which is the most similar to the query fashion sketch s_q in the dataset is returned as the retrieval result.

3.2. Fashion Image Dataset. We contribute a fine-grained Fashion Image dataset that contains a complete range of fashion types and can be used for fashion image cross-domain retrieval. We divide fashion images into four categories, i.e., clothes, pants, skirts, and shoes, and then further divide these four categories in detail. This dataset has three advantages. Firstly, as a multimodal (sketch and photo) fashion image dataset, it has a wide range of fashion categories, including clothes, pants, skirts and shoes. Secondly, it is a fine-grained dataset, where the clothes are divided into 11 subcategories, pants into 4 subcategories, skirts into 6 subcategories, and shoes into 5 subcategories. Thirdly, compared with other datasets of the same type, its size is larger, including 36,074 sketch-photo pairs. Next, we will describe the process of data collection and processing in detail.

3.2.1. Collecting Photos. The fashion photos we collect are mainly from three online shopping websites, including Taobao, Jindong, and Amazon, and a small part are from Baidu pictures and Google pictures. For fashion image, we divide them into four categories, i.e., clothes, pants, skirts, and shoes. Since the dataset we created is a fine-grained fashion image dataset, almost all relevant subcategories are included in each major category. For example, clothes consist of 11 subcategories, including suspender vests, short coats, long coats, short sleeve T-shirts, long sleeve T-shirts, short sleeve shirts, long sleeve shirts, vest, long cotton-padded jackets, short cotton-padded jackets, and leisure hoodies, covering almost all types of clothes. Finally, 12,603 representative cloth photos have been selected. For the collection of pants, skirts, and shoes, we also include different types and styles. We selected 5,610 photos of pants, including 4 types of back-belt pants, trousers, shorts, and jumpsuit; 13,321 photos of skirts, including 6 types of long skirts, mini-skirts, long sleeve dresses, short sleeve dresses, sleeveless dresses, and back-belt skirts; and 4,540 photos of shoes covering high heels, boots, flats, slippers, and sandals.

3.2.2. Collecting Sketches. The second step is to convert the collected photos to their corresponding sketches. We use the Structured Edge Detection Toolbox [33] to handle photos and obtain the edge maps, which are similar to free-hand sketches. Furthermore, in order to make the edge maps closer to the free-hand sketches, we performed an erasing operation on the edge maps, that is, to erase unnecessary line information in the edge maps and finally get the fashion sketches.

3.3. Cross-Domain Transformation Module. Since the fashion sketch and the fashion photo come from different domains, we transform the fashion sketch and the fashion photo into the same photo and sketch domain to bridge the domain gap. We propose a cross-domain transformation module, which is composed of 6 networks, namely the fashion sketch encoder Enc_S, the fashion photo encoder Enc_P, the fashion sketch generator Gen_S, the fashion photo generator Gen_P, the fashion sketch discriminator Dis_S, and the fashion photo discriminator Dis_P. The encoders include 3 convolutional layers and 4 residual basic blocks, which are used to encode the fashion sketch/photo into the latent code z_S/z_P. The generators include 4 residual basic blocks and 3 convolutional layers, which are used to decode the latent code z_S/z_P and generate the transformed fashion sketch/photo. The discriminators include 6 convolutional layers which are used to distinguish between the real fashion sketch/photo and the transformed fashion sketch/photo. The function of cross-domain transformation module includes self-reconstruction of the intradomain and the transformation of the cross-domain. We divide the cross-domain transformation module into two submodules $T_{S\to P}$ and $T_{P\to S}$. The first submodule $T_{S\to P}$ is used to transform the fashion sketch into the photo domain, and the second submodule $T_{P\to S}$ is used to transform the fashion photo into the sketch domain. The detailed cross-domain transformation training process is described as follows.

Suppose that the training sample pairs $\{s_i, p_i\}$ of the fashion sketch and the fashion photo are, respectively, given from the training dataset. We input the fashion sketch sample s_i into the first cross-domain transformation submodule $T_{S\to P}$, where the fashion sketch encoder Enc_S transforms the fashion sketch s_i into a latent code $z_S : z_S \sim \text{Enc}_S(s_i) = q_S(z_S \mid s_i)$, and the fashion sketch generator Gen_S decodes the latent code z_S to reconstruct the original input fashion sketch: $\hat{s}_i \sim \text{Gen}_S(z_S) = p_{\text{Gen}_s}(\hat{s}_i \mid z_S)$.

We use *VAE* [34–36] (variational autoencoder) to construct the encoder-decoder for the fashion sketch in our cross-domain transformation module. The objective function of the encode-decode process for the fashion sketch s_i is given by

$$\begin{aligned} L_{\text{Enc}_S} = {} & D_{\text{KL}}\left(q_S(z_S \mid s_i) \middle\| p_{\text{prior}}(z_S)\right) \\ & - \mathbb{E}_{z_S \sim q_S(z_S \mid s_i)}\left[\log\, p_{\text{Gen}_s}(\hat{s}_i \mid z_S)\right], \end{aligned} \tag{1}$$

where $q_S(z_S \mid s_i)$ represents that the fashion sketch encoder Enc_S maps the fashion sketch s_i into a latent code z_S and $p_{\text{prior}}(z_S)$ represents the prior distribution of the latent code

z_S. For simplicity, the prior distribution of latent code z_S can be assumed to follow a zero mean Gaussian distribution $N(0, I)$. $D_{\mathrm{KL}}(q_S(z_S \mid s_i) \| p_{\mathrm{prior}}(z_S))$ represents the KL divergence between the probability distribution $q_S(z_S \mid s_i)$ and $p_{\mathrm{prior}}(z_S)$. Therefore, the first term of this objective function is to ensure that the posterior distribution $q_S(z_S \mid s_i)$ of the latent code z_S is similar to the true prior distribution $p_{prior}(z_S)$. $p_{\mathrm{Gen}_s}(\hat{s}_i \mid z_S)$ represents the fashion sketch generator Gen_S that reconstruct the fashion sketch $\hat{s}_i$ given the latent code z_S. The second term of this objective function is the reconstruction loss which measures the reconstruction error between the reconstructed fashion sketch $\hat{s}_i$ and the original fashion sketch s_i.

Moreover, for the purpose of encouraging the reconstructed fashion sketch $\hat{s}_i$ to resemble the original fashion sketch s_i as closely as possible, we build the generative adversarial network GAN_S in our proposed cross-domain transformation module by combing the fashion sketch generator Gen_S and the fashion sketch discriminator Dis_S. The objective function of GAN_S is given by

$$\begin{aligned} L_{\mathrm{GAN}_S} = {} & \mathbb{E}_{s_i \sim p_{\mathrm{data}}(S)}[\log\ Dis_S(s_i)] \\ & + \mathbb{E}_{z_S \sim q_S(z_S \mid s_i)}[\log\ (1 - Dis_s(Gen_S(z_S)))], \end{aligned} \tag{2}$$

where $p_{data}(S)$ represents the probability distribution of all the fashion sketches in the training dataset. The fashion sketch generator Gen_S is used to reconstruct the fashion sketch $\hat{s}_i$ that looks similar to the original fashion sketch s_i given the latent code z_S, and the fashion sketch discriminator Dis_S is used to distinguish between the real original fashion sketch s_i and the reconstructed fashion sketch $\hat{s}_i$. Therefore, this objective function is to calculate the cross-entropy loss that encourages Gen_S to reconstruct the same original fashion sketch s_i and simultaneously provides the best discrimination ability to recognize the reconstructed sketch $\hat{s}_i$.

Then, in order to transform the fashion sketch into the photo domain, we input the latent code z_S of fashion sketch s_i into the fashion photo generator Gen_P to generate the transformed fashion photo $\hat{p}_{s_i}$, and we will input the generated fashion photo $\hat{p}_{s_i}$ and the real fashion photo p_i into the fashion photo discriminator Dis_P to determine whether an input fashion photo is the real fashion photo p_i or the transformed fashion photo $\hat{p}_{s_i}$. Fashion photo generator Gen_P and fashion photo discriminator Dis_P constitute the generative adversarial network [29] $GAN_{S \to P}$. The $GAN_{S \to P}$ objective function can be defined as

$$\begin{aligned} L_{GAN_{S \to P}} = {} & \mathbb{E}_{p_i \sim p_{data}(P)}[\log\ Dis_P(p_i)] \\ & + \mathbb{E}_{z_S \sim q_S(z_S \mid s_i)}[\log\ (1 - Dis_P(Gen_P(z_S)))], \end{aligned} \tag{3}$$

where $p_{\mathrm{data}}(P)$ represents the probability distribution of all the fashion photos in the training dataset. The fashion photo generator Gen_P tries to generate the fashion photo $\hat{p}_{s_i}$ that looks similar to the real fashion photo p_i given the latent code z_S, while fashion photo discriminator Dis_P tries to distinguish between real fashion photo p_i and the generated fashion photo $\hat{p}_{s_i}$.

Similarly, the fashion photo encoder Enc_P and the fashion photo generator Gen_P constitute a VAE network, which is used for reconstructing the fashion photos in the photo domain P. We input the fashion photo p_i into the second cross-domain transformation submodule $T_{P \to S}$. The fashion photo encoder Enc_P encodes the input fashion photo p_i into a latent code $z_P \sim Enc_P(p_i) = q_P(z_P \mid p_i)$, and the fashion photo generator Gen_P decodes the latent code z_P to reconstruct the fashion photo p_i; the self-reconstruction of the fashion photo p_i in photo domain P can be expressed as $\hat{p}_i \sim \mathrm{Gen}_P(z_P) = p_{\mathrm{Gen}_P}(\hat{p}_i \mid z_P)$. Thus, the objective function of the fashion photo p_i encode-decode process can be defined as

$$\begin{aligned} L_{\mathrm{Enc}_P} = {} & D_{\mathrm{KL}}\left(q_P(z_P \mid p_i) \middle\| p_{\mathrm{prior}}(z_P)\right) \\ & - \mathbb{E}_{z_P \sim q_P(z_P \mid p_i)}\left[\log\ p_{\mathrm{Gen}_P}(\hat{p}_i \mid z_P)\right], \end{aligned} \tag{4}$$

where the $q_P(z_P \mid p_i)$ represents the probability distribution of decoding the fashion photo p_i into the latent code z_P, the $p_{\mathrm{prior}}(z_P)$ indicates that the prior probability of the latent code z_P obeys the zero mean Gaussian distribution model $N(0, I)$, and the $p_{Gen_P}(\hat{p}_i \mid z_P)$ represents the probability distribution of the fashion photo generator Gen_P that reconstruct the latent code z_P to the fashion photo p_i. The first term is to penalize the latent code distribution that deviates from the prior distribution, and the second term is to constrain the reconstructed photo $\hat{p}_i$ to be similar to the input photo p_i.

What is more, we input the reconstructed photo $\hat{p}_i$ and the fashion photo p_i into the fashion photo discriminator Dis_P to determine whether an input fashion photo is true or false. The objective function of the generative adversarial network GAN_P composed of Gen_P and Dis_P can be defined as

$$\begin{aligned} L_{\mathrm{GAN}_P} = {} & \mathbb{E}_{p_i \sim p_{\mathrm{data}}(\mathrm{P})}[\log\ \mathrm{Dis}_P(p_i)] \\ & + \mathbb{E}_{z_P \sim q_P(z_P \mid p_i)}[\log\ (1 - \mathrm{Dis}_P(\mathrm{Gen}_P(z_P)))]. \end{aligned} \tag{5}$$

Similar to the equation (3), the fashion photo generator Gen_P is used to reconstruct the fashion photo $\hat{p}_i$ given the latent code z_P, and the fashion photo discriminator Dis_P is used to distinguish between the real original fashion photo p_i and the reconstructed fashion photo $\hat{p}_i$.

The fashion sketch generator Gen_S and the fashion sketch discriminator Dis_S constitute $\mathrm{GAN}_{P \to S}$, which is used for transforming the fashion photo p_i from the photo domain P to the sketch domain S, and the transformed fashion sketch is $\hat{s}_{p_i} = Gen_S(z_P \mid p_i)$; Dis_S is trained to distinguish between the real sketch s_i and the transformed sketch $\hat{s}_{p_i}$, which gives high scores to real sketches and

low scores to generated sketches. The $\mathrm{GAN}_{P\to S}$ objective function is given by

$$L_{GAN_{P\to S}} = \mathbb{E}_{s_i\sim p_{data}(S)}[\log\ Dis_S(s_i)] + \mathbb{E}_{z_P\sim q_P(z_P|p_i)}[\log\ (1 - Dis_S(Gen_S(z_P)))]. \tag{6}$$

At last, in order to improve the robustness and stability of the submodule $T_{S\to P}$ and $T_{P\to S}$, we need to ensure that the fashion photo $\widehat{p}_{s_i}$ transformed by the original fashion sketch s_i can be transformed back to the same sketch s_i, and the fashion sketch $\widehat{s}_{p_i}$ transformed by the original fashion photo p_i can be transformed back to the same photo p_i. Meanwhile, the original sketch features and photo features will not be lost after these twice transformation. Therefore, we utilize a cycle-consistency constraint [31] for the entire cross-domain transformation network. To achieve this goal, we input $\widehat{p}_{s_i}$ to the fashion photo encoder Enc_P for encoding and use fashion sketch generator Gen_s that decodes the latent code z_P to reconstruct the fashion sketch s_i. *VAE* can also be used to construct the encoder-decoder. The objective function of cycle-consistency constraint for fashion sketch is given by

$$L_{\mathrm{cyc}_S} = D_{\mathrm{KL}}\left(q_P\left(z_P \mid \widehat{p}_{s_i}\right)\middle\|p_{\mathrm{prior}}(z_P)\right) - \mathbb{E}_{z_P\sim q_P\left(z_P|\widehat{p}_{s_i}\right)}\left[\log\ p_{\mathrm{Gen}_s}(s_i \mid z_P)\right].$$

Similar to the above process, $\widehat{s}_{p_i}$ is input to the fashion sketch encoder Enc_S for encoding, and the fashion photo generator Gen_P is used to decode the latent code z_S to reconstruct the fashion photo p_i. The objective function of cycle-consistency constraint for fashion photo is given by

$$L_{\mathrm{cyc}_P} = D_{\mathrm{KL}}\left(q_S\left(z_S \mid \widehat{s}_{p_i}\right)\middle\|p_{\mathrm{prior}}(z_S)\right) - \mathbb{E}_{z_S\sim q_S\left(z_S|\widehat{s}_{p_i}\right)}\left[\log\ p_{\mathrm{Gen}_P}(p_i \mid z_S)\right]. \tag{8}$$

In summary, combined with equations (1), (2), (3), and (7), the total objective function of the fashion sketch cross-domain transformation submodule $T_{S\to P}$ is given by

$$L_{T_{S\to P}} = L_{\mathrm{Enc}_S} + L_{\mathrm{GAN}_S} + L_{\mathrm{GAN}_{S\to P}} + L_{\mathrm{cyc}_S}, \tag{9}$$

and combined with equation (4), (5), (6), and (8), the total objective function of the fashion photo cross-domain transformation submodule $T_{P\to S}$ is given by

$$L_{T_{P\to S}} = L_{\mathrm{Enc}_P} + L_{\mathrm{GAN}_P} + L_{\mathrm{GAN}_{P\to S}} + L_{\mathrm{cyc}_P}. \tag{10}$$

During the training process, we use the Adam optimizer to alternately optimize the objective functions $L_{T_{S\to P}}$ and $L_{T_{P\to S}}$. After the objective function optimization, the entire training process of the cross-domain transformation module can be completed. During the testing process, for any input query sketch s_q and fashion photos $p_n\ (n = 1, 2, \cdots, N)$ in the retrieval dataset, we can transform them into the same domain by using our proposed cross-domain transformation module.

3.4. Cross-Domain Feature Extraction Module. After the transformation of cross-domain images is completed, we deploy a symmetric CNN as the feature extraction module, which uses the VGG-16 [32] pretrained on ImageNet as the backbone network of the feature extraction module. If we use the pooling operation as a split point to group the entire VGG-16 network, we will get five sets of convolutions. The first two groups of convolutions have the same form, which is conv-relu-conv-relu-pool; the last three groups of convolutions have the same form, which is conv-relu-conv-relu-conv-relu-pool. In addition to the convolution group, VGG-16 has three fully connected layers at the end. However, in this paper, we use the VGG-16 network until the last convolutional layer obtains the feature matrixes. Finally, the size of feature vector obtained from VGGNet is 1×512.

For the sketch-based fashion photo retrieval stream, we use the VGG-16 network for the photo domain to extract features for photos $p_n\ (n = 1, 2, \cdots, N)$ in the Fashion Image dataset and store in the database as vectors. For the query sketch s_q, firstly, it is transformed into a fashion photo $\widehat{p}_{s_q}$, and then we use the same VGG-16 network to extract its deep feature vector of the same size. At last, we get the vector extracted from the transformed photo $\widehat{p}_{s_q}$ and the vectors extracted from photos $p_n\ (n = 1, 2, \cdots, N)$, which are used as the input of the cross-domain similarity measurement module.

For the sketch-based fashion sketch retrieval stream, photos $p_n\ (n = 1, 2, \cdots, N)$ in the Fashion Image dataset have been transformed into the corresponding sketches $\widehat{s}_{p_n}(n = 1, 2, \cdots, N)$ in the sketch domain *S*. And now, a transformed sketch dataset is obtained. We extract features for each transformed sketch $\widehat{s}_{p_n}$ by using the VGG-16 network for the sketch domain; then, the obtained feature vectors are stored in the database. We also get a deep feature vector by using the same VGG-16 network to extract features for the query sketch s_q. Finally, the feature vectors of query sketch s_q and the transformed sketches $\widehat{s}_{p_n}(n = 1, 2, \cdots, N)$ are obtained as the input of the cross-domain similarity measurement module.

After performing the above procedure, we can extract the features of the transformed photo $\widehat{p}_{s_q}$, fashion photos $p_n\ (n = 1, 2, \cdots, N)$, the transformed sketches $\widehat{s}_{p_n}(n = 1, 2, \cdots, N)$, and query sketch s_q. Then, we can measure the similarity between the fashion sketch and photo.

3.5. Cross-Domain Similarity Measurement Module. In this section, we measure the similarity of the obtained feature vectors. For the sketch-based fashion photo retrieval stream, the similarity Sim_P between the feature vector $F_{\widehat{p}_{s_q}}$ extracted from the transformed photo $\widehat{p}_{s_q}$ and the feature vectors F_{p_n}

(a)

(b)

(c)

Figure 3: Continued.

(d)

Figure 3: Examples of sketch-based fashion image retrieval results on our newly created dataset by using our proposed model. For each figure, the first column shows the query sketch, and the rest shows the top-10 retrieved fashion photos. The true matches are displayed in the box. (a) Examples of sketch-based clothes retrieval results. (b) Examples of sketch-based pant retrieval results. (c) Examples of sketch-based skirt retrieval results. (d) Examples of sketch-based shoe retrieval results.

extracted from the fashion photos $p_n\,(n = 1, 2, \cdots, N)$ is calculated as

$$\mathrm{Sim}_P = \frac{\sum_{d=1}^{512} F_{\hat{p}_{s_q}}^d \times F_{p_n}^d}{\sqrt{\sum_{d=1}^{512}\left(F_{p\wedge_{s_q}}^d\right)^2} \times \sqrt{\sum_{d=1}^{512}\left(F_{p_n}^d\right)^2}}. \tag{11}$$

For the sketch-based fashion sketch retrieval stream, the similarity Sim_S between the feature vector F_{s_q} extracted from the sketch s_q and the feature vectors $F_{\hat{s}_{p_n}}$ extracted from the transformed sketches $\hat{s}_{p_n}\,(n = 1, 2, \cdots, N)$ is calculated as

$$\mathrm{Sim}_S = \frac{\sum_{d=1}^{512} F_{s_q}^d \times F_{\hat{s}_{p_n}}^d}{\sqrt{\sum_{d=1}^{512}\left(F_{s_q}^d\right)^2} \times \sqrt{\sum_{d=1}^{512}\left(F_{s\wedge_{p_n}}^d\right)^2}}. \tag{12}$$

We assign different weights to balance the influence of these two similarities on the overall similarity and then add them to get the final similarity, which can be expressed as

$$\mathrm{Sim}_{\mathrm{final}} = \mu_1 \mathrm{Sim}_P + \mu_2 \mathrm{Sim}_S, \tag{13}$$

where $\mu_1 = 0.38$ and $\mu_2 = 0.62$ are used in our experiments.

Finally, the relevant fashion photos from the dataset can be returned to the user according to the $\mathrm{Sim}_{\mathrm{final}}$.

4. Experiments and Results

4.1. Experimental Settings

4.1.1. Dataset Preprocessing. There are 12,603 cloth sketch-photo pairs, 5,610 pant sketch-photo pairs, 13,321 skirt sketch-photo pairs, and 4,540 shoe sketch-photo pairs in our introduced Fashion Image dataset. Of these, we use 11,803/4,810/12,321/3,540 pairs for training clothes/pants/skirts/shoes, respectively, and the rest for testing. Before we conduct the experiments, we adjust all the sketches and photos into a unified size of 256×256. In the testing phase, in order to make the sketch in our test set closer to the free-hand sketch, we erased them to remove the details as much as possible, retained the rough outline, and then tested.

Table 2: The retrieval accuracy on our Fashion Image dataset.

Category	acc.@1	acc.@10
Clothes	0.966	0.994
Pants	0.921	0.966
Skirts	0.910	0.971
Shoes	0.905	0.978

We also conduct experiments on two fine-grained instance-level SBIR datasets, i.e., QMUL-shoes and QMUL-chairs datasets [8]. The QMUL-shoes dataset contains 419 shoe sketch-photo pairs, and we use 300 pairs for training and 119 pairs for testing when training our model. The QMUL-chairs dataset contains 297 chair sketch-photo pairs, and we use 200 pairs for training and the rest for testing.

4.1.2. Implementation Details. We used the open source PyTorch to train our models. During training, we use the Adam solver with a batch size of 1. The initial learning rate is set to 0.0001, and momentums are set to 0.5 and 0.999. The maximum number of training iterations is set to 470,000 when training on our Fashion Image dataset. Our method is implemented by NVIDIA Tesla P4 GPU and Intel E5-2630 CPU.

4.1.3. Evaluation Metric. In order to evaluate the performance of our sketch-based fashion image retrieval task, we use retrieval accuracy, denoted as "*acc.@K*." It means the proportion of all the search tasks that can rank the true-match photos in the top K search results.

Table 3: Retrieval accuracy versus different training iterations.

Test	Clothes		Pants		Skirts		Shoes	
Train	Top-1	Top-10	Top-1	Top-10	Top-1	Top-10	Top-1	Top-10
440,000	0.949	0.986	0.920	*0.967*	0.877	0.941	*0.910*	*0.980*
470,000	*0.966*	*0.994*	*0.921*	0.966	*0.910*	*0.971*	0.905	0.978
500,000	0.963	0.991	0.905	0.955	0.874	0.937	0.891	0.974

Figure 4: Retrieval examples obtained by using a complex sketch and a simple sketch for retrieval.

4.2. Experiments on our Fashion Image Dataset. We first conduct retrieval experiments on our Fashion Image dataset for clothes, pants, skirts, and shoes. Figure 3 shows results of our proposed model on the four fashion image retrieval tasks.

4.2.1. Clothes Transformation between Photos and Sketches. We use 11,803 clothes sketch-photo pairs for training the clothes cross-domain transformation model, and the rest for testing. When we obtain the clothes model, we use the model to transform 12,603 Clothes photos to their corresponding clothes sketches, which becomes the transformed clothes sketches dataset.

4.2.2. Pant Transformation between Photos and Sketches. We use 5,610 pant sketch-photo pairs in our Fashion Image dataset to learn to transform pant photos to their corresponding pant sketches. After we get pants transformation model, we use the model to transform 5,610 pant photos to their pant sketches, which forms the transformed pant sketches dataset.

4.2.3. Skirt Transformation between Photos and Sketches. We also use the images of skirts in our Fashion Image dataset to learn to transform skirt images between skirt photos and skirt sketches. After we obtain the skirt transformation, we transform 13,321 skirt photos to 13,321 skirt sketches, which is the transformed skirt sketches dataset.

4.2.4. Shoe Transformation between Photos and Sketches. Finally, we use 3,540 shoe sketch-photo pairs for training the shoe transformation model, and the rest for testing. When we obtain the shoe transformation model, we use the model to transform 4,540 shoe photos to 4,540 shoe sketches, which is the transformed shoe sketches dataset.

After the above experiments, we can get (1) a clothes/pant/skirt/shoe transformation model, respectively, and (2) a transformed fashion sketches dataset consisting of 12,603 transformed clothes sketches, 5,610 transformed pant sketches, 13,321 transformed skirt sketches, and 4,540 transformed shoe sketches.

4.2.5. Sketch-Based Clothes/Pant/Skirt/Shoe Retrieval. Given a clothes/pant/skirt/shoe sketch as a query sketch, firstly, we use the clothes/pant/skirt/shoe transformation model to transform the clothes/pant/skirt/shoe sketch into clothes/pant/skirt/shoe sketch to retrieve the translated fashion sketches dataset, respectively. Therefore, for the query sketch, we perform two retrievals and calculate the weighted sum of the two retrieval results to obtain the final retrieval result based on the clothes/pant/skirt/shoe sketch.

As shown in Table 2, we can find that compared with the correct match in the top-1, the correct match in the top-10 is a much easier task. For sketch-based clothes retrieval, our model ranks the correct match in the top-1 96.6% of the times for clothes. For sketch-based pant retrieval, the pant retrieval accuracy of top-1 and top-10 on our Fashion Image dataset are 92.1% and 96.6%. For sketch-based skirt retrieval, the top-1 and top-10 retrieval accuracy are up to 91.0% and 97.1%. As for sketch-based shoe retrieval, the accuracy of the true-match shoe photo ranked in the top-1 and top-10 are 90.5% and 97.8%. Figure 3 shows several retrieval results of our proposed model on our contributed dataset, the left part of the figure shows the query sketches, and the right part shows the top-10 retrieved fashion photos. If there are true-match photos in the top-10, most of their positions are in the top-1.

Finally, we used different types of fashion images to conduct experiments and analysed the impact of the training iterations to the retrieval accuracy. For different training iterations on the cross-domain fashion image transformation experiments, we calculated the retrieval accuracy achieved in different training iterations. The results were reported in Table 3. We found that, when the training iterations in the cross-domain fashion image transformation experiment phase were 470,000 iterations, the overall performance of

Table 4: Accuracy comparison with baselines on the Fashion Image dataset, QMUL-shoes and QMUL-chairs datasets.

Methods	Fashion Image dataset		QMUL-shoes		QMUL-chairs	
	acc.@1	acc.@10	acc.@1	acc.@10	acc.@1	acc.@10
BoW-HOG + rank-SVM [6]	—	—	0.174	0.678	0.289	0.670
ISN Deep + rank-SVM [37]	—	—	0.200	0.626	0.474	0.825
Dense-HOG + rank-SVM [8]	—	—	0.244	0.652	—	—
3DS Deep + rank-SVM [10]	—	—	0.052	0.217	0.062	0.268
Sketchy [17]	0.673	0.953	—	—	—	—
Our model	0.924	0.977	0.308	0.654	0.495	0.794

Figure 5: Examples of query sketch retrieval results on QMUL-shoes and QMUL-chairs datasets by using our proposed model.

retrieval accuracy in the test phase is the best, i.e., the top-1 retrieval accuracy for clothes, pants, skirts, and shoes is 96.6%, 92.1%, 91.0%, and 90.5%, respectively. What is more, our Fashion Image dataset contains sketches of different styles and different complexity, and some sketches have problems such as noise, unclear images, and missing strokes. We used these sketches to test and found that no matter how complex or simple the input sketch of the model is, the model can achieve good retrieval performance. Some retrieval results are shown in Figure 4.

4.3. Comparison with Baselines. We conduct experiments with baselines on three datasets: our Fashion Image dataset, QMUL-shoes, and QMUL-chair datasets [8]. The baselines we selected include Sketchy [17], BoW-HOG + rank-SVM [6], Improved Sketch-a-Net (ISN) [37], Dense-HOG + Rank-SVM [8], and 3D shape (3DS) [10]. Compared with baselines, our model transforms the sketches and photos to the same domain before retrieval, which improves the retrieval accuracy to a certain extent. The detailed comparative experiment results are shown in Table 4.

4.3.1. Comparison with Baselines on Our Fashion Image Dataset. We compare our model with Sketchy on our newly created Fashion Image dataset. Table 4 shows the top-1 and top-10 retrieval accuracy comparison with baseline on our

Table 5: Impact of sketch-based fashion photo retrieval stream and sketch-based fashion sketch retrieval stream on retrieval accuracy, implemented on our Fashion Image, QMUL-shoes, and QMUL-chairs datasets.

Methods	Fashion Image dataset		QMUL-shoes		QMUL-chairs	
	acc.@1	acc.@10	acc.@1	acc.@10	acc.@1	acc.@10
(1) Sketch-based fashion photo retrieval stream only	0.612	0.811	0.192	0.462	0.381	0.660
(2) Sketch-based fashion sketch retrieval stream only	0.911	0.957	0.154	0.500	0.351	0.680
(3) Our full model	0.924	0.977	0.308	0.654	0.495	0.794

dataset. As it shows in this table, our approach outperforms the Sketchy by 25.1% and 2.4% in top-1 retrieval accuracy and top-10 retrieval accuracy, respectively.

4.3.2. Comparison with Baselines on the QMUL-Shoes Dataset. In addition to experimentally comparing our approach with the baselines on our newly created dataset, we also evaluate our approach on the QMUL-shoes dataset. The QMUL-shoes dataset is a fine-grained instance-level SBIR dataset which contains 419 shoe sketch-photo pairs. On this dataset, we compare our model with BoW-HOG + rank-SVM, Improved Sketch-a-Net (ISN), Dense-HOG + Rank-SVM, and 3D shape (3DS). We compare our method with baselines in terms of top-1 and top-10 accuracies on QMUL-shoes dataset. From Table 4, we can find that our model can achieve compelling performance on the QMUL-shoes dataset and outperform the Dense-HOG + rank-SVM by 6.4% in top-1 retrieval accuracy. Examples of query sketch retrieval results on QMUL-shoes are presented in Figure 5.

4.3.3. Comparison with Baselines on the QMUL-Chairs Dataset. The QMUL-chairs dataset contains 297 chair sketch-photo pairs. We also conduct experiments on this fine-grained instance-level SBIR dataset. We compare our model with BoW-HOG + rank-SVM, Improved Sketch-a-Net (ISN), and 3D shape. In Table 4, we present the top-1 and top-10 accuracies of our model over other three models on the QMUL-chairs dataset for fine-grained SBIR. Compared with other methods, the top-1 retrieval accuracy of our model is higher than ISN Deep + rank-SVM by 2.1%. Examples of the query sketch and top-10 retrieval results on QMUL-chairs dataset are shown in Figure 5.

4.4. Ablation Studies. In this section, in order to demonstrate the advantage of combining the sketch-based fashion photo retrieval stream with the sketch-based fashion sketch retrieval stream, we conduct three ablation studies on our Fashion Image, QMUL-shoes, and QMUL-chairs datasets. Table 5 shows the results. The three ablation studies are as follows: (1) Only the sketch-based fashion photo retrieval stream is used, and the sketch-based fashion sketch retrieval stream is not used for retrieval. From Table 5, we find that on our Fashion Image dataset, the top-1 retrieval accuracy is 61.2%, and the top-10 retrieval accuracy is 81.1%. (2) Instead of using the sketch-based fashion photo retrieval stream, only the sketch-based fashion sketch retrieval stream is used for retrieval. As shown in Table 5, on our Fashion Image dataset, the retrieval accuracy of the top-1 is 91.1% and that of the top 10 is 95.7%. (3) Our full model of combining the two methods is combines the sketch-based fashion photo retrieval stream and the sketch-based fashion sketch retrieval stream for the ablation study. As shown in Table 5, the retrieval accuracy of top-1 reaches the highest on all three datasets, i.e., 92.4%, 30.8%, and 49.5%, respectively. After the above ablation studies, from Table 5, we can draw the conclusion that combining the two retrieval streams has further improved the retrieval results.

5. Conclusions and Future Work

In this paper, we first contributed a Fashion Image dataset, which contains 36,074 sketch-photo pairs for conducting research on sketch-based fashion image retrieval. We then introduced a new algorithm for sketch-based fashion image retrieval based on cross-domain transformation, which improves the retrieval accuracy by fusing the sketch-based fashion photo retrieval stream and sketch-based fashion sketch retrieval stream. Among them, the sketch-based fashion photo retrieval stream is to transform the query sketch into the corresponding photo in the natural photo domain and then use the transformed photo to retrieve the dataset. The sketch-based fashion sketch retrieval stream is to transform the fashion photos in the dataset to the corresponding sketches in the sketch domain and then use the query sketch to retrieve the transformed sketch dataset. The two similarities obtained by these two methods are first weighted, then added to obtain a hybrid similarity, and finally use the hybrid similarity for sketch-based fashion image retrieval.

Also, the current network has limitation that some sketches cannot be transformed into ideal photos. In future work, we will collect more fashion images of different styles and commit ourselves to research a network that can transform simple sketches into ideal photos and improve retrieval accuracy.

Data Availability

The Fashion Image data used to support the findings of this study are available from the corresponding author upon request, and the QMUL-shoes and the QMUL-chairs data used to support the findings of this study are available from this website: http://www.eecs.qmul.ac.uk/~qian/Project_cvpr16.html.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was jointly supported by the National Natural Science Foundation of China (61762050, 61876074, 61877031) and China Scholarship Council (201908360112). The authors would like to thank Fan Yang from the School of Computer and Information Engineering, Jiangxi Normal University, for his help in experimental design.

References

[1] Y. Kalantidis, L. Kennedy, and L. J. Li, "Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos," in *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, pp. 105–112, Dallas, USA, 2013.

[2] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: powering robust clothes recognition and retrieval with rich annotations," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1096–1104, Las Vegas, NV, USA, June 2016.

[3] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2691–2699, Boston, MA, USA, June 2015.

[4] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: matching street clothing photos in online shops," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3343–3351, Santiago, Chile, December 2015.

[5] X. Liang, L. Lin, W. Yang, P. Luo, J. Huang, and S. Yan, "Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1175–1186, 2016.

[6] Y. Li, T. M. Hospedales, Y. Z. Song, and S. Gong, "Free-hand sketch recognition by multi-kernel feature learning," *Computer Vision and Image Understanding*, vol. 137, pp. 1–11, 2015.

[7] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–10, 2012.

[8] Q. Yu, F. Liu, Y. Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy, "Sketch me that shoe," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 799–807, Las Vegas, NV, USA, June 2016.

[9] K. Pang, K. Li, Y. Yang et al., "Generalising fine-grained sketch-based image retrieval," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 677–686, Long Beach, CA, USA, June 2019.

[10] F. Wang, L. Kang, and Y. Li, "Sketch-based 3d shape retrieval using convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1875–1883, Boston, MA, USA, June 2015.

[11] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao, "Deep sketch hashing: fast free-hand sketch-based image retrieval," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2862–2871, Honolulu, Hawaii, USA, July 2017.

[12] R. Hu and J. Collomosse, "A performance evaluation of gradient field hog descriptor for sketch based image retrieval," *Computer Vision and Image Understanding*, vol. 117, no. 7, pp. 790–806, 2013.

[13] J. M. Saavedra and B. Bustos, "An improved histogram of edge local orientations for sketch-based image retrieval," in *Pattern Recognition. DAGM 2010. Lecture Notes in Computer Science, vol 6376*, M. Goesele, S. Roth, A. Kuijper, B. Schiele, and K. Schindler, Eds., pp. 432–441, Springer, Berlin, Heidelberg, 2010.

[14] J. Collomosse, T. Bui, and H. Jin, "Livesketch: query perturbations for guided sketch-based visual search," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2879–2887, Long Beach, CA, USA, June 2019.

[15] Y. Li, T. M. Hospedales, Y. Z. Song, and S. Gong, "Fine-grained sketch-based image retrieval by matching deformable part models," *British Machine Vision Association, BMVA*, 2014.

[16] P. Xu, Q. Yin, Y. Huang et al., "Cross-modal subspace learning for fine-grained sketch-based image retrieval," *Neurocomputing*, vol. 278, pp. 75–86, 2017.

[17] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: learning to retrieve badly drawn bunnies," *ACM Transactions on Graphics*, vol. 35, no. 4, p. 119, 2016.

[18] K. Li, K. Pang, Y. Z. Song, T. M. Hospedales, T. Xiang, and H. Zhang, "Synergistic instance-level subspace alignment for fine-grained sketch-based image retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5908–5921, 2017.

[19] S. Zou, W. Chen, and H. Chen, "Image classification model based on deep learning in internet of things," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6677907, 16 pages, 2020.

[20] S. Chen, M. Wang, and X. Chen, "Image annotation via reconstitution graph learning model," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8818616, 9 pages, 2020.

[21] Ş. Öztürk, "Image inpainting based compact hash code learning using modified U-Net," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1–5, Istanbul, Turkey, October 2020.

[22] C. Cui, X. Wu, J. Yang, and J. Li, "A novel DIBR 3D image hashing scheme based on pixel grouping and NMF," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8820436, 14 pages, 2020.

[23] Ş. Öztürk, "Two-stage sequential losses based automatic hash code generation using Siamese network," *European Journal of Science and Technology*, vol. 1, pp. 39–46, 2020.

[24] M. A. Shah, N. Y. Khanday, M. Purohit, and M. H. Gulzar, *Enhancement and Segmentation of Lung CT Images for Efficient Identification of Cancerous Cells*, 2016.

[25] Ş. Öztürk, "Stacked auto-encoder based tagging with deep features for content-based medical image retrieval," *Expert Systems with Applications*, vol. 161, p. 113693, 2020.

[26] N. Y. Khanday and S. A. Sofi, "Taxonomy, state-of-the-art, challenges and applications of visual understanding: a review," *Computer Science Review*, vol. 40, article 100374, 2021.

[27] Q. Yu, Y. Yang, Y. Z. Song, T. Xiang, and T. Hospedales, "Sketch-a-Net that beats humans," in *British Machine Vision Conference 2015*, pp. 1–12, Swansea, UK, 2015.

[28] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5315–5324, Boston, MA, USA, June 2015.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *In Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[30] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, Venice, Italy, October 2017.

[31] M. Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," 2017, https://arxiv.org/abs/1703.00848.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[33] 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, 2014.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, http://arxiv.org/abs/1312.6114.

[35] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and variational inference in deep latent gaussian models," *International Conference on Machine Learning*, vol. 2, 2014.

[36] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *Proceedings of Machine Learning Research*, vol. 48, pp. 1558–1566, 2016.

[37] Q. Yu, Y. Yang, F. Liu, Y. Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-Net: a deep neural network that beats humans," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 411–425, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5514003, 11 pages
https://doi.org/10.1155/2021/5514003

Research Article

RAPOT: An Adaptive Multifactor Risk Assessment Framework on Public Opinion for Trial Management

Weina Jiang,[1] **Qi Yong**,[1] **Ning Liu**,[1] **and Yuze Luo**[2]

[1]*School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China*
[2]*National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510006, China*

Correspondence should be addressed to Yuze Luo; yzluo26@126.com

Received 8 January 2021; Revised 8 April 2021; Accepted 17 April 2021; Published 17 May 2021

Academic Editor: Amr Tolba

Since public opinion from social media has a growing impact and supervision on trial, risk assessment on public opinion is increasingly important in refined trial management. However, the tremendous amount of public opinion and the insufficient historical logs of trial procedures bring challenges to risk assessment on public opinion. To address this, we propose an adaptive multifactor risk assessment framework on public opinion with fuzzy numbers. Initially, we establish a multilayer indicator model for assessing the risk of public opinion (POR) with multilayer analysis and decision methods. Then, we explore the association rules hidden in the process logs to update the indicator model periodically. Moreover, we design a public opinion analysis module for indicator evaluation, including analysis in public opinion sentiment, hot search, and social media coverage to deal with big data on social media. Especially, the public opinion sentiment is classified by topic-based BiLSTM (T-BiLSTM), which is more accurate. Finally, the fuzzy number similarity is employed to determine POR's level in the nine-level risk system. Experimental results validate the efficiency of our framework when assessing the POR.

1. Introduction

Serious and complicated cases bring severe challenges to trial management nowadays. Some of them have raised much attention due to their case type, related parties, and well-known judges. Simultaneously, people are used to expressing their opinions for the concerned cases on platforms such as Facebook, WeChat, Weibo, and Twitter. A mass of public opinion has both positive and negative impacts on trial procedures. Hence, public opinion assessment and supervision are crucial for credible trials. Actually, public opinion in social media has its characteristics, such as mass amount, fast propagation, and chaotic content. Furthermore, the mass data in social media reveals the inherent information we are concerned about. After analyzing multisource public opinion comprehensively, we could figure out its propagation mode to make POR's warning come earlier. Therefore, POR assessment is beneficial for early responding to negative public opinions and improving the court's initiative ability. There are two main tasks while accomplishing the task. One is to handle public opinion with big data theory, and the other is to conduct the risk assessment with insufficient historical data.

For the explosive comments that emerge on social media, sentiment analysis has become a research hotspot. Besides, sentiment analysis on comments about hot cases plays a vital role in promoting trial management. Thus, it is crucial to carry out an efficient analysis and supervision method for comments about cases. So far, research on machine learning-based sentiment analysis has a lot of achievements, such as KNN [1], maximum entropy [2], SVM [3], and Bayes [4]. Nowadays, with the rapid development and outstanding performance of deep learning, many researchers concentrate on methods with CNN [5], RNN [6], and LSTM [7] to improve the classification accuracy and have significant progress.

For risk assessment, due to insufficient historical data, together with the fuzziness and uncertainty of risks, researchers adopt a fuzzy set theory to analyze the risk [8]. Singh et al. [9] propose an assessment framework for risk

analysis of food disaster based on fuzzy similarity, and they quantitatively calculate the risk level of targets separately. The fuzzy similarity-based method performs well with a quantitative risk assessment for trial cases.

However, there still exists some challenges to achieve the assessment of POR. Firstly, there is no suitable indicator model for this task. An efficient assessment relies on fine-grained indicators and objective weights for indicators, and it remains unsolved. Secondly, comments about cases in the trial on social media have many characteristics that are hard to analyze. Hence, it remains much work to ensure the accuracy of sentiment classification for the specific use. Thirdly, how to evaluate risks quantitatively is not easy but crucial.

To address these issues, this paper implements a Risk Assessment framework on Public Opinion for Trial management (RAPOT). The framework provides a fine-grained risk assessment based on fuzzy numbers. By computing fuzzy number similarities, the framework decides its risk level in the nine-level assessment system. Our main contributions in this paper are as follows:

(i) Fine-Grained Risk Rating System. We employ fuzzy number similarities to achieve risk assessment with little historical data in trial procedure management. At first, a multilayer risk indicator model is established based on the analytic hierarchy process (AHP) method and extended technique for order preference by similarity to an ideal solution (extended TOPSIS) method. The model contains a fine-grained indicator layer, and each one contains a risk indicator and its impact factor. When assessing the risks, we transform both impact factors and indicator values into fuzzy numbers. Then, we aggregate the fuzzy numbers into one and rank the integrated one in the nine-level assessment system

(ii) Adaptive Indicator Model. Considering that the system logs accumulated during trial processing contain many latent association rules of the procedures, we propose the RApriori algorithm to explore the association rules. These latent rules are updated to the indicator model for improving the applicability and robustness of the model

(iii) Efficient Comment Sentiment Analysis. We define three kinds of input sources and submodules for indicator evaluation. Significantly, the sentiment of public opinion is classified based on topics. The sentiment analysis that we propose consists of single-pass-based topic clustering and T-BiLSTM-based sentiment analysis. Sentiment analysis for topics is precise and more comprehensive. Besides, our framework has extensive indicators such as the topic's heat and coverage of media

(iv) Experimental Evaluations. To demonstrate the performance of RAPOT, we conduct a case study with three cases that are paid much attention recently. The results illustrate that our framework is applicable and efficient in practical cases with a reasonable assessment level

The rest of this paper is structured as follows. We talk about the related work in Section 2. The RAPOT framework is described in Section 3. In Section 4, we illustrate the experimental results, and we conclude the paper in Section 5.

2. Related Work

Due to the fuzziness and uncertainty of risks, researchers adopt a fuzzy set theory to analyze the risk. The theory of fuzzy numbers has been widely applied in risk analysis [10], approximate reasoning [11], and risk pattern recognition [12]. For risk analysis, the existing methods can be divided into the fuzzy ranking-based [13], fuzzy inference-based [14, 15], fuzzy matrix-based [16, 17], and fuzzy number similarity-based [9] risk assessment models. Zhang et al. [13] figure out the risky area based on a water security evaluation framework by comparing the risk of related areas. Hence, the qualitative analysis measures the risk level comparatively. Nevertheless, in the trial, the POR of the two cases cannot be compared at the same pace. Karasan et al. [14] propose the safety and critical effect analysis (SCEA). Furthermore, it adopts Pythagorean fuzzy sets [18] to provide a comprehensive risk assessment. However, fuzzy inference-based methods are usually used in industry and are not suitable for trial applications. Can et al. [16] present a three-stage fuzzy risk matrix-based risk assessment and dynamically combine multicriteria decision-making with fuzzy logic. Though fuzzy matrix-based methods can reduce risk ties [19] efficiently, they still provide a qualitative assessment that is not precise enough. As for similarity-based method, Khorshidi and Nikfalazar [20] present an improved method to compute the degree of similarity between generalized fuzzy numbers. The proposed method has been used for fuzzy risk analysis, and it could determine each manufacturer's risk level. In summary, the similarity-based model is suitable for the quantitative risk assessment for an individual object. At the same time, risks in the trial process management system (TPMS) are quite fuzzy and uncertain in fact. Besides, the historical data have not been digitalized well. So we adopt a fuzzy number similarity-based model to achieve risk assessment.

The existing fuzzy number similarity-based methods always have three main modules. They are the risk indicator model, risk aggregation, and risk level determination. Among them, fuzzy number similarity calculation is important for risk level determination precisely. Referring to fuzzy number similarities, researchers have defined various features of generalized fuzzy number (GFN) to distinguish the numbers, such as the center of gravity (COG) [21], the area [20], and the radius of gyration [22]. Then, researchers adopt geometric distance, Hausdorff distance [23], and so on to measure the similarity of the feature values. Xu et al. [24] present a COG-based method while with a limitation that two different fuzzy numbers may have the same COG. To address the limitation, Yong et al. [25] employ ROG of the area to measure the similarities. Moreover, Chutia and Gogoi [10] expand

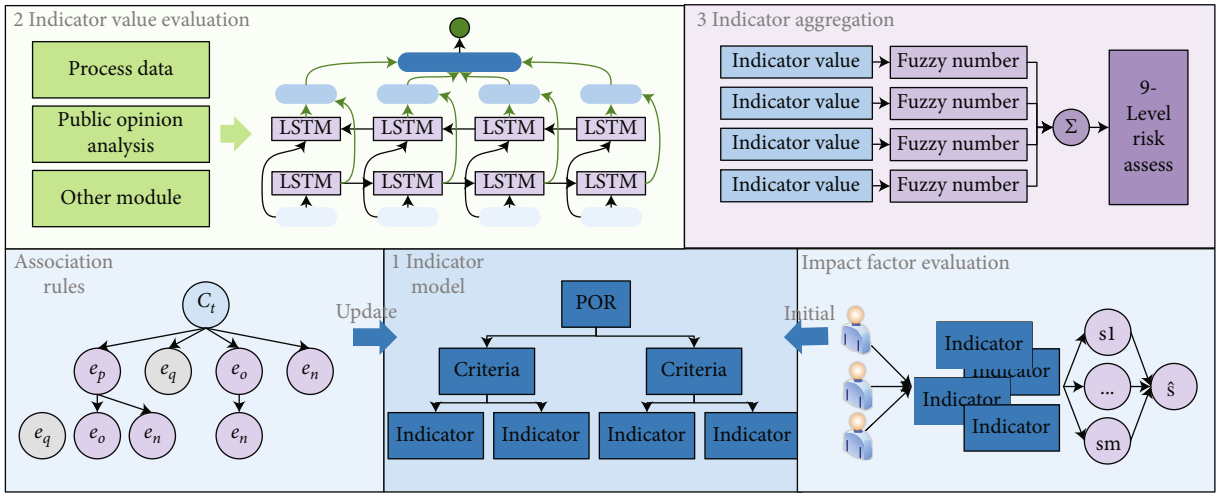


Figure 1: The framework of RAPOT.

GFN with left height and the right height to further distinguish traditional GFNs with the same COG. However, these two methods still suffer from invalid results. Therefore, we select a similarity measurement on generalized fuzzy numbers to map the integrated fuzzy number into a linguistic term in the nine-level risk system [26]. The similarity measure algorithm we employ constrains the similarity of two fuzzy numbers in the range of [0, 1], with fewer invalid results and at the same time has a high distinguishability.

3. Framework of Risk Assessment of Public Opinion

In this section, we discuss the critical issues while assessing the POR. Firstly, we present the risk indicator model in Sections 3.1 and 3.2. Secondly, we talk about evaluating risk indicators and the public opinion sentiment analysis in Section 3.3. Then, we explain the fuzzy number similarity-based risk assessment method in Section 3.4.

Figure 1 is the framework of our RAPOT. It includes a risk indicator model, indicator evaluation module, and risk aggregation module. In the beginning, a multilayer indicator model is built to define the fine-grained risk indicators with corresponding impact factors, and the model is dynamically updated by exploring new association rules on system logs. Then, the indicator evaluation module figures out the indicator values based on process data, data from the public opinion analysis module, and the other systems. The indicator aggregation module is aimed at deciding the risk level with the impact factors and the risk probabilities.

3.1. Risk Indicator Model Initialization. To overcome the difficulty of lacking historical data, we employ AHP and extended TOPSIS to construct an initialized risk indicator model. The hierarchy model defines amounts of risk indicators along with their impact factors. Figure 2 describes the procedures for building our indicator model. First, a hierarchy structure is built, and an evaluation dataset for risk indicators is collected based on AHP. Then, an evaluation matrix for the risk indicators is constructed based on the collected

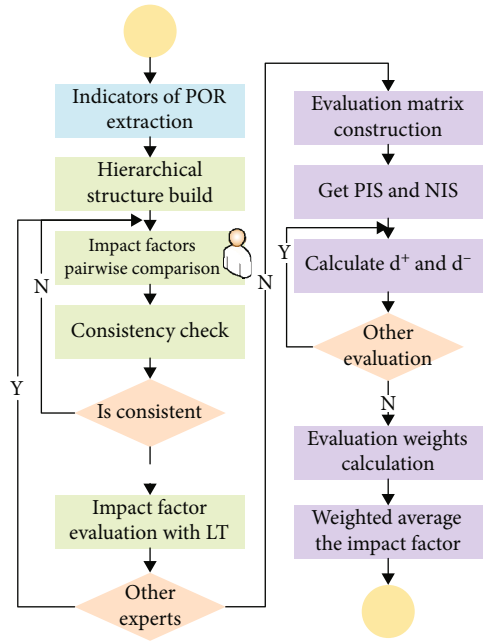


Figure 2: The flowchart of risk indicator model initialization.

dataset. We adopt extended TOPSIS to analyze the evaluation dataset to calculate the impact factors of the indicators. Our model construction method combines AHP and extended TOPSIS to work out a group of accurate impact factors with limited historical data.

3.1.1. Hierarchical Structure Determination. AHP is an efficient multilayer analysis and decision method [27, 28]. It first composes the decision problem into a hierarchy of subproblems that each one can be treated independently. Once the hierarchy is built, the expert group evaluates the elements in the same layer by comparing them to each other according to their impact on the father element. Table 1 shows the 1-9 scales used to evaluate each element's impact factor. The APH transforms the evaluations to numerical values that can be calculated over the decision problem's entire range.

Table 1: 1-9 scales of relative importance [29].

Intensity of importance	Definition
1	Equal importance
2	Weak
3	Moderate importance
4	Moderate plus
5	Strong importance
6	Strong plus
7	Very strong or demonstrated importance
8	Very, very strong
9	Extreme importance

Finally, a priority is derived for each element in the hierarchy by iteratively verifying the comparison matrix's consistency after adjusting the priorities each time.

At first, we refer to expertise, existing laws, regulations, and the classical hot cases and form the set of risks as $R = \{r_1, r_2, \cdots, r_N\}$, where N is the number of risks. Then, the hierarchical structure is established based on AHP. As shown in Figure 3, our risk indicator model consists of three layers:

(i) Objective Layer (OL). Risk assessment of public opinion for trial management is the objective of our work. We need to figure out the impacts of public opinion on the trial procedure

(ii) Criteria Layer (CL). The elements in this layer are the judge, the parties involved, the case, and the public opinion. The expert group defines the elements referring to the existing documents

(iii) Indicator Layer (IL). This layer contains the indicators which would impact the trial procedure by public opinion. Each indicator belongs to their father elements in the criteria layer

After that, an evaluation dataset is collected to gain the indicators' impact factors, and the impact factor represents the indicator's weight when integrating the POR. To evaluate the impact factor accurately, the expert compares the risk indicators with pairs to complete a comparison matrix as

$$\Delta = \begin{bmatrix} \delta_{11} & \cdots & \delta_{1N} \\ \vdots & \ddots & \vdots \\ \delta_{N1} & \cdots & \delta_{NN} \end{bmatrix}, \quad (1)$$

where δ_{ij} is the comparison value of r_i and r_j. The expert assigns the value δ according to Table 1. Then, the consistency of Δ has to be verified by

$$\mathrm{CI} = \frac{\lambda - n}{n - 1}, \quad (2)$$

where λ is the maximum eigenvalue of Δ and n is the dimension of the matrix. The consistency is complete when CI = 0 and decreases with CI increasing. Then, AHP uses a random consistency indicator RI to define a refined CR which is

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}}. \quad (3)$$

When CR < 0.1, the matrix Δ is consistent and RI is a predefined dictionary [29]. If the validation fails, the expert has to adjust the comparison matrix until the validation comes to success.

The eigenvector of the approved evaluation matrix gives a sort of risk indicators by their impact factors. For risk assessment with fuzzy numbers, the expert assigns a linguistic term in LT ={"AbsolutelyLow (AL)", "VeryLow (VL)", "Low (L)", "FairlyLow (FL)", "Medium (M)", "FairlyHigh (FH)", "High (H)", "VeryHigh (VH)", "AbsoluteHigh (AH)"} to each risk indicator based on the order.

3.1.2. Impact Factor Calculation. Serval law experts evaluate the impact factors according to our hierarchical structure and construct an evaluation dataset. The dataset contains several evaluation items $\{l_1, l_2, \cdots, l_M\}$ for M experts and l_m is consists of $l_{m1}, l_{m2}, \cdots, l_{mN}$, each item comes from a law expert for r_n in set R. Then, we employ TOPSIS to aggregate the evaluations of different experts. TOPSIS is a multicriteria decision analysis method, which identifies weights for each criterion by calculating the geometric distances from each alternative to the positive ideal solution and the negative ideal solution, respectively [30]. When evaluating the risk indicator's impact factor, the positive ideal solution is defined as the lowest impact on cost optimization. Namely, the lower impact of the risk indicator brings less cost in risk prevention and control. Hence, we adopt the extended TOPSIS [31] to calculate the impact factors for the POR assessment designed for the trial scene.

First, an evaluation matrix with linguistic terms is established based on the dataset as

$$L = \begin{bmatrix} l_{11} & \cdots & l_{1N} \\ \vdots & \ddots & \vdots \\ l_{M1} & \cdots & l_{MN} \end{bmatrix}, \quad (4)$$

where M, N are the number of experts and risk indicators. In the matrix, l_{mn} is given by expert m for the indicator n to measure the importance of the indicator. And then, l_{mn} is transformed into a fuzzy number according to Table 2 for weight fusion of impact. After that, we get an evaluation matrix with fuzzy numbers.

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1N} \\ \vdots & \ddots & \vdots \\ s_{M1} & \cdots & s_{MN} \end{bmatrix}, \quad (5)$$

here s_{mn} is a generalized fuzzy number represented as $(a, b, c, d; w)$ and $a, b, c, d, w \in R$.

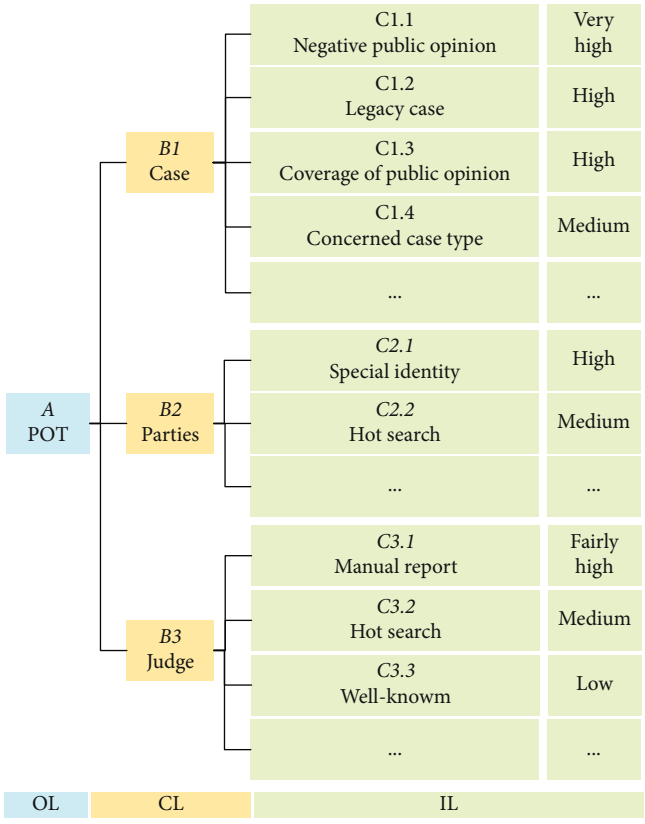


Figure 3: The indicator model for the POR.

Table 2: The transform from linguistic terms to the fuzzy numbers [26].

Lingustic terms	Generalized fuzzy numbers
AbsolutelyLow	(0.0, 0.0, 0.0, 0.0; 1.0)
VeryLow	(0.0, 0.0, 0.02, 0.07; 1.0)
Low	(0.04, 0.1, 0.18, 0.23; 1.0)
FairlyLow	(0.17, 0.22, 0.36, 0.42; 1.0)
Medium	(0.32, 0.41, 0.58, 0.65; 1.0)
FairlyHigh	(0.58, 0.63, 0.80, 0.86; 1.0)
High	(0.72, 0.78, 0.92, 0.97; 1.0)
VeryHigh	(0.93, 0.98, 1.0, 1.0; 1.0)
AbsolutelyHigh	(1.0, 1.0, 1.0, 1.0; 1.0)

In the extended TOPSIS, the positive and negative ideal solutions are

$$\begin{aligned} \text{PIS} &= [s_1^+, s_2^+, \cdots, s_N^+], \\ \text{NIS} &= [s_1^-, s_2^-, \cdots, s_N^-], \end{aligned} \tag{6}$$

here, s_n^+ and s_n^- are defined as

$$\begin{cases} s_n^+ = \min\ (s_{mn}) \\ s_n^- = \max\ (s_{mn}) \end{cases}, \tag{7}$$

Then, the distance between $s_m = [s_{m1}, s_{m2}, \cdots, s_{mN}]$ and the positive ideal solution PIS is calcuated as

$$d^+ = d(s_m, \text{PIS}) = \sqrt{\sum_{n=1}^{N} \otimes \sum_{j=1}^{4} \otimes \left(s_{mnj} - s_{nj}^+\right)^2}. \tag{8}$$

Similarly, the geometric distance between s_m and the negative ideal solution NIS is

$$d^- = d(s_m, \text{NIS}) = \sqrt{\sum_{n=1}^{N} \otimes \sum_{j=1}^{4} \otimes \left(s_{mnj} - s_{nj}^-\right)^2}, \tag{9}$$

here $s_{mn} = (s_{mn1}, s_{mn2}, s_{mn3}, s_{mn4}; w)$, $s_n^+ = (s_{n1}^+, s_{n2}^+, s_{n3}^+, s_{n4}^+; w)$, and $s_n^- = (s_{n1}^-, s_{n2}^-, s_{n3}^-, s_{n4}^-; w)$ are generalized fuzzy numbers. After that, we obtain the weight of each alternative s_m by normalizing the distance ratios as

$$\lambda_m = \coprod \left(\frac{2}{\left| d_m^+ - d_m^- \right|} \right). \tag{10}$$

Finally, the impact factor of indicator n is calculated by weighted summing the alternatives as

$$\hat{s}_n = \frac{\sum_{m=1}^{M} \lambda_m \otimes s_{mn}}{\sum_{m=1}^{M} \otimes \lambda_m}. \tag{11}$$

3.2. Risk Indicator Model Update. Considering that the trial process is strict and complicated, POR's initial indicator model can be hardly applicable to the POR assessment continuously. Also, the system logs accumulated during trial processing contain many latent association rules of the procedures. Figure 4 shows a fragment of the trial process, each block is a process node, and each ellipse represents the risk confirmation. Therefore, we propose a reversed Apriori (RApriori) algorithm to explore the association rules hide in the system logs. The association rule we want to search is defined as $[e_i, \cdots, e_j, c_t]$, here e_i represents a failed rule check in the process node i and c_t is a risk confirmation node. By investigating the practical TPMS, we figure out the process nodes are arranged in a single sequence. According to it, we optimize the classical Apriori by ordering the nodes and extending the association set in reverse. The details of the proposed RApriori are shown in Algorithm 1.

In the algorithm, we assign numerical codes to both process nodes and risk confirm nodes based on their sequence in trial. Firstly, the search of latent association rules always starts from a frequent risk confirm node c_t and set it as the root of the tree T we show in Figure 5. Secondly, the frequent process nodes whose numerical codes less than c_t are reversely sorted in a candidate list $[e_q, e_p, \cdots, e_i]$. Thirdly, we join each item in the list with c_t to form a set separately, such as $\{c_t, e_q\}$, and then check the corresponding support score

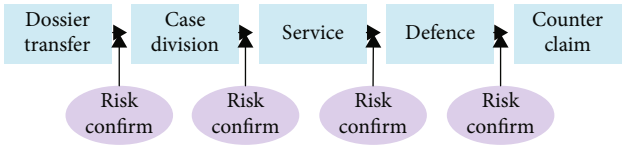


Figure 4: A fragment of the trial process.

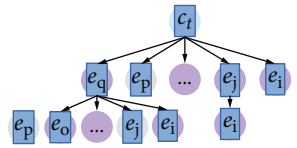


Figure 5: The procedures of joinSet operation.

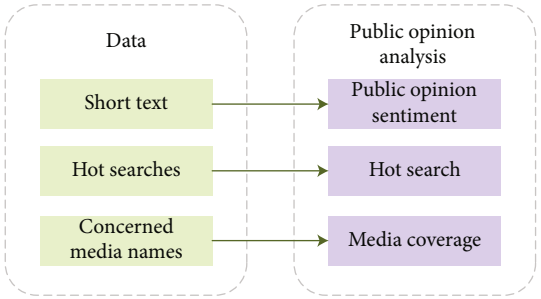


Figure 6: The structure of the social media analysis module.

Require: system logs generate from T1 to T2
Ensure: association rules $[[e_i, \cdots, e_j, c_t], \cdots]$
1: total = len(logs)
2: ens = getSupportErrorNodes(logs)
3: cns = getSupportConfirmNodes(logs)
4: **for** $t = 1$; $t <$ len(cns); $t++$ **do**
5: cn = cns[t]
6: rlogs = getRelatedLogs(cn)
7: rens = getRelatedErrorNode(cn)
8: cSet = joinSet(cn, rens)
9: cSet = getSupportSet(cSet, rlogs, total, sRecord)
10: sList = sort(cSet)
11: $k = 2$
12: **repeat**
13: fSet = union(cSet)
14: cSet = joinSet(cSet, sList, $k + 1$)
15: cSet = getSupportSet(cSet, rlogs, total, sRecord)
16: $k = k + 1$
17: **until** cSet is empty
18: **end for**
19: rules = getConfidenceRule(fSet)

Algorithm 1: RApriori

to create layer 2. The support score is defined as

$$\text{support}(A) = \frac{\text{count}(A)}{|D|}, \tag{12}$$

where A is a set and $|D|$ is the amount of the logs. Fourthly, the tree moves to the next layer by orderly combining a set in the current layer with items in the candidate list that are less than the minimum node in the set. Then, iteratively increase the height of T until there is no more satisfied new set. At last, we calculate the support score of the satisfied sets and work out the association rules. The support score is defined as

$$\text{support}\left(\{e_i, \cdots, e_j\} \Rightarrow c_t\right) = \frac{\text{count}\left(\{e_i, \cdots, e_j, c_t\}\right)}{\text{count}\left(\{e_i, \cdots, e_j\}\right)}. \tag{13}$$

The RApriori method is executed regularly, and the searched association rules are added to update the indicator model of POR. The experimental results show that our algorithm decreases the computational complexity significantly.

3.3. Risk Indicator Evaluation and Public Opinion Analysis. Besides the indicator factor, we have to calculate the probability of indicator occurrence, which we call the indicator value. The data sources of value computing can be divided into three categories: (1) social media, (2) manual input, and (3) document analysis. For indicator C3.1, the judge can report the POR during the trial. As for C1.2, C1.4, C2.1, and C3.3, the indicator values are determined by the other subsystems in the TPMS, for instance, the case division system. Apart from them, the values of indicators C1.1, C1.3, C2.2, and C3.2 are inferred from the social media analysis module. Figure 6 illustrates the structure of our module for social media analysis. It is composed of three parts listed as follows:

(i) Analysis in Public Opinion Sentiment. This part explores how people are interested in the case and how intensely they discuss the related topics. If the public cares much about the case and shows negative sentiment in their expressions, the indicator value will be large. On the contrary, the indicator value will come near zero

(ii) Analysis in Hot Search. The judge or the parties frequently searched in social media is an important indicator that this case may have the POT during the trial

(iii) Analysis in Media Coverage. If the media in our maintained important-media list has taken part in the related topic, this case's media coverage will increase. The POT level increases with the coverage reaching a threshold

In this section, we mainly describe the public opinion sentiment based on topics. The comments collected from social media related to the case are divided into some topics to address this. Then, the texts and the related topics are fed into a neural network to train a classifier used to analyze the sentiment. The details are as follows.

3.3.1. Input Embedding. Firstly, a short text is split into a word sequence $W = \{w_1, w_2, \cdots, w_n\}$ which contains n words. After that, we transform words to vectors by a Word2vec model [32] and obtain the embedding matrix E_W which consists of all word embeddings.

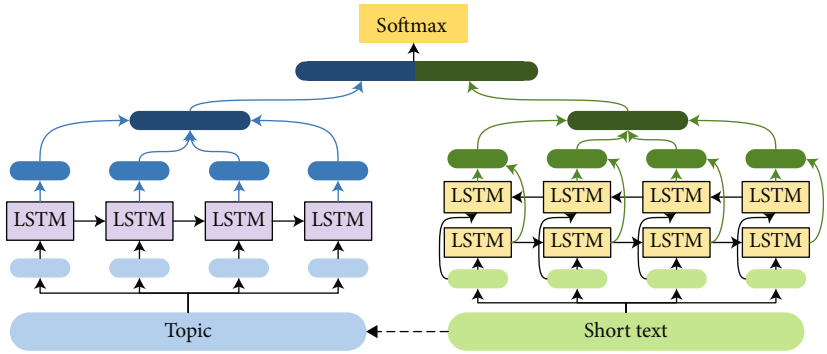


Figure 7: The structure of T-BiLSTM.

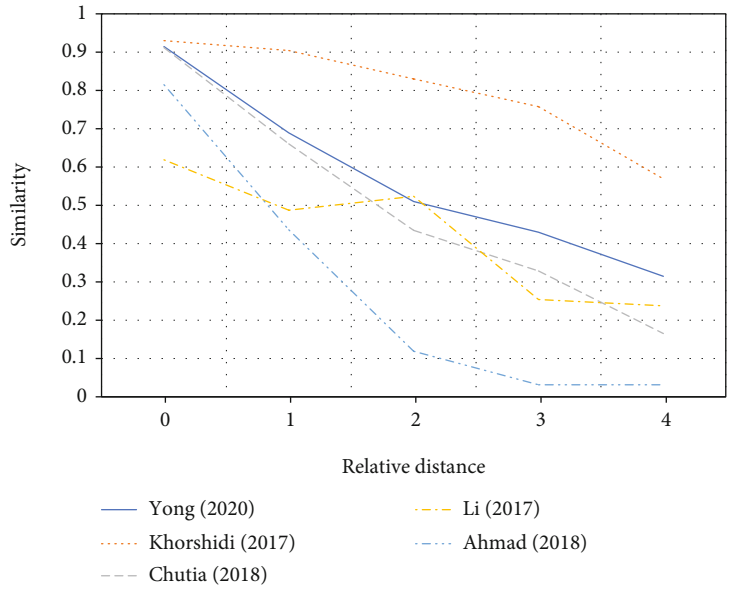


Figure 8: The comparison in attenuation of similarities with increasing distance for the five methods.

Table 3: The parameters of simulation datasets.

Parameters	Value
Count of process nodes	80
Count of confirm nodes	80
Error rate	0.15
Confirm rate	0.15

3.3.2. Topic Clustering. Single-pass clustering [33] with the cosine similarity is employed to literately partition m short texts into k clusters, the topics can be represented as $T = \{t_1, t_2, \cdots, t_k\}$, and t is a set of some keywords. The similarity is calculated as

$$dist(s_i, s_j) = \frac{s_i \cdot s_j}{|s_i||s_j|} \tag{14}$$

where s_i and s_j are vectors of two short texts. Then, the keywords in the clusters are detected to be the topics. Moreover, we get the embedding matrix E_T, which contains all keyword embeddings of a topic through word embedding.

3.3.3. T-BiLSTM-Based Comment Sentiment Analysis. Since BiLSTM [34] has been proven efficient for sentiment analysis, we propose the T-BiLSTM network to train a text sentiment classifier. Figure 7 illustrates the structure of the T-BiLSTM. On the right side, we employ a BiLSTM layer to capture the contextual features of the text. On the left side, we adopt a LSTM layer to explore the contextual features of the topic. Next, we concatenate the outputs of both sides and feed it into a softmax layer. The above processes are represented as

$$\begin{aligned} H_W &= \text{BiLSTM}(E_W), \\ H_T &= \text{LSTM}(E_T), \\ H &= [H_W, H_T], \\ p &= \text{softmax}(W * H + b), \end{aligned} \tag{15}$$

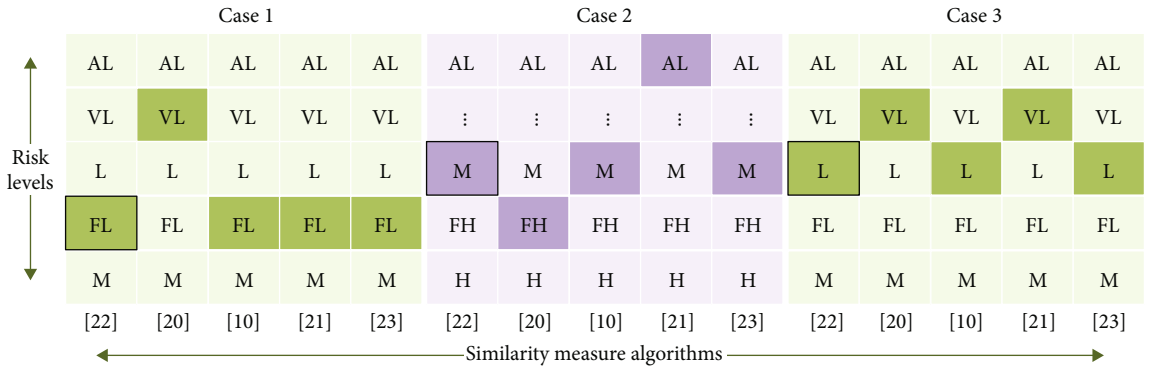


Figure 9: The comparison in results of five similarity measure methods.

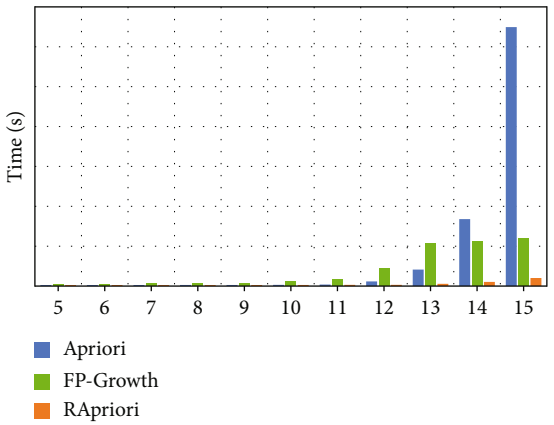


Figure 10: Time costs with different rule lengths.

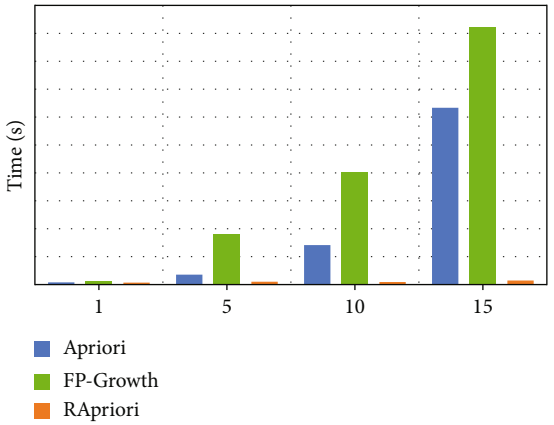


Figure 11: Time costs with different rule counts.

where W and b are the weight matrix and bias, respectively. In addition, we use cross-entropy loss to lead the network training.

3.3.4. Evaluation of Indicator C1.1. The public opinion sentiment for topics is defined as

$$f_i = \min\left(\max\left(\sum_{M}^{i=1} \boxtimes \frac{\text{neg}_i}{\tau}, 1\right), 9\right). \tag{16}$$

Here, neg_i is the number of negative comments in topic t_i, τ is the threshold which is used to testify whether a topic is discussed widely, and the evaluation of indicator C1.1 is calculated as

$$v = \left\lceil \frac{N_1 \times f_1 + N_2 \times f_2 + \cdots + N_k \times f_k}{M} \right\rceil, \tag{17}$$

where N_i is the count of texts in topic t_i, and M is the total amount of texts in the case.

3.4. Risk Assessment on Public Opinion for Trial Management. In this section, we describe the fuzzy number similarity-based risk assessment module which evaluates the risk level in the nine-level risk system. At first, the risk indicator evaluations we talk about in Section 3.3 are converted into fuzzy numbers as

$$\text{Evaluate}_i = \mathbf{GFNS}_{v_i,1}. \tag{18}$$

Here, $v_i \in [1, 9]$ and $\mathbf{GFNS}_m = (\text{lt}_m, \text{GFN}_m)$; lt_m is a linguistic term which is in $\{\text{AL}, \text{VL}, \text{L}, \text{FL}, \text{M}, \text{FH}, \text{H}, \text{VH}, \text{AH}\}$, and GFN_m is a generalized fuzzy number defined in Table 2. Since the risk of public opinion has various indicators, the risk assessment module aggregating risk of each indicator by the weighted average method is

$$R = \frac{\sum_{i=1}^{N} \boxtimes \text{Impact}_i \otimes \text{Evaluate}_i}{\sum_{i=1}^{N} \boxtimes \text{Impact}_i}. \tag{19}$$

As Figure 8 shows, the selected method's similarity drops smoothly with the distance increases compared with the other algorithms. The risk level is calculated as

$$m = \underset{m}{\text{argmax}}\ \text{Similarity}(R, \text{GFN}_m). \tag{20}$$

4. Experiment

In this section, we discuss the results of the three experiments: (A) efficiency of algorithm RApriori, (B) efficiency of the classifier T-BiLSTM, and (C) the case study of the whole framework RAPOT.

4.1. Efficiency of RApriori. To validate the efficiency of RApriori, we compare it with the classical Apriori and FP-Growth. There are three subexperiments in this section: (a) time costs with different rule lengths, (b) time costs with different rule counts, and (c) time costs with different datasets. We carry on these experiments on the simulation datasets generated with the parameters shown in Table 3. In experiment (a), we employ Apriori, FP-Growth, and RApriori to work out rules with different lengths. Figure 10 shows that Apriori and FP-Growth's time costs sharply increase with more extended rules. In experiment (b), we compare the three methods for dealing with different counts of rules. Figure 11 illustrates our method's time cost grows slower than the other methods. In experiment (c), we conduct the three algorithms on three datasets with different data sizes. Figure 12 shows that our method has a better efficiency than Apriori and FP-Growth while tolerating data explosion.

4.2. Efficiency of T-BiLSTM. We train the classifier for public opinion sentiment analysis with the dataset contains 18000 positive comments and 18000 negative comments come from Weibo. The validating set has 3600 positive items and 3600 negative items. In addition, we compare the T-BiLSTM-based sentiment classifier with the KNN, maximum entropy, Bayes, SVM, and traditional BiLSTM. We adopt accuracy, positive-precision, positive-recall, and Macro-F1 as the evaluation metrics that are defined as

$$\begin{aligned}
\text{Accuracy} &= \frac{T}{N},\\
\text{Precision} &= \frac{TP}{TP + FP},\\
\text{Recall} &= \frac{TP}{TP + FN},\\
\text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},\\
\text{Macro-F1} &= \frac{1}{2}\left(\text{F1}_{\text{pos}} + \text{F1}_{\text{neg}}\right),
\end{aligned} \tag{21}$$

where T is the number of correct predictions, and N is the total number of valide samples. For TP and FP, they represent the amount of the predicted "Positive" samples which are correct and incorrect, respectively, which are similar to TN and FN. As for Macro-F1, it is defined as the average of F1_{pos} and F1_{neg} and is used to evaluate the efficiency of each classifer comprehensively. Table 4 shows the comparison result, and we can see that our T-BiLSTM exceeds the other methods.

4.3. Case Study of RAPOT. In this section, we evaluate the efficiency and applicability of RAPOT with a case study. It includes three sets of short texts corresponding to three cases; the size of the three sets are 764, 306, and 156. At first, the risk indicator model of RAPOT is shown as Figure 3. There are nine indicators in the aspects of the case, the related parties, and the judge. Then, we figure out the indicator values for each case, and the mapped linguistic terms are shown in Table 5. In the next step, the linguistic terms are turned into

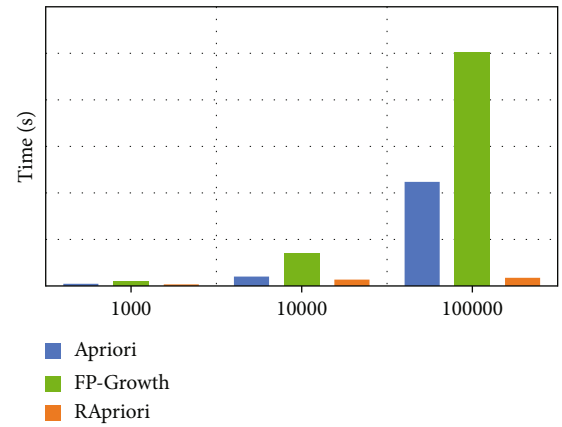


Figure 12: Time costs with different dataset.

Table 4: The comparison in accuracy of the five classifiers.

Classifier	Acc	Precision	Recall	Macro-F1
T-BiLSTM	0.88	0.90	0.85	0.88
BiLSTM	0.87	0.88	0.86	0.87
ME	0.81	0.82	0.79	0.81
Bayes	0.84	0.81	0.89	0.84
KNN	0.73	0.67	0.90	0.72
SVM	0.80	0.79	0.82	0.80

Table 5: Indicator evaluation for the three cases.

Indicators	Impact factors	Case 1	Case 2	Case 3
C1.1	VL	VL	FH	FH
C1.2	H	AL	AL	AL
C1.3	H	M	L	VL
C1.4	M	AL	AH	AH
C2.1	H	AH	AH	AL
C2.2	M	AH	AH	AL
C3.1	FH	AL	AH	AL
C3.2	M	AL	AL	AL
C3.3	L	AL	AL	AL

Table 6: The results of fuzzy similarities.

Risk level	Case 1	Case 2	Case 3
Absolutely low	0.4970	0.3506	0.5813
Very low	0.5229	0.3412	0.6386
Low	0.6917	0.4318	0.8592
Fairly low	0.9146	0.5872	0.7900
Medium	0.6866	0.7997	0.5561
Fairly high	0.4700	0.7347	0.3880
High	0.3895	0.5955	0.3246
Very high	0.3185	0.4770	0.2712
Absolutely high	0.3108	0.4881	0.2616

corresponding fuzzy numbers. Then, the impact factors and evaluations of the indicators are aggregated into a fuzzy number for each case. Finally, we compute the fuzzy number similarities to figure out the risk level.

Table 6lists the similarities. Therefore, the POR of case 1 is fairly low, the POR of case 2 is medium, and the POR of case 3 is low. Combined with Table 5, case 3 has the least heat. Meanwhile, the judge and the parties are not unique identities. Even though the case type is at high risk, without hot discussion, the POR is low. As for case 1, the public opinion is quite positive, so the risk assessment result is "Fairly-Low". Referring to case 2, one of the related parties has unique identities, and he has attracted much attention on social media. Nevertheless, media coverage is low, which illustrates that the issue has not been widespread yet. As we can see, the RAPOT recognizes the risk of POR successfully and distinguishes the three cases in risk measurement. To validate our framework's efficiency, we compare five similarity measure algorithms. As we can see in Figure 9, the selected method's output is the same as the majorities without outliner.

5. Conclusion

The accurate and fine-grained risk assessment on public opinion in the trial procedure is crucial for refined trial management. Our framework proposed in this paper provides an objective and efficient assessment for POR in the trial without using a large amount of historical data, which is quite lacking, and we propose T-BiLSTM to analyze public sentiment opinion based on topics. The method is more comprehensive than traditional BiLSTM in practice. The risk assessment framework for POR consists of three modules: (1) an adaptive multifactor indicator model for POR assessment, (2) the indicator evaluation module with an accurate public opinion analysis, and (3) the objective risk ranking module. The experimental results show the efficiency and practicability of our framework. In the future, we will work hard on the considerable amount of processing logs in the TPMS to further improve our indicator model's adaptation and robustness.

Data Availability

The dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the support of the National Key R&D Program of China under grant No. 2018YFC0830500.

References

[1] M. Bilal, H. Israr, M. Shahid, and A. Khan, "Sentiment classification of Roman-Urdu opinions using naive Bayesian, decision tree and KNN classification techniques," *Journal of King Saud University - Computer and Information Sciences archive*, vol. 28, no. 3, pp. 330–344, 2016.

[2] H. Htet, S. S. Khaing, and Y. M. Yi, "Tweets sentiment analysis for healthcare on big data processing and IoT architecture using maximum entropy classifier," in *International Conference on Big Data Analysis and Deep Learning Applications*, pp. 28–38, Singapore, 2019.

[3] J. Prusa, T. M. Khoshgoftaar, and D. J. Dittman, "Using ensemble learners to improve classifier performance on tweet sentiment data," in *2015 IEEE International Conference on Information Reuse and Integration*, pp. 252–257, San Francisco, CA, USA, 2015.

[4] S. Shubha and P. Suresh, "An efficient machine learning bayes sentiment classification method based on review comments," in *2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, pp. 1–6, Bangalore, India, 2017.

[5] H. Kim and Y. S. Jeong, "Sentiment classification using convolutional neural networks," *Applied Sciences*, vol. 9, no. 11, 2019.

[6] S. Lai, X. Liheng, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15*, pp. 2267–2273, Austin, Texas, USA, 2015.

[7] M. Yang, T. Wenting, J. Wang, X. Fei, and X. Chen, "Attention-based lstm for target-dependent sentiment classification," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17*, pp. 5013-5014, San Francisco, California, USA, 2017.

[8] H. Kanj and P. E. Abi-Char, "A new fuzzy-TOPSIS based risk decision making framework for dangerous good transportation," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Zhangjiajie, China, 2019.

[9] P. Singh, N. K. Mishra, M. Kumar, S. Saxena, and V. Singh, "Risk analysis of flood disaster based on similarity measures in picture fuzzy environment," *Afrika Matematika*, vol. 29, no. 7-8, pp. 1019–1038, 2018.

[10] R. Chutia and M. K. Gogoi, "Fuzzy risk analysis in poultry farming using a new similarity measure on generalized fuzzy numbers," *Computers & Industrial Engineering*, vol. 115, pp. 543–558, 2018.

[11] S. Kabir, C. Wagner, T. C. Havens, and D. T. Anderson, "A bidirectional subsethood based similarity measure for fuzzy sets," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–7, Rio de Janeiro, 2018.

[12] S. Cheng, S. Chen, and T. Lan, "A new similarity measure between intuitionistic fuzzy sets for pattern recognition based on the centroid points of transformed fuzzy numbers," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2244–2249, Hong Kong, China, 2015.

[13] Y. Zhang, X. Yin, and Z. Mao, "Study on risk assessment of pharmaceutical distribution supply chain with bipolar fuzzy information," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 2, pp. 2009–2017, 2019.

[14] A. Karasan, E. Ilbahar, S. Cebi, and C. Kahraman, "A new risk assessment approach: safety and critical effect analysis (SCEA) and its extension with pythagorean fuzzy sets," *Safety Science*, vol. 108, pp. 173–187, 2018.

[15] M. Zaky, "Risk analysis using fuzzy system based risk matrix methodology," *Arab Journal of Nuclear Sciences and Applications*, vol. 51, no. 4, pp. 204–212, 2018.

[16] G. F. Can and P. Toktas, "A novel fuzzy risk matrix based risk assessment approach," *Kybernetes*, vol. 47, no. 9, pp. 1721–1751, 2018.

[17] T. Luo, C. Wu, and L. Duan, "Fishbone diagram and risk matrix analysis method and its application in safety assessment of natural gas spherical tank," *Journal of Cleaner Production*, vol. 174, pp. 296–304, 2018.

[18] R. R. Yager, "Pythagorean membership grades in multicriteria decision making," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 958–965, 2014.

[19] H. Ni, A. Chen, and N. Chen, "Some extensions on risk matrix approach," *Safety Science*, vol. 48, no. 10, pp. 1269–1278, 2010.

[20] H. A. Khorshidi and S. Nikfalazar, "An improved similarity measure for generalized fuzzy numbers and its application to fuzzy risk analysis," *Applied Soft Computing*, vol. 52, pp. 478–486, 2017.

[21] J. Li and W. Zeng, "Fuzzy risk analysis based on the similarity measure of generalized trapezoidal fuzzy numbers," *Journal of Intelligent and Fuzzy Systems*, vol. 32, no. 3, pp. 1673–1683, 2017.

[22] Y. Qi, W. Jiang, and N. Liu, "Trial risk analysis based on a novel similarity measure on generalized fuzzy numbers," in *Proceedings of the 2020 4th International Conference on Management Engineering, Software Engineering and Service Sciences*, pp. 157–163, Wuhan, China, 2020.

[23] S. A. Ahmad, D. Mohamad, N. H. Sulaiman, J. M. Shariff, and K. Abdullah, "A distance and set theoretic-based similarity measure for generalized trapezoidal fuzzy numbers," in *AIP Conference Proceedings*, vol. 1974, pp. 020–043, AIP Publishing LLC, 2018.

[24] Z. Xu, S. Shang, W. Qian, and W. Shu, "A method for fuzzy risk analysis based on the new similarity of trapezoidal fuzzy numbers," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1920–1927, 2010.

[25] D. Yong, S. Wenkang, D. Feng, and L. Qi, "A new similarity measure of generalized fuzzy numbers and its application to pattern recognition," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 875–883, 2004.

[26] K. J. Schmucker, *Fuzzy Sets, Natural Language Computations, and Risk Analysis*, vol. 27, no. 3, 1984Computer Science Press, 1984.

[27] A. Darko, A. P. C. Chan, E. E. Ameyaw, E. K. Owusu, P Ã.¤r. Erika, and D. J. Edwards, "Review of application of analytic hierarchy process (AHP) in construction," *International Journal of Construction Management*, vol. 19, no. 5, pp. 436–452, 2019.

[28] G. Tian, M. Zhou, H. Zhang, and H. Jia, "An integrated AHP and VIKOR approach to evaluating green design alternatives," in *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*, pp. 1–6, Mexico City, Mexico, 2016.

[29] A. Farkas, "Multi-criteria comparison of bridge designs," *Acta Polytechnica Hungarica*, vol. 8, p. 173, 2011.

[30] S. A. K. Muhammad, A. Ali, S. Abdullah, F. Amin, and F. Hussain, "New extension of TOPSIS method based on pythagorean hesitant fuzzy sets with incomplete weight information," *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 5, pp. 5435–5448, 2018.

[31] N. L. H. Mo and Y. Qi, "Trial risk index model and assessment system based on extended TOPSIS method," in *2020 International Conference on Data Intelligence and Security*, South Padre Island, TX, USA, 2020.

[32] H. Tian and L. Wu, "Microblog emotional analysis based on TF-IWF weighted Word2vec model," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 893–896, Beijing, China, 2018.

[33] H. Bo, Y. Yang, A. Mahmood, and W. Hongjun, "Microblog topic detection based on LDA model and single-pass clustering," in *International Conference on Rough Sets and Current Trends in Computing*, pp. 166–171, Berlin, Heidelberg, 2012.

[34] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6630802, 10 pages
https://doi.org/10.1155/2021/6630802

Research Article

A Method of Surface Defect Detection of Irregular Industrial Products Based on Machine Vision

Mengkun Li,[1] **Junying Jia**,[2] **Xin Lu**,[2] **and Yue Zhang**[1]

[1]*School of Management, Capital Normal University, Beijing/100089, China*
[2]*Shenyang Fengchi Software Co. LTD, Shenyang/110167, China*

Correspondence should be addressed to Mengkun Li; limengkun@cnu.edu.cn and Xin Lu; luxin@fchsoft.com

Received 23 November 2020; Revised 8 December 2020; Accepted 9 April 2021; Published 10 May 2021

Academic Editor: Amr Tolba

In recent years, the surface defect detection technology of irregular industrial products based on machine vision has been widely used in various industrial scenarios. This paper takes Bluetooth headsets as an example, proposes a Bluetooth headset surface defect detection algorithm based on machine vision to quickly and accurately detect defects on the headset surface. After analyzing the surface characteristics and defect types of Bluetooth headsets, we proposed a surface scratch detection algorithm and a surface glue-overflowed detection algorithm. The result of the experiment shows that the detection algorithm can detect the surface defect of Bluetooth headsets fast as well as effectively, and the accuracy of defect recognition reaches 98%. The experiment verifies the correctness of the theory analysis and detection algorithm; therefore, the detection algorithm can be used in the recognition and detection of surface defect of Bluetooth headsets.

1. Introduction

In the mass production process of Bluetooth headsets, it is necessary to glue the connection of the Bluetooth headset. During the bonding process, the problem of glue overflow may occur. In addition, the surface of the headset is also prone to scratches during the processing process. Therefore, after the production of the Bluetooth headset, it is necessary to check whether the appearance of the headset is scratched and the glue-overflowed. The traditional manual detection efficiency is low, labor and time costs are high, the work is boring and tedious, and it is difficult for the human eye to recognize the relatively small scratches and glue-overflowed, and it is easy to make mistakes when working for a long time. In view of this, this article proposes a Bluetooth headset surface defect detection algorithm based on machine vision.

With the continuous development of machine vision technology, machine vision has been more and more widely used in surface defect detection of irregular industrial products. Jian et al. [1] proposed a detection algorithm to detect mobile phone screen glass. Zhou et al. [2] proposed a new surface defect detection framework. Meng et al. [3] identified the surface defects of the hose. Zhi et al. [4] designed an online automatic detection technology for wafer surface defects. Bo et al. [5] detected the surface defects of tiles. Li et al. [6] proposed a novel defect extraction and classification scheme for mobile phone screen based on machine vision. Zhu et al. [7] presented a machine vision based method for detecting surface defects of pipe joints. Le et al. [8] presented a novel framework based on machine vision known as the optical film defect detection and classification system for use in the real-time inspection. Wang et al. [9] designed a detection device for air bubbles, impurities, and other defects inside the flexible connection of aviation aircraft canopy. Yang et al. [10] proposed a method for wafer defect detection. Hu et al. [11] proposed an algorithm based on ellipse fitting and distance threshold to detect the pit defect of steel shell. Sun et al. [12] proposed a weld defect detection and classification algorithm based on machine vision to effectively identify and classify weld defects of thin-walled metal canisters. Wu et al. [13] examined a surface defect detection method based on support vector machine. Lib et al. [14] proposed a novel patterned method for fabric defect detection based on a novel texture descriptor and the low-rank decomposition

model. Li et al. [15] proposed a machine-vision-based defect detection method for packaging bags. Yu et al. [16] proposed a method fusing near infrared spectroscopy and machine vision to improve the accuracy in recognizing defects on wood surfaces. Yang et al. [17] proposed efficient approaches based on three-point circle fitting and convolutional neural network (CNN) to achieve automatic aperture detection. Han et al. [18] presented a fast machine-vision-based surface defect detection method using the weighted least-squares model. Hanzaei et al. [19] presented an automatic image processing system with high accuracy and time efficient approaches. Liu et al. [20] proposed a surface defect detection method based on gradient local binary pattern (GLBP), which uses image subblocks to reduce the dimensionality of the LBP data matrix. Mujeeb et al. [21] proposed an algorithm which detects surface level defects without relying on the availability of defect samples for training.

2. Design of the Detection Algorithm

The flow chart of the Bluetooth headset surface defect detection algorithm is shown in Figure 1. Firstly, we use an industrial camera to collect the headset image and then preprocess the input image. The preprocessing includes threshold segmentation and image enhancement. The purpose of threshold segmentation is to find the area that needs to be detected, and the image enhancement is to make the defects on the headsets surface more obvious. Next, perform surface scratch detection and surface glue-overflowed detection on the preprocessed image to confirm whether there are scratches and glue-overflowed on the headset surface, and finally, output the detection results. If no defects are found in the two detections, it is determined that there is no defect on the surface of the headsets.

2.1. Preprocess Image

2.1.1. Threshold Segmentation. After we use the industrial camera to collect the headset picture, the image needs to be preprocessed to find out the area to be detected, that is, the area that needs to detect scratches and glue overflow. The method of threshold segmentation is used to extract the region to be detected.

The basic principle of threshold segmentation is:

$$G(i,j) = \begin{cases} 0, f(i,j) < T \\ 255, f(i,j) > T \end{cases}. \quad (1)$$

$G(i,j)$ is the generated image after threshold segmentation, $f(i,j)$ is the input image, and T is the threshold used for segmentation value. The selection of threshold is very important for the detection in the image. Currently, the commonly used threshold selection methods mainly include fixed threshold method and automatic threshold method. The threshold in the fixed threshold segmentation is generally set manually, and the method is simple, but the threshold set the manual threshold set needs to be based on a large amount of experimental data, which is low in efficiency and cannot adapt to environmental changes. Automatic threshold segmentation automatically selects the segmentation threshold through image data statistics. There are mainly the maximum between-class variance segmentation method, the maximum entropy threshold segmentation method, and so on.

(1) Maximum Between-Class Variance Segmentation Method. The maximum between-class variance method was first proposed by a Japanese scholar Otsu Zhanzhi in 1979, so it is also called Otsu method. The basic idea is to divide the image grayscale histogram into two groups of background area and target area at a certain threshold according to the grayscale characteristics of the image. The variance between the two groups increases with the gray value difference between the background area and the target area. When the variance between the two groups is the largest, the threshold is determined and segmented [22].

(2) Maximum Entropy Threshold Segmentation Method. The maximum entropy threshold segmentation method is mainly based on the information entropy in the image, designing a reasonable entropy criterion and optimizing it, so that the threshold when the image entropy is maximum can accurately segment the target area and the background area in the image.

Use Eq. (1) to segment the image, and the result is shown in Figure 2. (a) is the input image; (b) is the threshold segmentation result.

2.1.2. Image Enhancement. There is a difference in the gray value between the defective area on the headset surface and the nondefective area, but the difference is not obvious enough. Increasing this difference through image enhancement algorithms can help to improve the accuracy of subsequent specific defect detection. The specific method is as follows: Firstly, we transform the input image from spatial domain image to frequency domain image by Fourier transform. The high frequency components of frequency domain image are usually defect edge and noise. The low-pass filter is used to filter the frequency domain image to remove the high-frequency components in the frequency domain, and then, the filtered frequency domain image is transformed into the spatial domain image through the inverse Fourier transform. Compared with the original image, the area with larger difference may be the glue-overflowed area, and the surface defect of the difference image is more obvious than the original image, and the image enhancement is realized.

The Fourier transform formula of picture $f(x,y)$ with image size of $M \times N$.

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi((ux/M)+(vy/N))}. \quad (2)$$

$u = 0, 1, 2, \cdots, M-1$, $v = 0, 1, 2, \cdots, N-1$, u, and v are the frequency domain variables and x and y are the spatial domain variables.

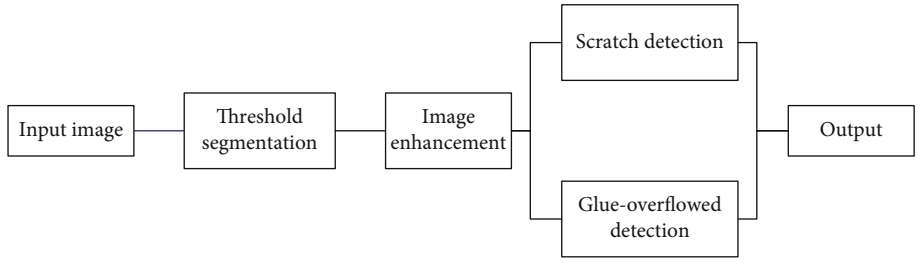


Figure 1: The flow chart of the surface defect detection algorithm.

(a) (b)

Figure 2: The result of threshold segmentation.

Corresponding inverse Fourier transform formula.

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi((ux/M)+(vy/N))}. \quad (3)$$

We get the frequency domain image after Fourier transform, then filter the frequency domain image. Commonly used methods of filtering include mean filtering, Gaussian filtering, median filtering, and so on.

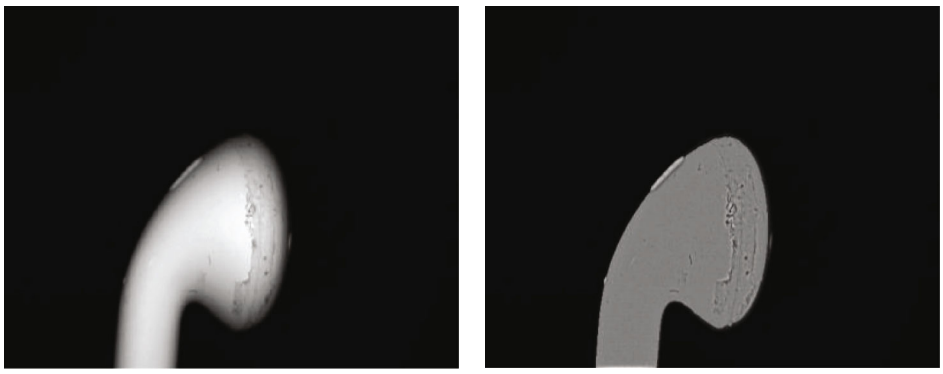
Mean filtering is a typical linear filtering algorithm. It refers to giving a template to the target pixel on the image. The template includes neigh boring pixels around it (8 pixels around the target pixel as the center to form a filter template). The average value of all pixels in the template replaces the original pixel value. Mean filtering uses a linear method to average the pixel values in the entire window range. Mean filtering has inherent defects. It cannot protect image details well. It also destroys the details of the image while denoising. The image becomes blurred, and the noise points cannot be removed well. Mean filtering is better for Gaussian noise.

Gaussian filtering is to scan each pixel in the image with a template, and use the weighted average gray value of the pixels in the neighbor-hood determined by the template to replace the value of the center pixel of the template. Gaussian filtering is the process of weighted averaging the input image. The value of each pixel is obtained by weighted averaging of itself and other pixel values in the neighbor-hood.

Median filtering is to take the points of adjacent pixels, and then sort the points of adjacent pixels, and take the gray value of the midpoint as the gray value of the pixel. The median filter uses a nonlinear method, which is very effective in smoothing impulse noise, and it can protect the sharp edges of the image, but it performs poorly against Gaussian noise.

Compared with mean filtering and median filtering, Gaussian filtering can keep more features of the overall gray distribution of the image when smoothing the image, so we choose Gaussian filtering in this process. Through Fourier transform, Gaussian filtering, and inverse Fourier transform, get the filtered image, after subtracting from the original image, the image enhanced image is obtained as shown in Figure 3(b). Compared to Figure 3(a) without any image processing, the defects on the headsets surface in Figure 3(b) are more obvious.

2.2. Scratch Detection. During the processing of the Bluetooth headset, scratches may occur on the surface, which affects the appearance of the headset. The scratches are linear, so the usual method in scratch detection is to detect the lines on the surface of the object. For example, some edge detection operators are used to detect the lines on the surface of the object. The more commonly used methods are Sobel operator, Canny operator, etc. The Sobel operator detection method has a better effect on image processing with gray gradient and more noise. When the accuracy requirements are not very high, it is a more commonly used edge detection method. Canny operator [23] is a detection operator proposed by computer scientist John F. Canny in 1986. It is currently the most comprehensive detection algorithm theoretically. Canny operator is not easy to be interfered by noise. Canny operator is a multistage optimization operator with



(a) (b)

Figure 3: The result of image enhancement.

filtering, enhancement, and detection. Before processing, Canny operator first uses Gaussian smoothing filter to smooth the image to remove noise. The operator uses the finite difference of the first-order partial derivative to calculate the gradient amplitude and direction. In the process, the Canny operator also goes through a process of nonmaximum suppression. Finally, the Canny operator also uses two thresholds to connect the edges. We use the canny operator to detect the scratches on the surface of the headset, and the results are shown in Figure 4(b).

From a morphological point of view, a scratch is composed of a series of adjacent pixels with a large difference in gray value from the background, usually a connected area. However, in the process of image preprocessing, the breakpoints with a small area divided by the shallow scratch may be filtered out, causing the scratch to break, which affects the result of line detection. At this time, performing a morphological expansion operation on the scratch to fill the scratch defect cavity can connect some adjacent fracture scratches, thereby solving the problem of scratch truncation. Then, through the skeleton extraction algorithm, a connected area is refined into the width of one pixel, and all the obtained skeleton subsets are combined to form the final scratch skeleton. The skeleton is constructed in a way that each point on it can be seen as the center point of a circle with the largest radius possible while still being completely contained in the region.

The result of skeleton extraction is as shown in Figure 4(c).

From the detection results in Figure 4, the performance of the skeleton extraction algorithm is better than the line detection algorithm.

2.3. Glue-Overflowed Detection

2.3.1. Watershed Algorithm. The watershed segmentation [24, 25] method is a mathematical morphology segmentation method based on topological theory. The basic idea is to regard the image as a topological topography in geodesy. The gray value of each pixel in the image indicates the altitude of the point. Each local minimum and its affected area are called collection basins, while the boundary of the collection basin forms a watershed.

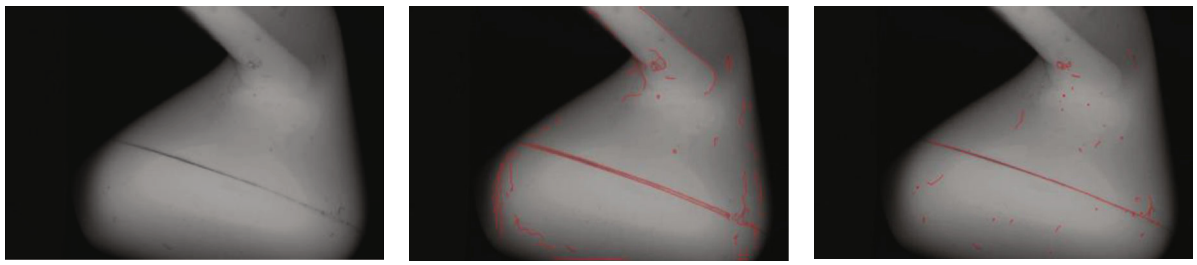
Firstly, we compute the watersheds without applying a threshold T, resulting in the same basins that would be obtained when calling watersheds. Secondly, the basins are successively merged if they are separated by a watershed that is smaller than threshold T. Let B_1 and B_2 be the minimum gray values of two neighboring basins and W the minimum gray value of the watershed that separates the two basins. The watershed is eliminated, and the two basins are merged if

$$\max\{W - B_1, W - B_2\} < T. \tag{4}$$

Through testing, we can get better image segmentation results by setting T to 5. Figure 5 shows the detection results of watershed detection algorithm.

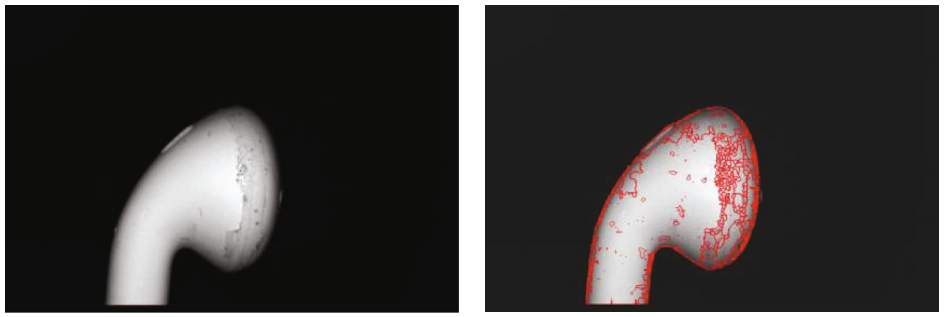
Watershed algorithm is an image region segmentation algorithm. In the process of segmentation, it will take the similarity with adjacent pixels as an important reference basis. Pixels with similar spatial positions and similar gray values are connected to each other to form a closed contour. Through the watershed algorithm, the headset is divided into several small areas according to the similarity of the gray values, which are used as the input of the subsequent gray level cooccurrence matrix.

2.3.2. Gray Level Cooccurrence Matrix. The gray level cooccurrence matrix (GLCM) was first proposed by Haralick et al. in 1973. [26]. GLCM describes the spatial relationship of gray levels in texture images and has been widely used in texture statistics and analysis. The texture is formed by alternating gray levels in spatial positions, so there is a certain spatial relationship between two pixels separated by a certain distance in the image. GLCM is a commonly used method to express the spatial correlation of pixel gray levels, mainly describing the image from the distance, direction, and degree of change between adjacent pixels. Its essence is to calculate the appearance frequency of two gray pixels under a certain spatial relationship, which can indicate the regional consistency and relativity of the image. GLCM changes quickly in fine textures with distance, while coarse textures change slowly. GLCM defines a square matrix whose size represents the probability of the gray value i from a fixed spatial position relationship (size and direction) to another gray value j.



(a) (b) (c)

Figure 4: The result of scratch detection.



(a) (b)

Figure 5: The result of watershed.

Suppose $f(x, y)$ is a 2D grayscale image, where S is a set of pixels with a certain spatial relationship in the area and P is GLCM, which can be expressed as

$$P(\mathrm{x}, y) = \frac{\#\{(x_1, y_1), (x_2, y_2) \in S \mid f(x_1, y_1) = i, f(x_2, y_2) = j\}}{\#S}. \tag{5}$$

$\#(X)$ represents the number of elements in the set X.

Applying GLCM to describe texture features is based on two-order statistical parameters as texture metrics. Haralick [26] proposed 14 feature statistics to describe image texture features. However, we usually use the following 4 statistical types: energy (ASM), contrast (CON), correlation (COR), and entropy (ENT). The arc second moment (ASM) reflects the regularity and uniformity of image distribution. Contrast (CON) reflects the depth and smoothness of the image texture structure. Correlation (COR) reflects the similarity of image texture in the horizontal or vertical direction. Entropy (ENT) is a measure of image information, reflecting the complexity of texture distribution. The 4 types of statistics are as follows:

$$\mathrm{ASM} = \sum_{i}^{N} \sum_{j}^{M} P(i, j)^2, \tag{6}$$

$$\mathrm{CON} = \sum_{i}^{N} \sum_{j}^{M} (i - j)^2 P(i, j), \tag{7}$$

$$\mathrm{COR} = \frac{\sum_{i}^{N} \sum_{j}^{M} (i - \bar{x})(j - \bar{y}) P(i, j)}{\sigma_x \sigma_y}, \tag{8}$$

$$\mathrm{ENT} = -\sum_{i}^{N} \sum_{j}^{M} P(i, j) \lg P(i, j). \tag{9}$$

We pick out 100 glue overflow areas and 100 normal areas from the small areas extracted by the watershed algorithm and count their energy (ASM), contrast (CON), and correlation (COR), respectively. The results are as follows:

In the gray level cooccurrence matrix, the energy value reflects the uniformity of the image gray level distribution and the texture thickness. If the element values of the gray level cooccurrence matrix are similar, the energy value is smaller, which means the texture is detailed, and the energy value is large, which indicates a more uniform and regular texture pattern. Therefore, the area without glue-overflowed is more regular and usually has a larger energy value, while the area with glue-overflowed has more texture and smaller energy value.

As shown in Figures 6 and 7, the ASM score of the glue-overflowed area is almost less than 0.2, while the ASM score of the area without glue-overflowed is almost greater than 0.3.

Therefore, set the ASM threshold T_{asm} to 0.2; $\mathrm{ASM}(i)$ represents the ASM value of the area i.

The area i that satisfies the following formula is judged to be a possible glue-overflowed area.

$$\mathrm{ASM}(i) < T_{\mathrm{asm}}. \tag{10}$$

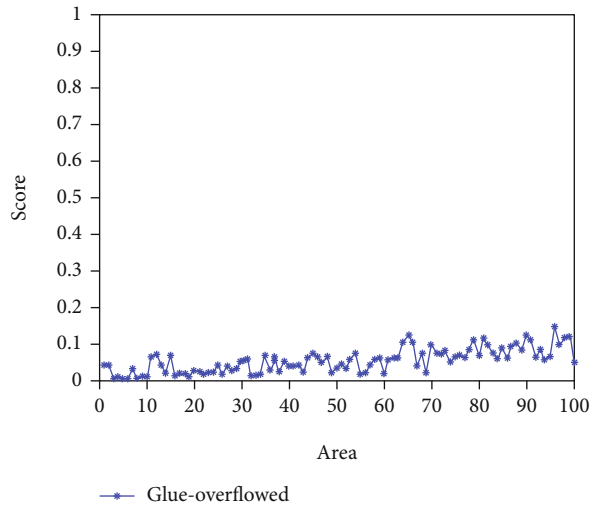


Figure 6: The result of ASM (glue-overflowed).

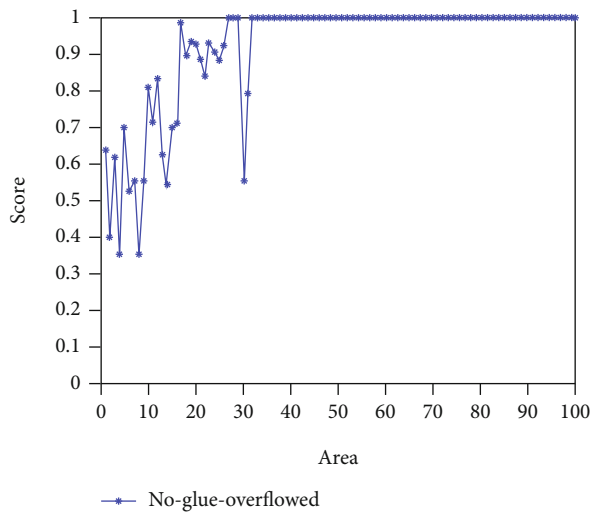


Figure 7: The result of ASM (no-glue-overflowed).

In the gray level cooccurrence matrix, the contrast (CON) reflects the sharpness of the image and the depth of texture. The deeper the texture, the greater the contrast.

As shown in Figures 8 and 9, the CON score of the area without glue-overflowed is almost less than 0.15, while the ASM score of the glue-overflowed area is almost greater than 0.2.

Therefore, set the CON threshold T_{con} to 0.2; CON(i) represents the CON value of the area i.

The area i that satisfies the following formula is judged to be a possible glue-overflowed area.

$$\mathrm{CON}(i) > T_{con}. \tag{11}$$

In the gray level cooccurrence matrix, the correlation (COR) reflects the local gray-scale correlation in the image. If there is texture distribution along a certain direction, the correlation value of GLCM is larger.

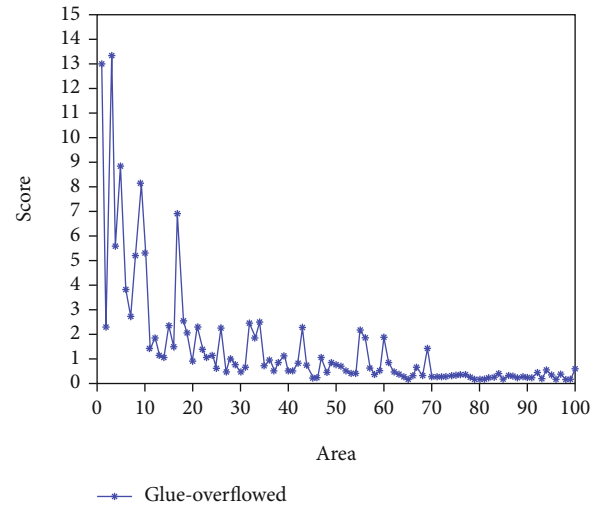


Figure 8: The result of CON (glue-overflowed).

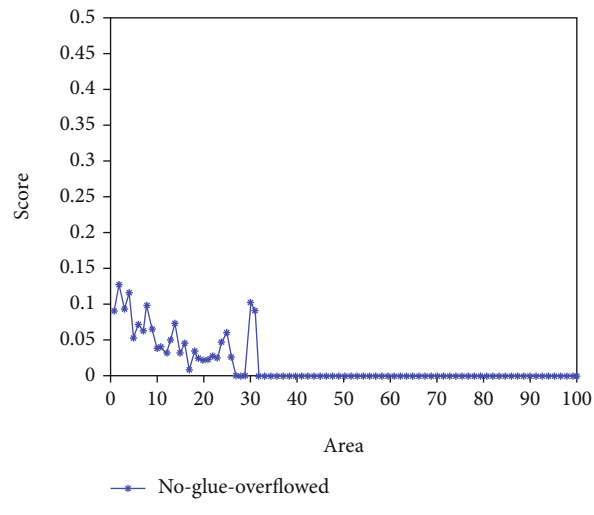


Figure 9: The result of CON (no-glue-overflowed).

As shown in Figures 10 and 11, the COR score of the glue-overflowed area is almost greater than 0.9, while the COR score of the area without glue-overflowed is almost less than 0.9.

Therefore, set the COR threshold T_{cor} to 0.9. COR(i) represents the COR value of the area i.

The area i that satisfies the following formula is judged to be a possible glue-overflowed area.

$$\mathrm{COR}(i) > T_{cor}. \tag{12}$$

If the area i satisfies Eq. (10), Eq. (11), and Eq. (12) at the same time, we judge the area i as the glue-overflowed area. Finally, the adjacent glue-overflowed areas are merged, and the areas that cannot be merged and are particularly small are removed. The final test results are shown in Figure 12. (a–d) are the detection results of glue-overflowed at different positions of the headsets. From the figure below, the glue-

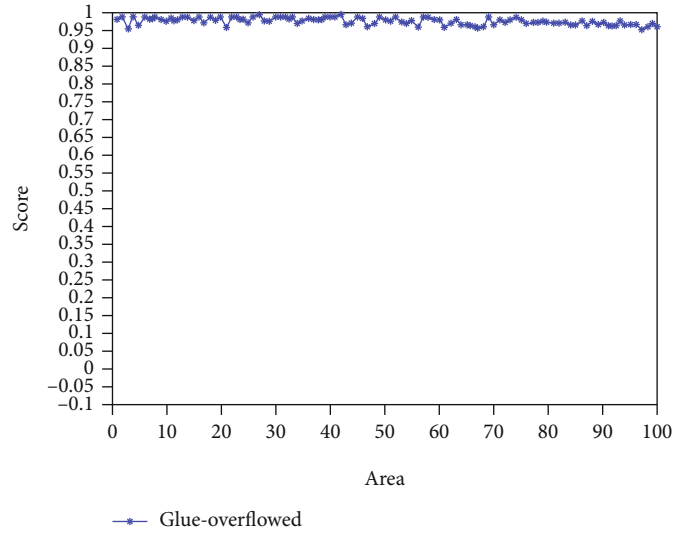


Figure 10: The result of COR (glue-overflowed).

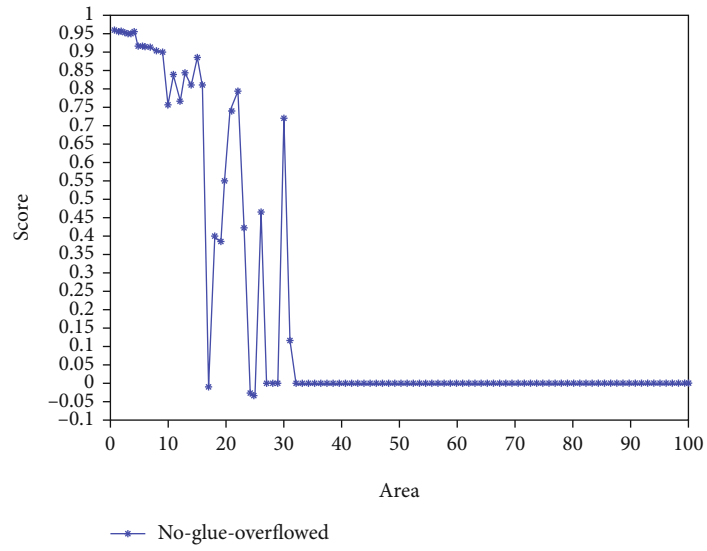


Figure 11: The result of COR (no-glue-overflowed).

overflowed detection effect based on the watershed algorithm, and the gray level cooccurrence matrix algorithm is good.

3. Experimental Results and Analysis

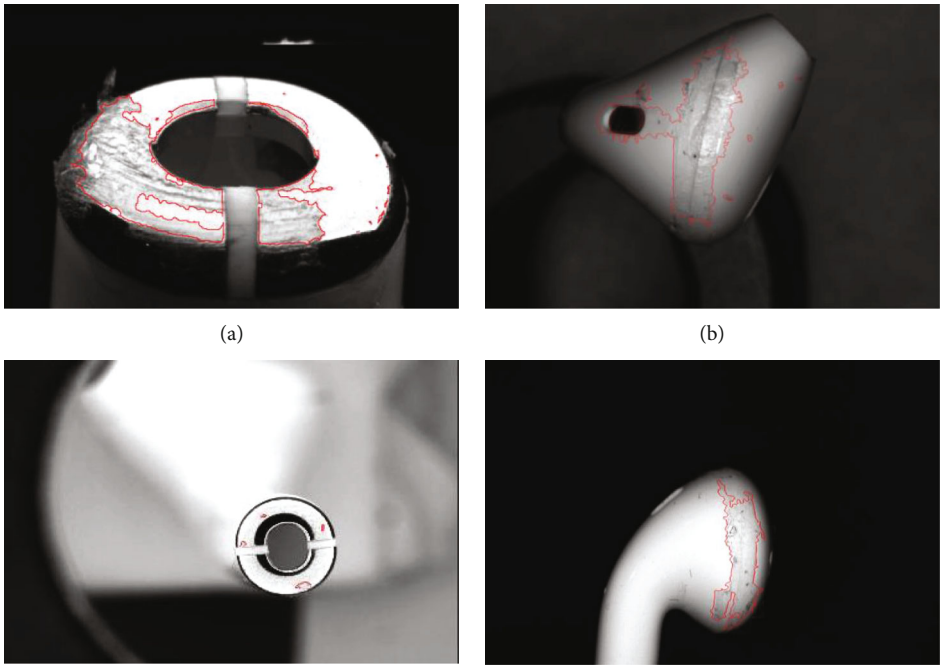
In order to verify whether the image enhancement algorithm improves the defect detection ability, we compare the image of the headset without image enhancement algorithm and image enhancement algorithm. As shown in Figure 13, (a) is the original picture of the headset, (b) is the picture after image enhancement, and (c) and (d) are the results of glue-overflowed detection on (a) and (b), respectively. It is proved from the figure that through the image enhancement algorithm, the glue-overflowed area in the image is more obvious, and the misdetection caused by the illumination is reduced. Therefore, the image enhancement algorithm proposed in this paper helps to improve the accuracy of the headset surface defect detection.

To further verify the proposed surface defect detection algorithm performance, we have defined the precision (Pr) and recall (Re) which are calculated as Eq. (13) and Eq. (14). The corresponding results are reported in Tables 1 and 2.

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{13}$$

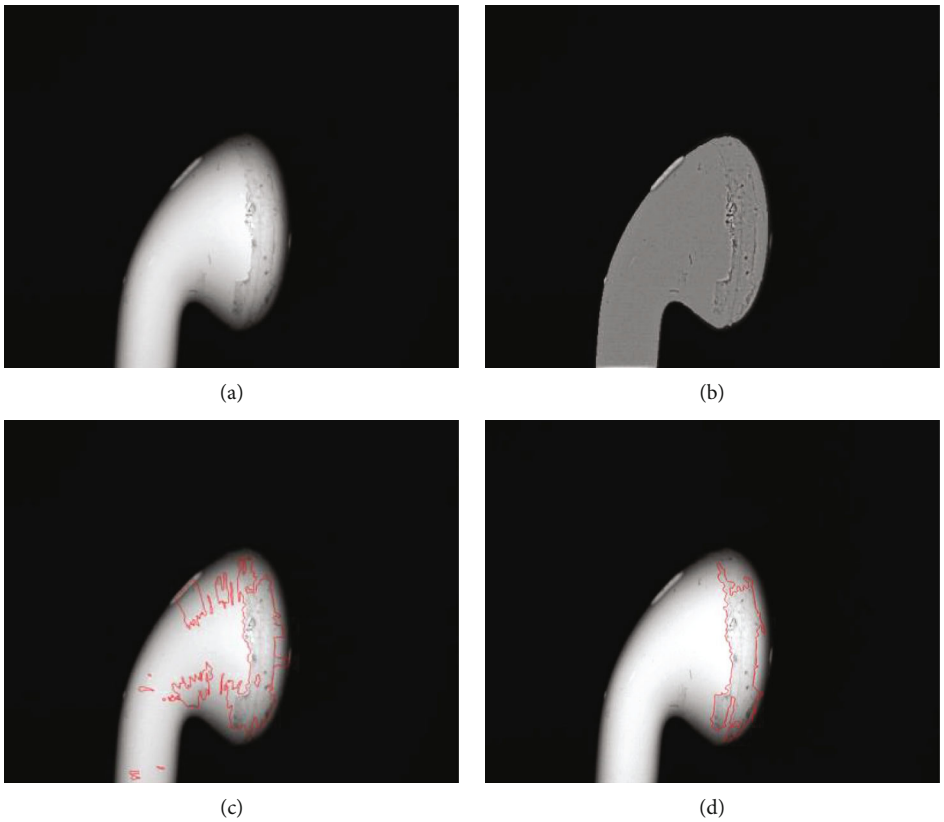
$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{14}$$

where *TP*, *FP*, and *FN* denote True Positive, False Positive, and False Negative which correspond to the count of actual defective products detected with defects, actual



(a) (b) (c) (d)

Figure 12: The result of glue-overflowed detection.



(a) (b) (c) (d)

Figure 13: The comparison of the image enhancement algorithm.

Table 1: Quantitative index for scratch detection.

Product quantity	2000	4000	6000	8000	10000
Precision (Pr)	97.80%	98.04%	97.88%	97.98%	98.06
Recall (Re)	100%	99.88%	99.83%	99.86%	99.90%

Table 2: Quantitative index for glue-overflowed detection.

Product quantity	2000	4000	6000	8000	10000
Precision (Pr)	98.28%	98.40%	98.12%	98.16%	98.28%
Recall (Re)	100%	100%	100%	100%	100%

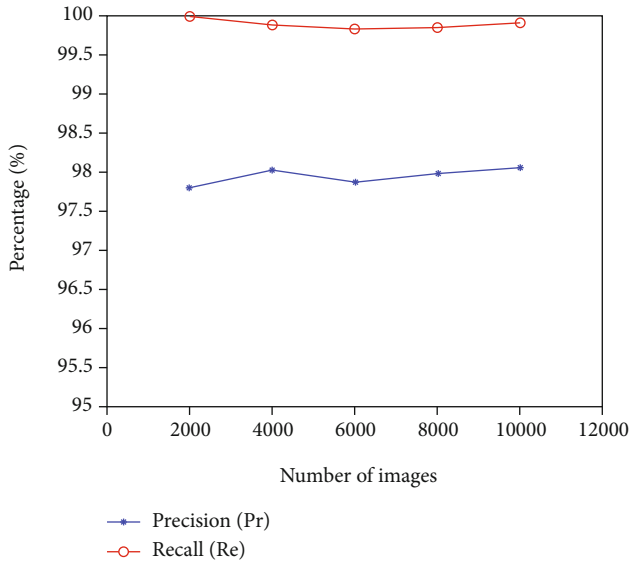


Figure 14: The quantitative indexes of scratch detection.

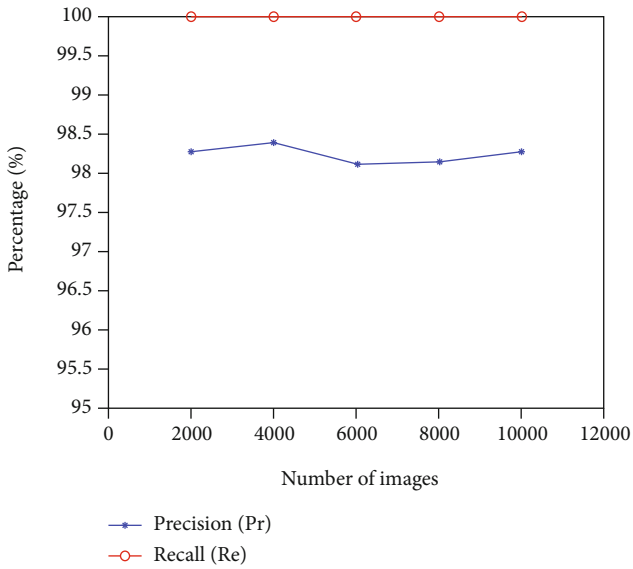


Figure 15: The quantitative indexes of glue-overflowed detection.

qualified products detected with defects, and actual defective products detected without defects, respectively.

We draw the precision and recall of the two detections into a line graph, as shown in Figures 14 and 15.

Through actual tests, the scratch detection precision is approximately 98%, and the recall is approximately 100%. The glue-overflowed detection precision is approximately 98%, and the recall is approximately 100%. In addition, the precision and recall of the detection are not affected by the number of images. Basically meet the testing requirements of industrial testing.

4. Conclusions

Taking the surface defect detection of Bluetooth headset as an example, this paper studies a surface defect detection algorithm for irregular industrial products based on machine vision, which can detect both surface scratches and glue-overflowed of irregular industrial products. At present, the Bluetooth headset surface detection algorithm has been applied in the actual industrial production; the practice proves that the detection algorithm proposed in this paper is effective. Meanwhile, it is prone to be applied to the surface defect detection of other products with similar surface features.

Data Availability

Data available on request. The data are available by contacting Mengkun Li (limengkun@cnu.edu.cn).

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was funded by the Major Technology Project of China National Machinery Industry Corporation (SINOMACH): "research and application of key technologies for industrial environment monitoring, early warning and intelligent vibration control (SINOMAST-ZDZX-2017-05)," and partially supported by the Scientific Research Foundation of Beijing Municipal Education Commission (KM201810028021).

References

[1] J. Chuanxia, G. Jian, and A. Yinhui, "Automatic surface defect detection for mobile phone screen glass based on machine vision," *Applied Soft Computing*, vol. 52, pp. 348–358, 2016.

[2] X. Zhou, Y. Wang, Q. Zhu et al., "Machine vision based automatic apparatus and method for surface defect detection," in *2018 13th World Congress on Intelligent Control and Automation(WCICA)*, pp. 1697–1702, Changsha, China, 2018.

[3] F. Meng, J. Ren, Q. Wang, and T. Zhang, "Rubber hose surface defect detection system based on machine vision," *IOP Conference Series: Earth and Environmental Science*, vol. 108, article 022057, 2018.

[4] C. Zhishan and L. Benyong, "Wafer surface defect detection based on machine vision," *Journal of Guizhou University(Natural Sciences)*, vol. 4, pp. 68–73, 2019.

[5] X. Bo and Z. Ping, "Research on tile surface defect detection based on machine vision," *Mechanical Engineering & Automation*, vol. 5, pp. 130–132, 2017.

[6] C. Li, X. Zhang, Y. Huang, C. Tang, and S. Fatikow, "A novel algorithm for defect extraction and classification of mobile phone screen based on machine vision," *Computers & Industrial Engineering*, vol. 146, article 106530, 2020.

[7] L. Jiang, K. Sun, F. Zhao, and X. Hao, "Machine vision based on pipe joint surface defect detection and identification," *Journal of Physics: Conference Series*, vol. 1621, no. 1, article 012012, 2020.

[8] N. T. Le, J.-W. Wang, M.-H. Shih, and C.-C. Wang, "Novel framework for optical film defect detection and classification," *IEEE Access*, vol. 8, pp. 60964–60978, 2020.

[9] P. Wang, X. Lu, J. Chen, and P. Zhang, "Flexible connection of cockpit canopy defect detection device based on machine vision," *IOP Conference Series: Materials Science and Engineering*, vol. 711, article 012007, 2020.

[10] J. Yang, Y. Xu, H. J. Rong, S. Du, and H. Zhang, "A method for wafer defect detection using spatial feature points guided affine iterative closest point algorithm," *IEEE Access*, vol. 8, pp. 79056–79068, 2020.

[11] H. Hu, D. Xu, X. Zheng, and B. Zhang, "Pit defect detection on steel shell end face based on machine vision," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2020.

[12] J. Sun, C. Li, X.-J. Wu, V. Palade, and W. Fang, "An effective method of weld defect detection and classification based on machine vision," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6322–6333, 2019.

[13] Y. Wu and Y. Lu, "An intelligent machine vision system for detecting surface defects on packing boxes based on support vector machine," *Measurement and Control*, vol. 52, no. 7-8, pp. 1102–1110, 2019.

[14] C. Li, G. Gao, Z. Liu, D. Huang, and J. Xi, "Defect detection for patterned fabric images based on ghog and low-rank decomposition," *IEEE Access*, vol. 7, no. 99, pp. 83962–83973, 2019.

[15] L. Dan, B. Guojun, J. Yuanyuan, and T. Yan, "Machine-vision based defect detection algorithm for packaging bags," *Laser & Optoelectronics Progress*, vol. 56, no. 9, article 091501, 2019.

[16] H. Yu, Y. Liang, H. Liang, and Y. Zhang, "Recognition of wood surface defects with near infrared spectroscopy and machine vision," *Journal of Forestry Research*, vol. 30, no. 6, pp. 2379–2386, 2019.

[17] Y. Yang, Y. Lou, M. Gao, and G. Ma, "An automatic aperture detection system for led cup based on machine vision," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23227–23244, 2018.

[18] Y. Han, Y. Wu, D. Cao, and P. Yun, "Defect detection on button surfaces with the weighted least-squares model," *Frontiers of Optoelectronics*, vol. 10, no. 2, pp. 151–159, 2017.

[19] S. H. Hanzaei, A. Afshar, and F. Barazandeh, "Automatic detection and classification of the ceramic tiles' surface defects," *Pattern Recognition*, vol. 66, pp. 174–189, 2017.

[20] X. Liu, F. Xue, and L. Teng, "Surface defect detection based on gradient LBP," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, Chongqing, China, 2018.

[21] A. Mujeeb, W. Dai, M. Erdt, and A. Sourin, "Unsupervised surface defect detection using deep autoencoders and data augmentation," in *2018 International Conference on Cyberworlds (CW)*, Singapore, 2018.

[22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems Man & Cybernetics*, vol. 9, no. 1, pp. 62–66, 2007.

[23] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[24] S. Avinash, K. Manjunath, and S. S. Kumar, "An improved image processing analysis for the detection of lung cancer using Gabor filters and watershed segmentation technique," in *International Conference on Inventive Computation Technologies*, Coimbatore, India, 2017.

[25] C. R. Jung and J. Scharcanski, "Robust watershed segmentation using wavelets," *Image and Vision Computing*, vol. 23, no. 7, pp. 661–669, 2005.

[26] Y. Jian, "Texture image segmentation based on Gaussian mixture models and gray level co-occurrence matrix," in *International Symposium on Information Science & Engineering IEEE*, Shanghai, China, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5517500, 16 pages
https://doi.org/10.1155/2021/5517500

Research Article

An Approach of Linear Regression-Based UAV GPS Spoofing Detection

Lianxiao Meng,[1,2] **Lin Yang**,[2] **Shuangyin Ren**,[2] **Gaigai Tang**,[2,3] **Long Zhang**,[2] **Feng Yang**,[2] **and Wu Yang**[1]

[1]*Information Security Research Center of Harbin Engineering University, Harbin, China*
[2]*National Key Laboratory of Science and Technology on Information System Security, Systems Engineering Institute, AMS, PLA, Beijing, China*
[3]*Harbin Engineering University, Harbin, China*

Correspondence should be addressed to Wu Yang; yangwu@hrbeu.edu.cn

Received 8 January 2021; Revised 23 February 2021; Accepted 1 April 2021; Published 7 May 2021

Academic Editor: Xiaojie Wang

A prominent security threat to unmanned aerial vehicle (UAV) is to capture it by GPS spoofing, in which the attacker manipulates the GPS signal of the UAV to capture it. This paper introduces an anti-spoofing model to mitigate the impact of GPS spoofing attack on UAV mission security. In this model, linear regression (LR) is used to predict and model the optimal route of UAV to its destination. On this basis, a countermeasure mechanism is proposed to reduce the impact of GPS spoofing attack. Confrontation is based on the progressive detection mechanism of the model. In order to better ensure the flight security of UAV, the model provides more than one detection scheme for spoofing signal to improve the sensitivity of UAV to deception signal detection. For better proving the proposed LR anti-spoofing model, a dynamic Stackelberg game is formulated to simulate the interaction between GPS spoofer and UAV. In particular, for GPS spoofer, it is worth mentioning that for the scenario that the UAV is cheated by GPS spoofing signal in the mission environment of the designated route is simulated in the experiment. In particular, UAV with the LR anti-spoofing model, as the leader in this game, dynamically adjusts its response strategy according to the deception's attack strategy when upon detection of GPS spoofer's attack. The simulation results show that the method can effectively enhance the ability of UAV to resist GPS spoofing without increasing the hardware cost of the UAV and is easy to implement. Furthermore, we also try to use long short-term memory (LSTM) network in the trajectory prediction module of the model. The experimental results show that the LR anti-spoofing model proposed is far better than that of LSTM in terms of prediction accuracy.

1. Introduction

With the progress of science and technology and the continuous reduction of manufacturing costs, UAV has entered the industrial production and people's daily life from the military field. Nowadays, UAV has been widely used in film and television shooting, agricultural monitoring, power inspection, personal aerial photography, meteorological monitoring, forest fire detection, traffic control, cargo transportation, and emergency rescue [1–3]. However, while UAV brings all kinds of convenience to our production and life, the security problems it faces are being gradually exposing.

At present, the common attacks on UAV mainly include the attacks on UAV sensors, UAV network, radio interference and hijacking, and GPS spoofing [4]. In these attacks, GPS spoofing is regarded as one of the most urgent threats, because it is practical and can be easily executed against UAV [5–7].

GPS spoofing refers to the following: in order to mislead the GPS navigation and positioning signal in the designated area, GPS attacker transmits pseudonavigation signal which cannot be effectively detected under the concealment condition because of its certain similarity with the real GPS signal, and user can get the false positioning, speed, and time

information from this type spoofing signal and finally be captured [8]. It should be pointed out that GPS spoofing is different from GPS jamming. GPS suppression jamming uses high-power jammer to transmit different types of suppression signals, which makes the target receiver unable to receive normal GPS signals, and users cannot obtain navigation, positioning, and timing results, which leads to the unavailability of GPS system [9]. GPS spoofing refers to the false signal to induce the GPS receiver to capture and track errors, so as to solve the wrong positioning, time, and speed information without being detected, achieving the purpose of cheating users. Because GPS spoofing often does not need strong transmitting power, it has good concealment and can guide related users to navigate in the wrong way to a certain extent, which also makes the deception have strong survivability. To some extent, the harm of spoofing jamming is more serious than that of suppressing jamming.

The vulnerability of GPS is the basis of GPS deception. The vulnerability of GPS mainly includes navigation signal format disclosure, navigation data format disclosure, and no protection for broadcast channel. In the current situation, GPS spoofing can be divided into three types [10, 11]: forwarding spoofing, generative spoofing, and track tracking spoofing. Detailed descriptions of the three spoofing are as follows, among them, the first two are the most commonly used and we choose the second type to solve in this paper.

(i) Forwarding spoofing: by recording the real GPS signal in the predeception positioning, forwarding spoofing uses the software to define radio and other signal transmitting equipment. Due to the fact that the structure of PN (pseudonoise) code cannot be changed and only the measurement value of pseudorange can be changed in the process of deception, the control flexibility is relatively poor, and the forwarding deception signal is easily detected. Therefore, the use of forwarding spoofing is often limited

(ii) Generative spoofing: it is to extract time, positioning, satellite ephemeris, and other necessary information from the real GPS signal, generate false GPS signal according to the predeception time and positioning information, and send it to the GPS receiver through the matrix antenna. This method does not require the current state of the receiver. It can cheat both the receiver in the acquisition state and the receiver in the steadytracking state [12]. Therefore, generative deception is often more practical

(iii) Track tracking spoofing: it mainly aims at the real-time flying air target [13]. Generally, the ground radar and other sensors detect the flight path of the aircraft in real time and send the detected air target positioning, speed, and other motion information to the deception equipment through the data link. Compared with the real signal, the deception signal produced by this method has higher fidelity and is not easy to be detected by other sensors such as inertial navigation system. Meanwhile, it has high requirements for the accuracy of the generated signal simulation, so it is difficult to be realized in practice

1.1. Problems to Be Solved. UAV is now in the critical period of transition from semi-intelligent to intelligent, and its main barrier is the degree of human intervention in flight tasks. Among them, fixed-point cruise only depends on preset information in flight, without any human operation, whether it can be completed safely is the first step to enter the era of UAV intelligence in the future. So in this paper, we choose the flight security of UAV in fixed-point cruise mission as the main research issue, that is, the UAV flies along the points selected in advance based on GPS positioning function. When the next selected flight location of the UAV is cheated by the fake GPS location, it is worth mentioning, the GPS spoofing here does not change the positioning of UAV, but rather changes its cognitive belief. By doing so, it is obvious that the UAV will betray its flight trajectory and fly in the direction of the connection between the deceiving location and the destination until it reaches the capture point [14], as showed in Figure 1. We expect to build an antispoof model, which can effectively prevent UAV from being trapped in such entrapment.

From the above analysis, it can be seen that the current GPS spoofing technology has a relatively clear technical implementation path. Therefore, it is necessary to put forward prevention strategies for the main deception technology in the current navigation system of UAV and other similar equipment.

1.2. Contributions. In view of the above problem, there are indeed many solutions, but they basically stop at detecting GPS spoofing, and there is no further action to ensure the mission going. Thus, the main contribution of this paper is to provide a general framework for UAV to reduce the impact of acquisition attack by detecting and defending GPS spoofing interference. Unlike previous work, our framework not only supports UAV detection of GPS spoofing attacks but also can guide UAV return to the previous flight path after detecting the attack and deviating from the route. This will enable the UAV to avoid being captured and complete its mission. In summary, our contributions are threefold:

(i) An LR anti-spoofing model for UAV is proposed in this paper. The flight trajectory prediction model of UAV is built by fitting the flight log of UAV with LR model, and the prediction accuracy is relatively high among all the methods. The model not only realizes the safety detection of UAV flight status in the process of mission but also realizes the deception mitigation when UAV is cheated, so as to ensure the smooth completion of flight mission

(ii) In order to meet the experimental needs, we built a GPS deception generator and realized the reappearance of deception scene when we analyzed the GPS deception problem faced by UAV

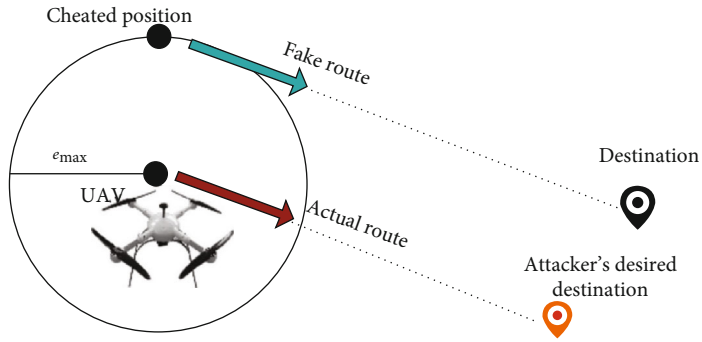


Figure 1: UAV actual and fake route.

(iii) In order to prove the effectiveness of the proposed LR anti-spoofing model, we design a Stackelberg attack and defense game consisting of GPS spoofer and UAV with LR anti-spoofing model. In this game, for the dynamic change of spoofing signal, it strongly proves that our algorithm can still achieve effective detection and resistance

The rest of this paper is organized as follows. Section 2 mainly introduces the current situation of GPS spoofing detection scheme. The UAV's LR anti-spoofing model is presented in Section 3. Section 4 describes the Stackelberg game scenarios in detail. In Section 5, we give details of our experimental setting, results, and corresponding analysis. The conclusions and future works are discussed in Section 6.

2. Related Work

In the world, there are frequent incidents of GPS positioning and navigation [15]. The most serious incident in the field of UAV security is Iran's capture of RQ-170 military UAV of the United States in 2011 [16]. In June 2012, Humphreys' research team of Texas State University successfully demonstrated in a track and field that the GPS spoofing device with hardware cost less than $1000 can change the flight path of a small UAV in real time by releasing deceptive jamming signals. Later, the team successfully demonstrated at the white sand missile range of the United States. In addition, in 2013, the team successfully used GPS deception technology to induce an $80 million white rose yacht to deviate 3° to the left, causing it to deviate 1 km from the scheduled route [17].

Nowadays, there are several protection methods for GPS spoofing at home and abroad, as follows:

(1) Signal physical layer characteristic detection method: GPS false signals are identified by comparing the characteristics of false signals and real signals in the signal physical layer. These differences mainly include automatic gain control [18], signal arrival direction, carrier phase value, and Doppler frequency shift [19]. Psiaki and others [20] [21] analyzed the principle of GPS deception detection based on the direction of arrival of signals. The angle of arrival of signals was determined by the change of signal carrier phase between different antennas, so as to judge whether the current target was attacked by GPS spoofing, and proposed a deception detection scheme based on the arrival direction of GPS signal. Ranganathan and others [22] proposed a deception detection method called auxiliary peak tracking, which can be used in combination with navigation message checker to track the strongest satellite signal and other weak environmental signals. Kang et al. [23] proposed a method to estimate the difference between the direction of arrival (DOA) and the measured DOA using GPS ephemeris and ephemeris data and used GPS directional antenna to detect deception

(2) Verification detection method based on cryptography: after receiving the signal, the receiver needs to decode the signal and authenticate the sender of the signal. Wesson et al. [24] proposed a probability model GPS signal authentication method based on statistical hypothesis test, which combines cryptographic source authentication with code timing authentication [25] and detects GPS spoofing attacks by using pseudorandom noise code of GPS signals

(3) Using other equipment to assist positioning detection method, through the use of inertial navigation, wireless network and cellular network, and other auxiliary means combined with GPS receiver to achieve the purpose of antideception, Panice et al. [6] proposed an anti-GPS spoofing detection mechanism based on state distribution combined with inertial navigation system and detected GPS spoofing attack by analyzing the error distribution between GPS and inertial navigation by using support vector machine. Magiera and Katulski [26] proposed a GPS deception detection and mitigation technology based on phase delay and spatial processing, which uses multiple receiving antennas to estimate the signal phase delay and spatial filter the signal to protect the GPS receiver from deception attack. Jansen et al. [27] proposed a group crowdsourcing method to detect the GPS spoofing attack of UAV. The method uses multiple aircraft to report the positioning difference and detects the GPS spoofing attack of UAV positioning through wireless air traffic control system. Kwon

and Shim [28] proposed a method to detect GPS spoofing attack by comparing the acceleration difference between GPS receiver and accelerometer

In the scenario of UAV flying along the designated route, such as power inspection and logistics distribution, the existing schemes still have the following problems:

(1) The method based on the physical layer detection of GPS signal can only detect simple GPS spoofing. When the attacker uses multidirectional GPS deception devices to transmit false GPS signals or dynamically adjust the frequency and power of GPS signals at the same time, the deception attack cannot be detected only by the physical layer characteristics of GPS signals. Therefore, this method cannot solve the problem of UAV trajectory deviation caused by the abovementioned GPS deception interference in power inspection

(2) The verification method based on cryptography cannot solve the replay attack of signal, and the encryption of signal is not suitable for civil GPS signal

(3) Using other equipment-aided positioning detection methods can improve the anti-spoofing ability of GPS receiver to a certain extent, but it will increase the cost of equipment positioning and the load of UAV in power inspection

Moreover, the focus of these schemes is mainly on the technology of detecting attack. A UAV is attacked in the process of moving towards a specific destination. The best it can do is to identify the attack and stop using the changed GPS signal. There is no other attack mitigation or defense mechanism to ensure the UAV to fly to the designated destination safely.

3. LR anti-spoofing Model

In LR (linear regression) anti-spoofing model proposed, UAV trajectory prediction is an important part, and LR is the final selected trajectory prediction method.

3.1. Linear Regression Analysis. Regression analysis is a statistical method that deals with the dependence between variables. It is one of the most widely used methods in mathematical statistics. Least squares regression analysis is the most typical linear regression algorithm [12, 29]. Regression analysis is based on the observation data to establish a quantitative relationship between two or more variables to analyze the inherent laws of the data. According to the number of independent variables, it can be divided into univariate regression analysis and multiple regression analysis; according to the relationship between independent variables and dependent variables, it can be divided into linear regression analysis and nonlinear regression analysis. Regression analysis is a predictive modeling technology, which is often used in predictive analysis. For example, the equipment frequency measurement method based on regression analysis is more accurate than other methods, and it is easier to realize; using regression analysis method to analyze the main factors affecting road traffic accidents can effectively prevent traffic accidents and improve road traffic efficiency.

In this paper, the univariate linear regression analysis method is used to establish a UAV positioning interval with the change of time stamp to predict the UAV trajectory in the mission. The method of univariate linear regression analysis is as follows.

According to the characteristics of the research object, the appropriate dependent variable and independent variable are selected. If the sample data shows that the two are in line with the linear relationship, then the univariate linear regression model is established:

$$y = a + bx + \varepsilon, \tag{1}$$

where y is the dependent variable, a is the constant term, b is the regression coefficient, x is the independent variable, and ε is the random error term, which reflects the influence of random factors on y except the linear relationship between x and y.

Assuming that the random error term ε in the regression model is a random variable with an expected value of 0 ($E(\varepsilon) = 0$) and it obeys normal distribution, then for a given x value, the expected value of y is

$$E(y) = a + bx. \tag{2}$$

The population regression parameters a and b are unknown and need to be estimated with sample data. For a selected sample, the regression parameters a and b in the model are replaced by sample statistics $\hat{a}$ and $\hat{b}$, and the estimated regression equation in linear regression is obtained, the sample regression equation

$$\hat{y} = \hat{a} + \hat{b}x \tag{3}$$

where $\hat{y}$ is the estimation of the mean value of dependent b variable y, $\hat{a}$ is the constant term of sample regression b equation, and $\hat{b}$ is the sample regression coefficient. For a dataset with sample size N, the values of $\hat{a}$ and $\hat{b}$ estimated by the least square method are

$$\hat{a} = \bar{y} - \hat{b}\bar{x}, \tag{4}$$

$$\hat{b} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}, \tag{5}$$

where $\bar{x}$ and $\bar{y}$ are the average values of sample data x_i and y_i, respectively. After $\hat{a}$ and $\hat{b}$ are obtained, the linear regression equation of univariate can be obtained.

3.2. LR anti-spoofing Model Parameter. In our prediction model, the physical meaning of the parameters is as follows: x is the deviation of time stamp, and y is the deviation of longitude/latitude. To keep synchronization, the period of

deviation calculation is set to the same value of the frequency at which UAV receives GPS signals. Moreover, the data used in the deviation calculation is taken from the UAV under the condition of stable flight. Finally, the mapping relationship between x and y is established, and then, two linear regression models for latitude and longitude prediction are formed, respectively.

3.3. Workflow of LR anti-spoofing Model. Based on the LR model proposed in this paper, the flight trajectory prediction value of UAV and the positioning value of GPS receiver of UAV are fused at the decision level, which can quickly detect the GPS spoofing of UAV. The workflow chart of LR anti-spoofing model proposed is shown in Figure 2.

We will carry out single-step deception detection and multistep deception detection in the model. The difference lies in the discrepancy between the predicted value of single step or multistep with the current GPS positioning data to determine the status of UAV being cheated by GPS. If the deviation of longitude and latitude is less than the corresponding security threshold, it is determined that no GPS spoofing is detected; if the difference of either longitude or latitude is greater than the corresponding E (security threshold), it is determined that the target UAV has been spoofed. The predicted positioning data is used as the current positioning information to guide the UAV to fly. More details of single- and multistep predictions are as follows.

(i) Single-step detection: for each time interval, we first input the correction value, time stamp, and current GPS time stamp of the previous time to the linear regression trajectory prediction model, which outputs the positioning information of the predicted current time

(ii) Multistep detection: for each time interval, we first input the correction value, time stamp, and current GPS time stamp of the last m times to the linear regression trajectory prediction model, which outputs the positioning information of the predicted current time

The reason why we introduce multistep detection is that the deception signal is set in a reasonable error range in order to improve its credibility. In particular, we set up a sliding window to store the correction data of m histories for multistep prediction. For each multistep prediction, the corrected data at the previous m times is compared with the predicted data to detect deception. After that, this data is eliminated and the data in the window is pushed forward one step. Finally, the correction data of this time is saved in the $m-1$ positioning. The physical meanings of parameters in Figure 2 are shown in Table 1.

3.4. E (Noise Threshold) Setting. Given a group of UAV's continuous historical trajectory of normal fixed-point cruise, $T = \{(t_1, \mathrm{lat}_1, \mathrm{lon}_1), \cdots, (t_i, \mathrm{lat}_i, \mathrm{lon}_i)\}$, $i = 1, 2, \cdots, N$. Each track point is represented by a tuple, which contains three elements: time stamp, latitude, and longitude. Then, we can extract the deviation of longitude and latitude in the range of two adjacent time stamps:

$$\begin{aligned} \delta_{\mathrm{lat}} &= \mathrm{lat}_i - \mathrm{lat}_{i-1}, \\ \delta_{\mathrm{lon}} &= \mathrm{lon}_i - \mathrm{lon}_{i-1}. \end{aligned} \tag{6}$$

The 1.5 times of the maximum value of δ_{lon} is taken as the deviation threshold E of longitude, and the latitude takes the same setting. This setting is due to the consideration of physical environment interference in actual flight.

4. Attack Defense Game

For approaching the real scene as much as possible, we take quadrotor UAV as the research object and design an attack defense game based on Stackelberg leader-follower game theory [14, 30, 31] between the simulated GPS dynamic deception signal generator and the UAV with our LR anti-spoofing model.

4.1. Stackelberg Leader-Follower Game. The concept of leader-follower game was first proposed by Heinrich von Stackelberg, a German economist, in 1934. In the Stackelberg leader-follower game, after the leader makes the decision, the follower makes the optimal response to the leader's decision, and finally, the leader makes the most favorable decision according to the follower's decision. Principal subordinate game belongs to the category of asymmetric game, the positioning of participants in the game is unequal, and the strategy choice of followers depends on the strategy choice of leaders. This idea is consistent with our LR defense model and GPS dynamic deception signal generator positioning in UAV mission.

4.2. Stackelberg Game Scenario. In the attack defense game, UAV with our LR anti-spoofing model is the leader, named LR defender, and the simulated GPS dynamic deception signal generator is the follower, named GPS spoofer. In the planning game, each player will choose a strategy and take actions to control the positioning of UAV in each time step. In this way, both players can observe the initial positioning of the drones and their subsequent positions to the current time step. In addition, the game is based on the assumption of complete information, that is, both players have the complete information of their opponents. Our work includes three game rounds, seven steps.

(1) LR defender: receive a two-point fixed voyage mission

(2) GPS spoofer: according to the current positioning and the expected deception positioning of UAV, a deception trajectory (a group of GPS trajectory data) is calculated, and a deception signal is sent every 200 ms

(3) LR defender: the deviation between the current predicted trajectory point and GPS real-time positioning data is calculated every 200 ms. If the deviation is greater than the safety threshold of the prediction

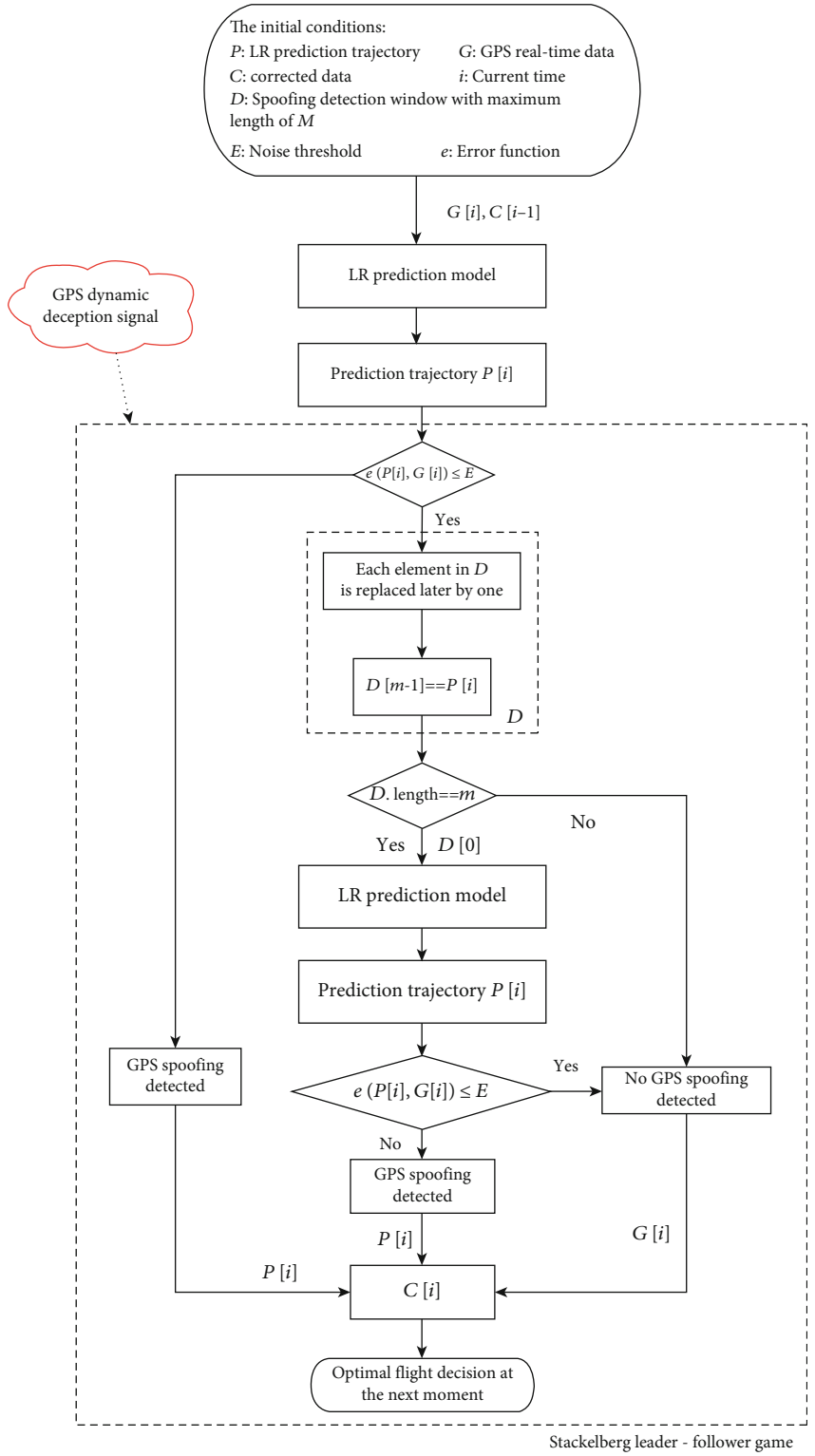


Figure 2: Workflow chart of LR anti-spoofing model.

module, the GPS real-time data will be removed at the next time, and the LR predicted trajectory points will be used to guide the UAV to complete the flight mission; if the deviation is less than the safety threshold, the GPS real-time data will continue to be received for trajectory positioning

(4) GPS spoofer: if the flight trajectory of UAV is not in accordance with the expected deception trajectory, the trajectory point information of GPS deception trajectory is adjusted until the data deviation between each two trajectory points is less than the safety threshold of UAV prediction module

Table 1: Physical meaning of parameters in workflow chart of LR anti-spoofing model.

Parameters	Physical meaning
P: LR predicts trajectory points	The continuous mission track points of UAV predicted by LR prediction model through historical track.
G: GPS real-time data	At present, the UAV airborne sensors receive the real signal from the mission environment.
C: corrected data	The value ($P[i]/G[i]$) transmitted to the UAV navigation system is selected according to the judgment of whether the current UAV mission environment is safe (whether there is GPS deception signal).
E: noise threshold	From the analysis of flight experience, the reasonable path error of UAV in a safe and normal mission environment due to its own attitude control and physical environment is obtained.
D: spoofing detection window with maximum length of M	The model provides two detection means. The window is set for further detection of deception, recording the trajectory values of the UAV at five adjacent moments (M is set to 5 in the invention).
e: error function	The variables involved in the calculation are LR predicted value and GPS real-time data at the current time. Compared with E, the results are used to judge whether the current UAV mission environment is safe or not.

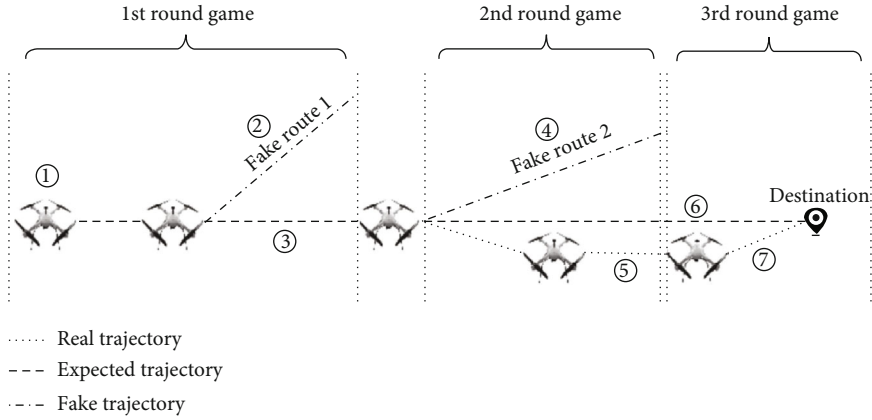


Figure 3: The expected motion state of UAV in Stackelberg game.

(5) LR defender: the deviation between the current predicted track point and GPS real-time positioning data is calculated at each time. If the deviation at five consecutive times is less than the safety threshold of the prediction module, the deviation between the track point data at the fifth time predicted by LR and the current GPS real-time data is calculated. If it is still less than the safety threshold, the flight will continue according to the GPS real-time data. If it is greater than the safety threshold, the GPS spoofer will be removed at the next time. The real-time data is used to guide the UAV flight with LR predicted trajectory points

(6) GPS spoofer: observe the flight trajectory of UAV at several times, and give up the deception if it does not follow the expected deception trajectory

(7) LR defender: after receiving the predicted trajectory values, we synchronously calculate the longitude/latitude variation, Δ, of GPS signals received at adjacent times (e.g., n time and $n+1$ time). Because the generation of deception signal is based on constant deviation, the longitude/latitude variation of adjacent time is also fixed. When

$$\Delta_{(n+1)-n} \neq \Delta_{n-(n-1)}, \tag{7}$$

it can be determined that there is no GPS spoofing at present. Then, the UAV stops using the predicted value and starts to use the GPS signal currently received to locate and continue to complete the task.

In this game, the expected motion state of UAV is shown in Figure 3.

5. Simulation and Evaluation

Our aim is to evaluate the performances of LR anti-spoofing model. To be specific, the experiment is mainly carried out from the following aspects.

5.1. Experiment Setting. The experiment is based on the UAV Simulation Platform consisted of jMAVSim and QGroundControl. jMAVSim is a simple and lightweight multirotor simulator. It connects directly to the hardware-in-the-loop (HITL, via serial) or software-in-the-loop (SITL, via UDP) instance of the autopilot. QGroundControl is simulation ground control station. It provides full flight control and mission planning for any MAVLink-enabled UAV and collects flight logs. The flight log contains the data collected by various sensors and some system output data during the flight. We extract GPS-related data (time stamp, longitude, and latitude) from flight log to consist the training dataset. In the experiments, the training dataset of LR anti-spoofing model is generated by a preset fixed-point cruise flight mission in the UAV simulation environment. The relevant parameters of the dataset are as follows in Table 2.

5.2. Deception Scenario Validation. Before verifying our proposed LR anti-spoofing model, we first verify the effectiveness of our deception scenario.

5.2.1. Deception Scenario Construction. In order to better simulate the real situation, we build a simulation deception scene which depends on a simulated GPS dynamic deception signal generator designed by us. We expect to realize the decoy capture of UAV in this scene. In this scenario, the deception means is to dynamically generate a group of trajectory point signals to deceive the UAV by observing the track changes of the target aircraft after entering the stable flight state. The deception trajectory setting is based on the noise range of UAV GPS itself. The specific implementation details are as follows.

In the beginning, we can calculate the noise threshold of GPS data of UAV in normal flight through the intermediate interpolation method. Based on this background, a group of deceptive trajectories is generated randomly. In order to capture the target more quickly, the GPS change value of two adjacent moments is greater than the upper limit of noise threshold. When the UAV does not fly according to the expected deception trajectory, it is speculated that the UAV may have certain detection and filtering ability for the signals with large changes. Based on the purpose of acquisition, a new deception trajectory is generated according to the error threshold so that the GPS change value of the two adjacent moments is within the noise threshold range, and the credibility of the deception signal is improved. The simplest way is to add a fixed increment to the GPS deception data at the next time.

5.2.2. Validity Verification. Figure 4 shows the trajectory diagram of UAV completing a given flight mission in the environment without any interference and deception is based on the ground coordinate system, and the abscissa and ordinate represent the latitude information and longitude information, respectively. Figure 5 is the visual expression of mission route in QGC. As you can see, H represents the home point of the UAV, and 1 represents the destination.

Table 2: Collected dataset parameters.

Parameters	Value
Signal frequency	20 Hz
Total number of tracks	40000
Total length of mission	1243.31 m
Threshold-latitude	$63 * 10e\text{-}6$
Threshold-longitude	$133.8 * 10e\text{-}6$

Figure 6 shows the experimental results of the target that is affected by the GPS deception signal we send in the UAV mission environment.

At the beginning, GPS spoofer did not send deception signals. From the ground control station, we can observe that the target aircraft was flying normally along the established route during this period. Therefore, we can also see from the chart that the trend of the blue line is a smooth and regular straight line. After flying for a period of time, we started the GPS simulation deception signal generator designed by us, GPS spoofer, to send GPS deception signal to the target's mission environment. Point A in Figure 6 represents the beginning time of deception. According to the principle of GPS deception signal mentioned in Section 1, the route of target plane after being spoofed by GPS spoofer is determined by deception signal and target point. We can see from Figure 6 that the trend of the blue line changes with the change of the red line after point A, which indicates that the target has indeed accepted the GPS deception signal, changed its belief in its positioning, and thus changed its movement state.

It is a fact that the target aircraft periodically returns to adjust the trajectory: in the fixed-point cruise mission, the UAV does not always take the current positioning and destination positioning as the optimal trajectory planning, but sets a local prediction point within a certain distance based on the given route so that the UAV will fly to the destination first after a certain distance from the route. The next point is predicted near this prediction point and the flight path is planned. Finally, Figure 6 shows that the target plane flies almost perpendicularly to the established route. What is worse, with the accumulation of time there is no tendency that the target UAV fly to the mission destination, and it is in a state of complete and serious yaw. This phenomenon can also be intuitively seen on the ground control station of the simulation platform, as shown in Figure 7. This proves that our GPS simulation deception signal generator and deception scene can effectively realize the deception acquisition of UAV.

5.3. Validation of LR anti-spoofing Model. In the constructed simulated deception scenario, we put on a Stackelberg game to verify the effectiveness of LR anti-spoofing model. According to Section 4, the GPS spoofer dynamically adjusts the deception signal according to the flight state of the target plane and plays a game with the UAV with LR anti-spoofing model.

Figure 8 shows the experimental results of our LR anti-spoofing model deployed on UAV.

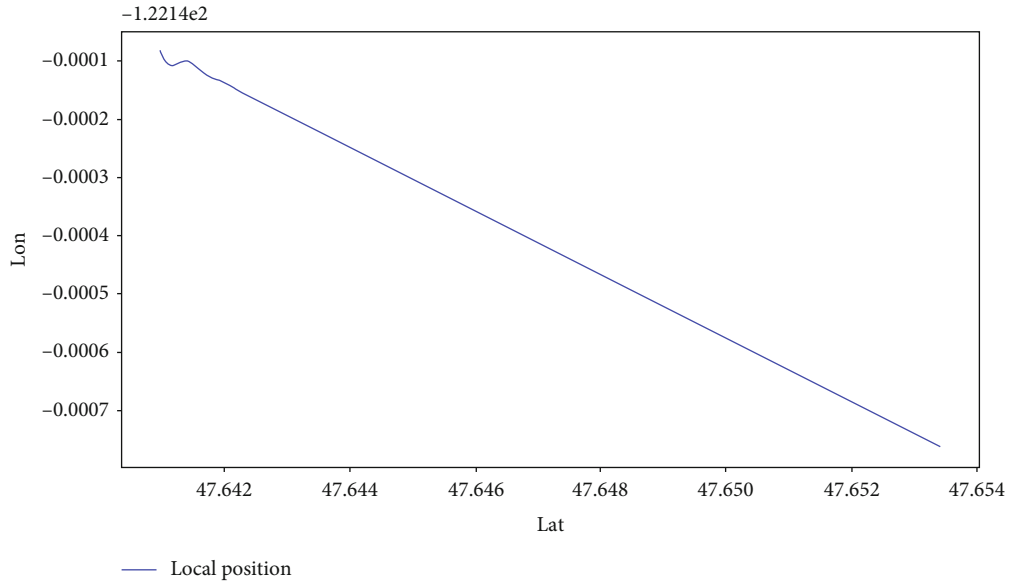


Figure 4: The trajectory diagram of a given flight mission is based on the ground coordinate system, and the abscissa and ordinate represent the latitude information and longitude information, respectively.

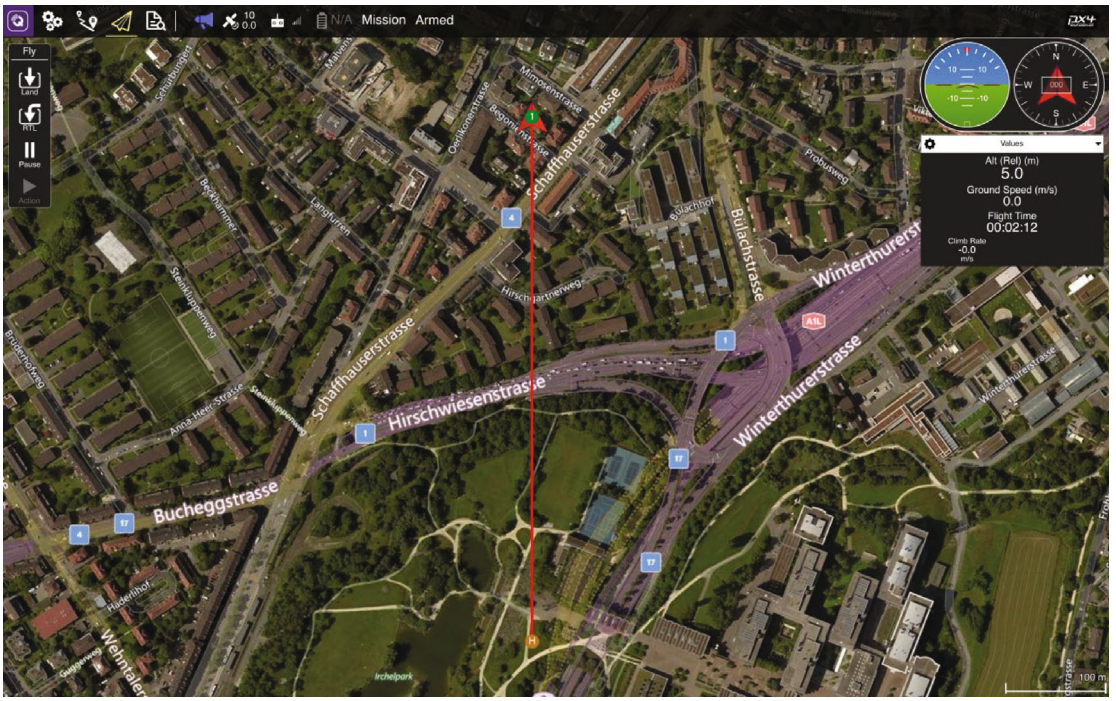


Figure 5: The visual expression of mission route in QGC.

We can see from Figure 8 the following:

(1) LR defender: the UAV enters a stable flight state after taking off for a period of time

(2) GPS spoofer: after observing the UAV in a stable flight state, GPS spoofer starts to send deception signals in the mission environment. In order to capture the target UAV as soon as possible, the deception signal is set outside the current positioning of the target which is greater than E in the AB segment

(3) LR defender: due to the deployment of our LR anti-spoofing model, the target plane will directly detect the step source and abandon it and follow the prediction module in LR anti-spoofing model to continue to move. As can be seen from the AB segment, the target UAV is in normal flight state

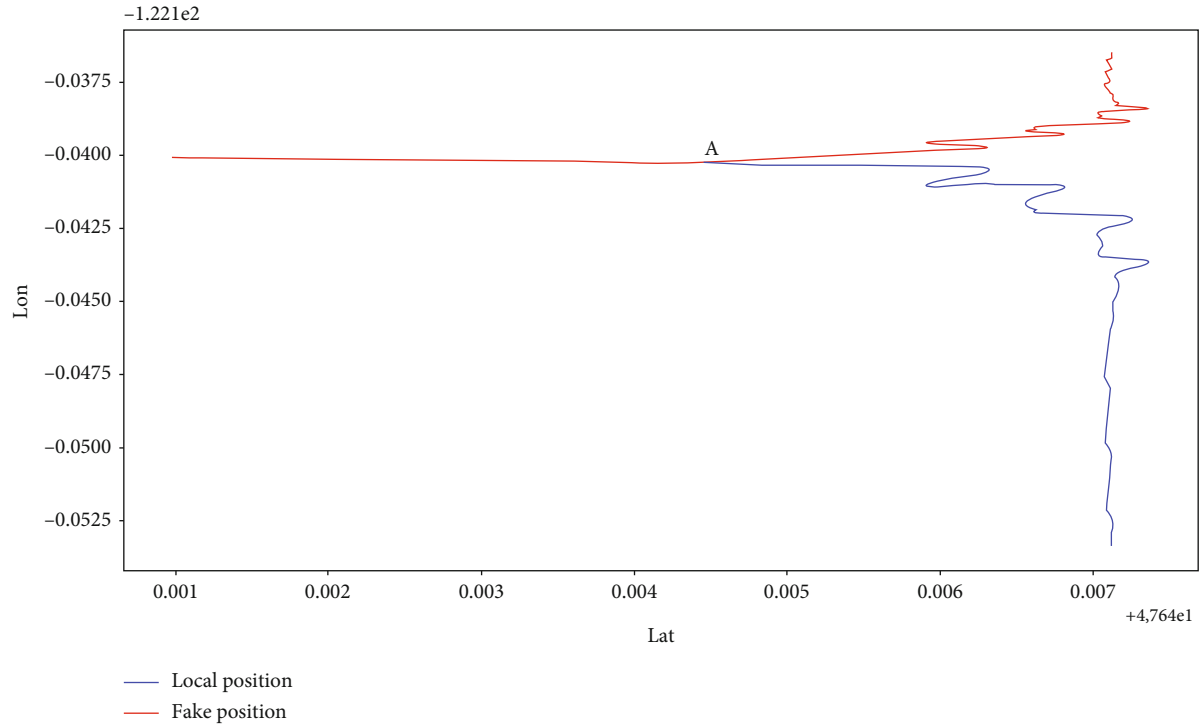


Figure 6: The red line indicates the dynamic change track of GPS deception signal in the process of trapping. The blue line indicates the real positioning of the target aircraft during the mission, that is, the real flight path.

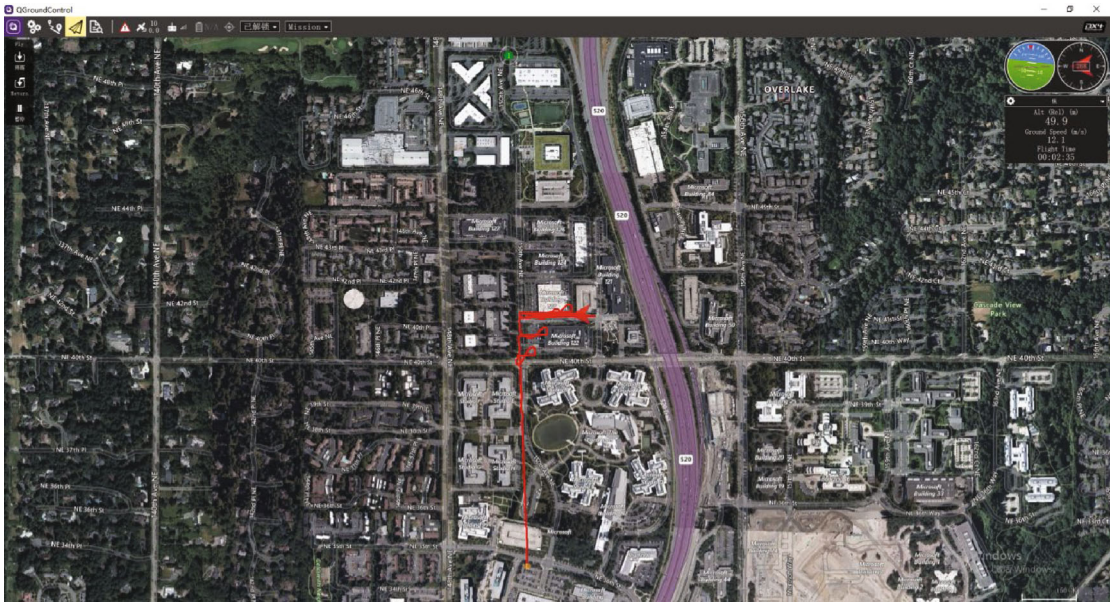


Figure 7: The complete yaw trajectory of UAV can be seen directly in QGC.

(4) GPS spoofer: after observing that the target UAV is not affected by the deception signal, GPS spoofer adjusts the deception signal to make it change within the range of E. At this time, we can see that in the BC segment, the target UAV has received the deception signal and has replanned its flight route. The spoofing is successful and effective in this period of time

(5) LR defender: it is worth mentioning that, in order to better ensure the safe flight of UAV, our detection mechanism, LR anti-spoofing model, is a two-step reinforcement type. At moment C, the target UAV detects the adjusted deception signal through the multistep detection mechanism in LR anti-spoofing model, starts to output the predicted value in time

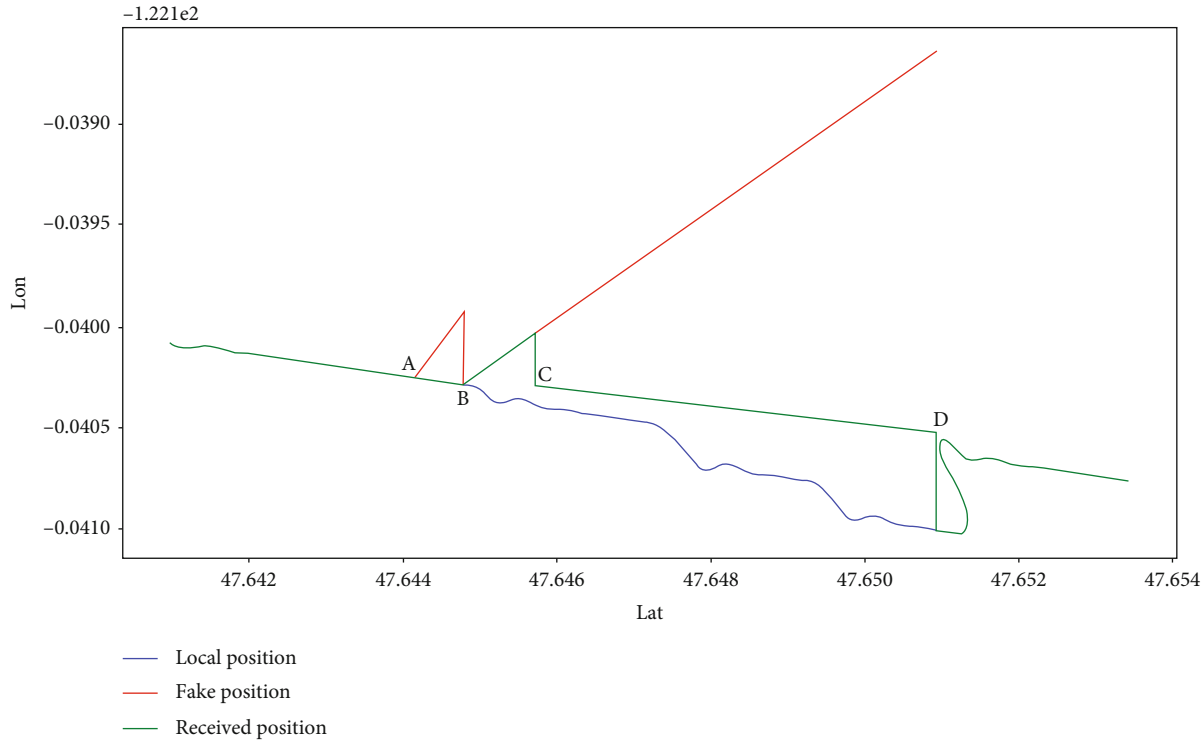


Figure 8: The red line indicates the dynamic change track of GPS deception signal in the process of trapping. The green line represents where the UAV thinks it is. The blue line indicates the real positioning of the target aircraft during the mission, that is, the real flight path.

Figure 9: Trajectory correction of UAV deployment LR anti-spoofing model (1).

window D to the UAV, and makes a self-adjustment according to the fact mentioned above. In the CD segment, the target receives the predicted trajectory value, which is equivalent to a self-deception for the UAV that has deviated from the course. From the first half of CD, we can see that the motion state of the target UAV is consistent with the deception principle mentioned in Section 1. The drift of the second half is due to the small cumulative deviation between the predicted route and the established route; the deviation has been verified by engineering and is within a reasonable range

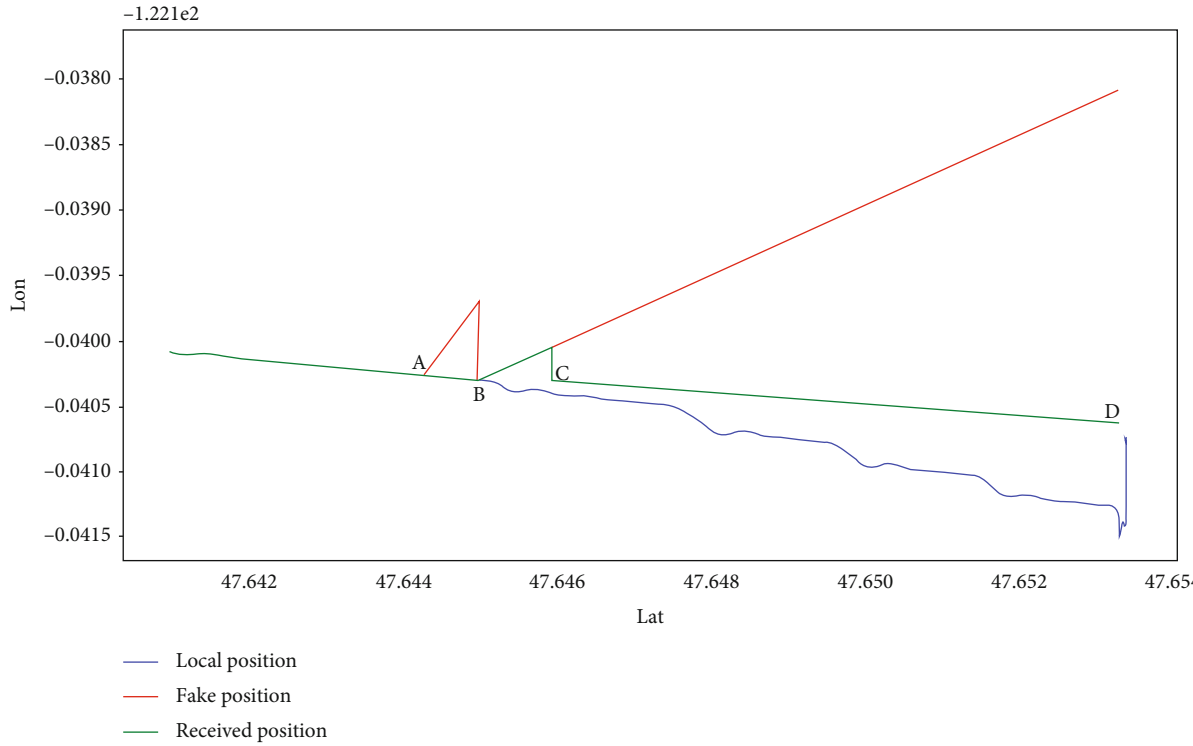


Figure 10: The red line indicates the dynamic change track of GPS deception signal in the process of trapping. The green line represents where the UAV thinks it is. The blue line indicates the real positioning of the target aircraft during the mission, that is, the real flight path.

Figure 11: Trajectory correction of UAV deployment LR anti-spoofing model (2).

(6) GPS spoofer: gave up spoofing at D moment

(7) LR defender: after moment D, target UAV to receive the real GPS signal when it detects that there is no deception interference in the mission environment and then finds that it has a certain degree of yaw; it will automatically adjust to the route and then continue to complete the task along the established route

Figure 9 is a QGC visual chart of the UAV that successfully resisted deception, completed the flight mission, and arrived at the established destination safely. Due to the scale

problem, the performance of flight status in the chart is not obvious, but some track fluctuations can still be seen.

In order to further verify the effect of our LR anti-spoofing model, we also designed an experiment while the GPS spoofer did not give up cheating in the whole process. The result is shown in Figure 10.

From Figure 10, we can see that under the guidance of our predicted value, although the UAV has a fixed deviation from the established routes, resulting in the UAV having a small stage yaw, the target UAV has still finally completed the flight mission. When the UAV receives the predicted value and thinks that it will arrive at the destination, the deviation from the real destination is only 72.35 m, which is within the visual range of the real destination. The total length of the mission route is 1243.31 m. This experiment also proves that the LR anti-spoofing model proposed is effective.

It can also be seen from QGC (Figure 11) that there is no uncontrollable yaw phenomenon in the whole course of UAV.

5.4. Comparison of Different Methods' Performance for UAV Trajectory Prediction Performance. The trajectory prediction module in our anti-spoofing model plays the role of navigation after the target UAV is affected by deception signal, so we expect the prediction accuracy to be as high as possible. On the same dataset, in addition to linear regression, we also try to use neural network, LSTM, in the selection of trajectory prediction module [32]. The result is inferior to the current linear regression.

5.4.1. Description and Processing of Experimental Data. The relevant parameters of the dataset are as follows in Table 3.

5.4.2. Evaluation Metrics. In order to determine the performance of the LSTM-KF defense model, the root mean square error is used to evaluate the fitting performance of the model.

Root mean square error (RMSE) is the relationship between the data sequence and the real value, which is the square root of the average of the sum of squares of the distances that each data deviates from its true value.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=2}^{n}\left(X_{\mathrm{obs},i} - X_{\mathrm{model},i}\right)^2}. \tag{8}$$

5.4.3. Trajectory Prediction of UAV Based on LSTM. The LSTM [33, 34] model has strong ability to predict time series data, which is the main reason why we choose it for trajectory prediction [35]. The LSTM model is trained. The historical trajectory characteristic data of UAV is taken as input, and the future UAV trajectory characteristic data is taken as the corresponding label. By training LSTM recurrent neural network, the mapping relationship between UAV historical flight trajectory and UAV future flight trajectory is established to realize the prediction of UAV future flight trajectory.

Let $x(t)$ be the triple data of UAV at each time, where t represents the time of UAV flight, and the information represented by triple data is $[\mathrm{lon}_t, \mathrm{lat}_t, v_t]$, where lon, lat, and v are longitude, dimension, and velocity of UAV at time t, respectively. Then, the trajectory characteristic $x(t)$ of UAV at time t can be expressed as

$$x(t) = \{\mathrm{lon}_t, \mathrm{lat}_t, v_t\}. \tag{9}$$

Table 3: Collected dataset parameters.

Parameters	Value
Signal frequency	5 Hz
Pseudocode type	C/A
Total number of tracks	30000
Training set : test set	4 : 1

After training, the flight trajectory of UAV can be predicted by using the trained LSTM model. The UAV flight trajectory data $[x_{t-n+1}, \cdots, x_t]$ of n consecutive moments are taken as the input data of LSTM model, and the prediction n steps backward, that is, the UAV trajectory data $[x_{t+1}, \cdots, x_{t+n}]$ at the future n moments is taken as the output, where n is the step size of input layer in the LSTM model. Therefore, the expression of UAV flight trajectory prediction model is

$$\{x_{t+1}, \cdots, x_{t+n}\} = f(\{x_{t-n+1}, \cdots, x_t\}). \tag{10}$$

For LSTM, the main parameters that affect its performance are the input step size and the number of neuron nodes. Through experiments, we choose the optimal parameters for LSTM.

From Table 4, we can see that when the number of neurons is 8, the prediction accuracy is relatively low. With the increase of the number of neurons, the prediction error decreases significantly. When the number of neurons is 16, the overall prediction error is the smallest, showing the best prediction accuracy, so we set the number of neurons as 16. From Table 5, we can clearly see from the results that with the increase of input step size from 5 to 10, the prediction error of the model gradually decreases. When the input step size is 12, the prediction error of the model increases greatly, which shows poor prediction performance. This may be because the input step size is too large, which leads to the overfitting phenomenon and the degradation of generalization performance. Therefore, we set the input step size of the LSTM prediction model to 10.

5.5. Comparison and Evaluation. It has been mentioned many times in this paper that trajectory prediction module is an important part of LR anti-spoofing model. Figure 12 mainly shows the performance of LR and LSTM in trajectory prediction, respectively, blue represents the established waypoint, and green and red represent the predicted waypoint of LR and LSTM separately. It is obvious in this figure that the fitting ability of LR-based prediction model is better than that of LSTM-based prediction model, taking the given route as the criterion. This is because, for the trajectory prediction problem of UAV two-point cruise mission, there is a linear relationship between the longitude and latitude change and the time change of UAV positioning. The target value

Table 4: Comparison of RMSE corresponding to neuron node.

Step_in	Step_out	Neurons	$e * 10^{-3}$	e_lat $* 10^{-5}$	e_lon $* 10^{-5}$	vel $* 10^{-3}$
10	5	8	68.114	5.600	25.750	5.943
10	5	16	1.923	0.035	0.160	0.755
10	5	32	5.135	0.449	2.060	0.321
10	5	48	4.555	0.368	1.696	0.426
10	5	64	9.472	0.847	3.890	0.499

e is the RMSE of all predicted data and real data, e_lat is the RMSE of latitude, e_lon is the RMSE of longitude, and vel is the RMSE of speed.

Table 5: Comparison of RMSE corresponding to input timing steps.

Step_in	Step_out	Neurons	$e * 10^{-3}$	e_lat $* 10^{-5}$	e_lon $* 10^{-5}$	vel $* 10^{-3}$
5	5	16	2.636	0.076	0.360	0.896
8	5	16	2.063	0.076	0.360	0.594
10	5	16	1.923	0.035	0.160	0.755
12	5	16	61.393	5.092	23.410	5.142

e is the RMSE of all predicted data and real data, e_lat is the RMSE of latitude, e_lon is the RMSE of longitude, and vel is the RMSE of speed.

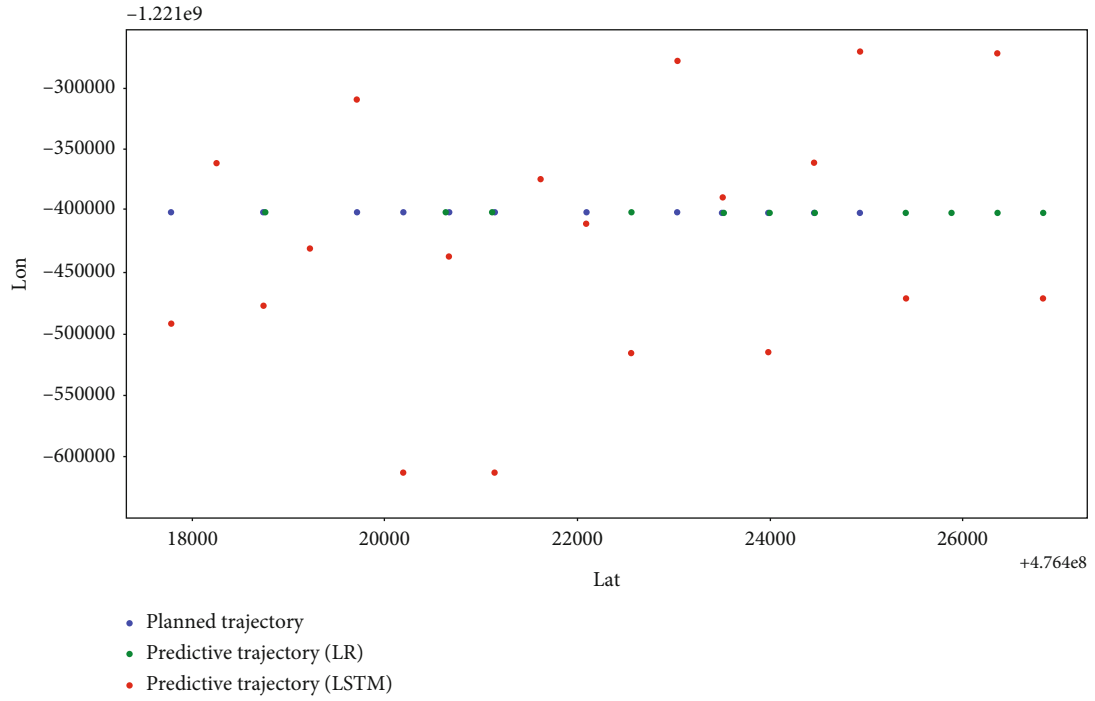


Figure 12: Performance comparison of trajectory prediction based on LSTM and LR in secure mission environment.

expectation of the LR model is a linear combination of input variables, and the model is simple and easy to model, so it is very suitable to solve this problem. Thus, the LSTM neural network is suitable for solving nonlinear problems; it has a big disadvantage in solving linear problems which is its own uncertainty, for the same input will produce different output, so the single use of LSTM is not suitable for the problem we want to solve.

6. Conclusion

Spoofing is one of the most important threats to GPS receivers. This paper discusses the detection model of UAV anti-GPS spoofing and proposes the LR anti-spoofing model. The flight trajectory prediction model of UAV is obtained by fitting the flight log of UAV with LR model, and the prediction accuracy is relatively high among all the methods. The

model not only realizes the safety detection of UAV flight status in the process of mission but also uses the decision fusion of sensor information to accurately detect the deception signal, so as to achieve the purpose of anti-spoofing interference. At the same time, when the UAV is cheated, it can also achieve deception mitigation, so as to ensure the smooth completion of the flight mission. Compared with the traditional anti-spoofing detection method or that based on neural network, this method not only has the characteristics of high accuracy and no need to increase the hardware cost of auxiliary equipment but also has fast linear regression modeling speed and does not require high computing ability of small computing board carried by UAV. In short, the LR anti-spoofing model can effectively achieve the effect of anti-GPS spoofing in the scene of UAV flying along the specified route.

Last but not least, although the LR anti-spoofing model successfully resists GPS spoofing and ensures the maximum completion of UAV tasks, strictly speaking, it is only a spoofing mitigation method. In the future work, we will further optimize our method from the perspective of UAV sensor integrated navigation and UAV attitude control, hoping to achieve the solution of GPS spoofing.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant No. 61771153, No. 61831007, and No. 61971154).

References

[1] J. I. Maza, F. Caballero, J. Capitán, J. R. M. de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 563–585, 2011.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (uavs) for energy-efficient Internet of things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[4] L. Kai, N. Ahmed, S. S. Kanhere, and S. Jha, "Reliable communications in aerial sensor networks by using a hybrid antenna," in *IEEE Conference on Local Computer Networks*, Clearwater Beach, FL, USA, 2012.

[5] D. P. Shepard, J. Bhatti, T. E. Humphreys, and A. Fansler, "Evaluation of smart grid and civilian uav vulnerability to gps spoofing attacks," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, Nashville, Tennessee,USA, 2012.

[6] G. Panice, S. Luongo, G. Gigante et al., "A svmbased detection approach for GPS spoofing attacks to UAV," in *23rd International Conference on Automation and Computing, ICAC 2017*, pp. 1–11, Huddersfield, United Kingdom, September 2017.

[7] Y. Qiao, Y. Zhang, and X. Du, "A vision-based gpsspoofing detection method for small uavs," in *13th International Conference on Computational Intelligence and Security, CIS 2017*, pp. 312–316, Hong Kong, China, December 2017.

[8] W. U. Bin and L. I. U. Hanwen, "A behavior-based covert channel based on GPS deception for smart mobile devices," in *2019 IEEE International Conference on Communications, ICC 2019*, pp. 1–6, Shanghai, China, May 2019.

[9] B. Van den Bergh and S. Pollin, "Keeping uavs under control during GPS jamming," *IEEE Systems Journal*, vol. 13, no. 2, pp. 2010–2021, 2019.

[10] J. Noh, Y. Kwon, Y. Son et al., "Tractor beam," *ACM Transactions on Privacy and Security*, vol. 22, no. 2, pp. 1–26, 2019.

[11] F. A. Milaat and H. Liu, "Decentralized detection of GPS spoofing in vehicular ad hoc networks," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1256–1259, 2018.

[12] L. Zhang, W. Hu, W. Qu, Y. Guo, and S. Li, "A formal approach to verify parameterized protocols in mobile cyber-physical systems," *Mobile Information Systems*, vol. 2017, Article ID 5731678, 10 pages, 2017.

[13] E. G. Manfredini and F. Dovis, "On the use of a feedback tracking architecture for satellite navigation spoofing detection," *Sensors*, vol. 16, no. 12, article 2051, 2016.

[14] A. R. Eldosouky, A. Ferdowsi, and W. Saad, "Drones in distress: a game-theoretic countermeasure for protecting uavs against GPS spoofing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2840–2854, 2020.

[15] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of uav: a survey," *Mobile Networks and Applications*, vol. 25, no. 1, pp. 95–101, 2020.

[16] W. M. Y. W. Bejuri, W. M. N. W. M. Saidin, M. M. B. Mohamad, M. Sapri, and K. S. Lim, "Ubiquitous positioning: integrated gps/wireless LAN positioning for wheelchair navigation system," in *Volume 7802 of Lecture Notes in Computer Science*, A. Selamat, N. T. Nguyen, and H. Haron, Eds., pp. 394–403, Springer, 2013.

[17] P. Moosbrugger, K. Y. Rozier, and J. Schumann, "R2U2: monitoring and diagnosis of security threats for unmanned aerial systems," *Formal Methods in System Design*, vol. 51, no. 1, pp. 31–61, 2017.

[18] A. Broumandan, A. Jafarnia-Jahromi, S. Daneshmand, and G. Lachapelle, "Overview of spatial processing approaches for gnss structural interference detection and mitigation," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1–12, 2016.

[19] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "Gps spoofing detection via dual-receiver correlation of military signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2250–2267, 2013.

[20] M. L. Psiaki and T. E. Humphreys, "Gnss spoofing and detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[21] X. Hu, J. Cheng, M. Zhou et al., "Emotion-aware cognitive system in multichannel cognitive radio ad hoc networks," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 180–187, 2018.

[22] A. Ranganathan, H. Ólafsdóttir, and S. Capkun, "SPREE: a spoofing resistant GPS receiver," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, MobiCom 2016*, pp. 348–360, New York City, NY, USA, October 2016.

[23] C. H. Kang, S. Y. Kim, and C. G. Park, "Adaptive complex-ekf-based doa estimation for gps spoofing detection," *IET Signal Processing*, vol. 12, no. 2, pp. 174–181, 2018.

[24] K. D. Wesson, M. Rothlisberger, and T. E. Humphreys, "Practical cryptographic civil gps signal authentication," *Navigation*, vol. 59, no. 3, pp. 177–193, 2012.

[25] H. Rao, S. Wang, X. Hu et al., "Self-supervised gait encoding with locality-aware attention for person re-identification," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 898–905, Yokohama, Japan, 2020, http://ijcai.org.

[26] J. Magiera and R. Katulski, "Detection and mitigation of gps spoofing based on antenna array processing," *Journal of Applied Research and Technology*, vol. 13, no. 1, pp. 45–57, 2015.

[27] K. Jansen, M. Schafer, D. Moser, V. Lenders, C. Popper, and J. Schmitt, "Crowd-gps-sec: leveraging crowdsourcing to detect and localize gps spoofing attacks," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 1018–1031, San Francisco, CA, USA, 2018.

[28] K. C. Kwon and D. S. Shim, "Performance analysis of direct gps spoofing detection method with ahrs/accelerometer," *Sensors (Basel, Switzerland)*, vol. 20, no. 4, p. 954, 2020.

[29] Y. Tang, X. Zhang, X. Hu, S. Wang, and H. Wang, "Facial expression recognition using frequency neural network," *IEEE Transactions on Image Processing*, vol. 30, pp. 444–457, 2021.

[30] A. Sinha, P. Malo, A. Frantsev, and K. Deb, "Finding optimal strategies in a multi-period multi-leader-follower stackelberg game using an evolutionary algorithm," *Computers & Operations Research*, vol. 41, pp. 374–385, 2014.

[31] N. Groot, B. De Schutter, and H. Hellendoorn, "Optimal affine leader functions in reverse Stackelberg games," *Journal of Optimization Theory and Applications*, vol. 168, no. 1, pp. 348–374, 2016.

[32] L. Zhang, Q. WanXia, Y. Huo, G. Yang, and S. Li, "An sat-based method to multithreaded program verification for mobile crowdsourcing networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3193974, 8 pages, 2018.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, "Data prediction model in wireless sensor networks based on bidirectional LSTM," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[35] S. Xu, H. Rao, H. Peng, X. Jiang, and B. Hu, "Attention based multi-level co-occurrence graph convolutional lstm for 3d action recognition," *IEEE Internet of Things Journal*, vol. 99, p. 1, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5577530, 9 pages
https://doi.org/10.1155/2021/5577530

Research Article

Reconstructing 3D Model from Single-View Sketch with Deep Neural Network

Fei Wang[1], **Yu Yang**[2], **Baoquan Zhao**[3], **Dazhi Jiang**[1], **Siwei Chen**[1] and **Jianqiang Sheng**[4]

[1]*Shantou University, Shantou, China*
[2]*Shenzhen Securities Information Co., Ltd, Shenzhen, China*
[3]*Guilin University of Electronic Technology, Guilin, China*
[4]*Shenzhen Institute of Information Technology, Shenzhen, China*

Correspondence should be addressed to Baoquan Zhao; zbqsys@gmail.com

Received 8 January 2021; Revised 5 March 2021; Accepted 26 March 2021; Published 27 April 2021

Academic Editor: Amr Tolba

In this paper, we introduce a novel 3D shape reconstruction method from a single-view sketch image based on a deep neural network. The proposed pipeline is mainly composed of three modules. The first module is sketch component segmentation based on multimodal DNN fusion and is used to segment a given sketch into a series of basic units and build a transformation template by the knots between them. The second module is a nonlinear transformation network for multifarious sketch generation with the obtained transformation template. It creates the transformation representation of a sketch by extracting the shape features of an input sketch and transformation template samples. The third module is deep 3D shape reconstruction using multifarious sketches, which takes the obtained sketches as input to reconstruct 3D shapes with a generative model. It fuses and optimizes features of multiple views and thus is more likely to generate high-quality 3D shapes. To evaluate the effectiveness of the proposed method, we conduct extensive experiments on a public 3D reconstruction dataset. The results demonstrate that our model can achieve better reconstruction performance than peer methods. Specifically, compared to the state-of-the-art method, the proposed model achieves a performance gain in terms of the five evaluation metrics by an average of 25.5% on the man-made model dataset and 23.4% on the character object dataset using synthetic sketches and by an average of 31.8% and 29.5% on the two datasets, respectively, using human drawing sketches.

1. Introduction

3D shape reconstruction as one of the most fundamental problems in computer graphics is playing an increasingly important role in a wide variety of fields such as virtual/augmented reality, computer-aided geometric design, gaming, and medical imaging. However, manually reconstructing 3D models from scratch is a nontrivial task. This is because the procedure generally involves intensive interactions, which will take a designer a lot of time and effort to craft an exquisite 3D model. To alleviate this situation, a large body of approaches have been developed to facilitate the 3D shape reconstruction process. Among them, automatically reconstructing 3D models from sketch images is gaining more and more popularity due to its high efficacy and simplicity of interaction.

To harvest 3D models from free-hand sketches, a crucial factor is how to accurately understand their semantic meaning. Towards this end, conventional sketch-based 3D shape reconstruction methods like [1] rely on hand-crafted features. With the recent advance in deep learning technology, deep neural network-based 3D shape reconstruction has achieved remarkable progress. Despite these achievements, there are still many challenging issues in this area that have not been effectively addressed, which are seriously hindering the adoption of this technique in many domains. These issues are mainly featured in the following three aspects. Firstly, there is a great semantic gap between sketch images and 3D

models. Compared with a nature image, a sketch is a visual representation form with a high level of abstraction and can easily cause ambiguity. This character could bring great challenges to the understanding of sketches and will affect the performance of 3D reconstruction. Besides, drawing skills and painting styles of different users vary greatly, which further exacerbate the difficulties of sketch semantic analysis. Secondly, deep learning-based 3D reconstruction models are generally highly dependent on sufficient training data. With the diversity of users' painting styles, it is very time-consuming and costly to collect tens of thousands of well-labelled sketch images that can be used to feed into deep neural networks for effective training. Last but not least, the dimensionality and features of sketches and 3D models are quite different. How to effectively exploit the very limited visual clues in sketch images to reconstruct 3D shape accurately is another challenging task that has not been adequately addressed by existing studies.

To tackle the aforementioned issues, in this paper, we introduce a novel 3D shape reconstruction framework from a sketch image using a deep neural network. Unlike conventional methods that need several sketch images from multiple views, the proposed approach only takes a single-view sketch image as input. This can significantly reduce the interaction during the reconstruction process, which is a very important factor that is highly concerned by practitioners in real-world applications. However, compared with multiview-based reconstruction methods, extracting meaningful features from a single-view sketch may be insufficient for accurate 3D shape reconstruction. This is because sketches from multiple views are more likely to convey more useful features to infer the underlying structure of a 3D shape. To gain the merit of multiview sketch-based reconstruction frameworks, we formulate the problem as a three-stage task, i.e., we first segment an input sketch into a series of basic units in the first stage and use the units to build a transformation template and create multifarious sketches in the second stage before reconstructing 3D shape in the third stage. Such a strategy paves the way to harvesting high-quality 3D models with less input and human-computer interaction. This work is an extension of our conference paper [2]. In this extended version, we introduce a new network optimization component and conduct more experiments to evaluate the effectiveness of our method.

The rest of this paper is organized as follows. In the next section, we briefly introduce existing researches on 3D shape reconstruction from sketch images. The details of the proposed framework are presented in Section 3. We conduct an extensive experiment to demonstrate the effectiveness of the proposed method in Section 4. Section 5 concludes the paper.

2. Related Work

In this section, we briefly review existing researches that are closely related to our work from two aspects: (1) sketch understanding and (2) sketch-based composition and reconstruction.

2.1. Sketch Understanding. Sketch understanding is an emerging branch in the field of artificial intelligence [3–5], aimed at recognizing and extracting the semantic knowledge from sketch images in a fully/semiautomatic fashion. It mainly encompasses two parts, i.e., sketch recognition and semantic understanding [6]. A pioneer work on large-scale sketch recognition is "How do users draw sketches?" [1], in which 20 k sketch images in total were collected. It adopts a Gaussian derivative to estimate the direction of lines and utilizes a bag-of-words model to encode a local curve direction as the feature vector of sketches. Then, the support vector machine (SVM) is employed to classify sketch images. With the great achievement of the deep neural network (DNN) on natural image recognition, the convolutional neural network (CNN) has also been applied to sketch recognition. However, taking sketch images as a 2D pixel array, CNN-based methods usually need to learn a huge amount of network parameters, which is very inefficient to train the network. Compared with natural images, the features of sketch images are sparse. Such spare data can significantly improve the compactness of networks.

2.2. Sketch-Based Composition and Reconstruction. A large body of studies have been carried out to extract features from sketch images and use them to perform a variety of tasks such as 3D model retrieval and shape reconstruction. For example, Chen and Fang [7] proposed to retrieve a 3D model using a sketch by constructing two individual deep CNN and metric networks. One network is for sketch images, and the other one is for 3D models. Interleaved active metric learning (IAML) is used to learn specific features from these two modalities, which is capable of mining important features from samples for training and learning discriminative feature representation effectively. Besides, to reduce the cross-modality difference between sketch features and 3D shapes, it also introduced a modality transformation network to convert sketch features into the feature space of 3D models, which achieves better retrieval performance.

This technique also paves the way for the development of sketch-based 3D shape reconstruction [8, 9]. Wang et al. [10] developed a label-free sketch neural network 3D-GAN. This model is integrated with an embedding latent vector space to harvest a similar feature vector distribution between sketch image and 2D rendered image. Then, it obtains final results by retrieving the k most similar 3D models with a sketch as the prior knowledge. Lun et al. [11] mapped a sketch to 3D shape by training a ConvNet to infer the structure and reconstructed a 3D model using a multiview framework. Unlike conventional methods that adopt voxels to represent 3D models, the shape of 3D objects can be represented with surface-based forms (for example, polygon mesh), which can achieve more accurate prediction by combining feedforward frameworks.

3. The Proposed Deep Neural Network for 3D Shape Reconstruction

3.1. Overview of the Proposed Method. The proposed pipeline is mainly composed of three components: (1) sketch component segmentation based on multimodal DNN fusion, (2) multifarious sketch generation based on nonlinear transformation network, and (3) deep 3D shape reconstruction using

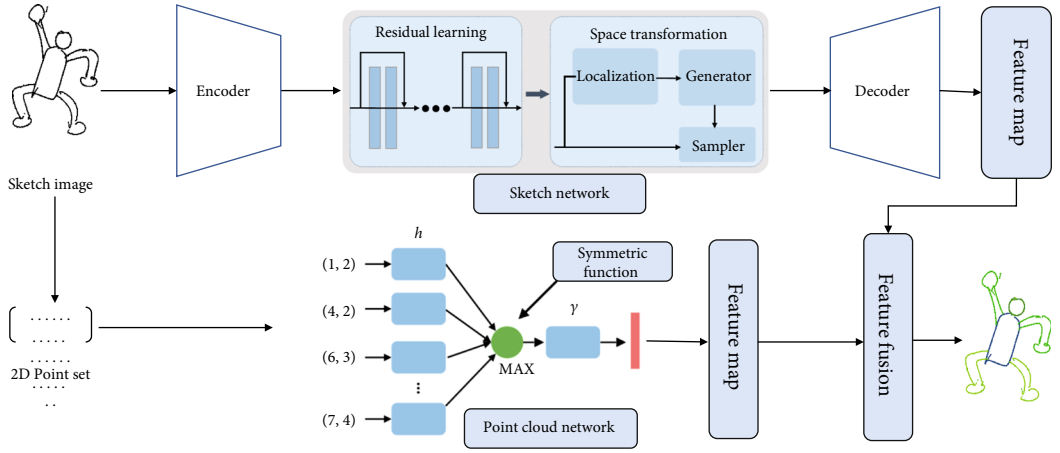


Figure 1: Sketch component segmentation based on multimodal DNN fusion.

multifarious sketches. The first component is a 2D point cloud-based sketch segmentation model, which is used to segment a given sketch into a series of basic units and build a transformation template by the knots between them. The advantage of this component is that it only relies on a small amount of sample data to achieve the learning task of the transformation network. The second component is developed for generating multifarious sketches with the aforementioned transformation template. It creates the transformation representation of a sketch by extracting the shape features of an input sketch and transformation template samples, which can avoid the problems existing in conventional models such as unitary transformation structure and distortion. The third component takes the obtained sketches as input to reconstruct 3D shapes with a generative model. It fuses and optimizes features of multiple views and thus is more likely to generate high-quality 3D shapes. Details of each component will be introduced in the following three subsections.

3.2. Sketch Component Segmentation Based on Multimodal DNN Fusion. Sketch component segmentation net is inspired by work [6]. The sketch network mainly consists of two parts: encoder and decoder. On the one hand, the network obtains the global feature of a sketch through the encoder. As shown in Figure 1, feature representation is learned and extracted using spatial invariance enhanced residual (SIER), which is composed of two modules: residual learning module and spatial transformation module. These features will be combined together in the decoding phrase to generate a pixel-level feature segmentation image. On the other hand, the coordinate information of sketch contour is an important geometry structure. Therefore, a 2D point cloud network can obtain the feature of a sketch by representing each point with 2D coordinates (x, y). Let $P = \{p_i \mid i = 1,\cdots,n\}$ be the coordinate set of sketch contour, where p_i represents the coordinate of each sample point; the 2D point cloud network takes P as input and gets the global features by gathering point features with maximum function MAX. Then, the probability of each point in P associated with all semantic units can be obtained by connecting local and global features through a segmentation network. More specifically, let $f : X \to R$ be a continuous set function regarding Hausdorff distance $d_H(\cdot, \cdot)$. For $\forall \varepsilon > 0$, there exists a continuous function $g(x_1,\cdots,x_n) = \gamma \circ \mathrm{MAX}$ such that for arbitrary $x_i \in X$,

$$|f(\{x_i,\cdots,x_n\}) - \gamma(\mathrm{MAX}\,(h(x_q), h(x_2),\cdots,h(x_n)))| < \varepsilon, \quad (1)$$

where γ and h are continuous functions and MAX is a vector maximum description operator. Therefore, the general function of the arbitrary 2D point set can be represented as

$$f(\{x_1,\cdots,x_n\}) \approx g(h(x_1),\cdots,h(x_n)). \quad (2)$$

As shown in Figure 1, h can be learned with multilayer perception (MLP), and $g = \gamma \circ \mathrm{MAX}$ can be obtained with single variable function and max pooling.

After performing sketch segmentation, we use the transformation templates of the original sketch to train the network. As illustrated in Figure 2, a sketch is segmented into a series of semantic units. The knot between units is represented with $\{(p_i, q_i)\}$, which is highlighted with a red circle in the figure.

3.3. Multifarious Sketch Generation Based on Nonlinear Transformation Network. As shown in Figure 3, the proposed pipeline is mainly composed of two modules, i.e., multiview sketch generation module and 3D model generation module. The aim of segmentation is to obtain small samples of sketches for the use of training the network in Section 3.2. Multiview sketches are generated by the Sketch-VAE network with the input sketch image. The obtained multiview sketches are then fed into the encoder and decoder to harvest a depth image and a normal image, with which we can generate a 3D point cloud. And finally, a 3D polygon model can be obtained by performing remeshing on the point cloud.

In this step, we first carry out transformation feature extraction and representation. For $m(m \geq 2)$ sketch images, each of which has n vertexes, let p_i be the v_ith vertex of an input sketch and p'_i be the one in the transformation

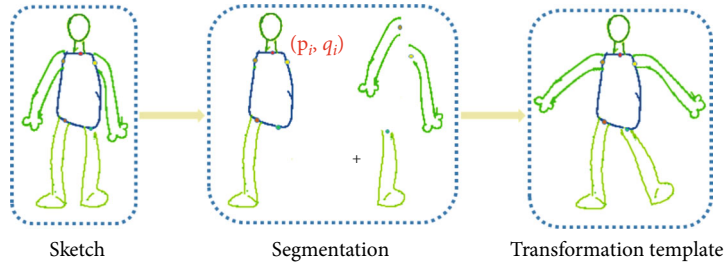


Figure 2: Sketch segmentation, knots, and transformation template.

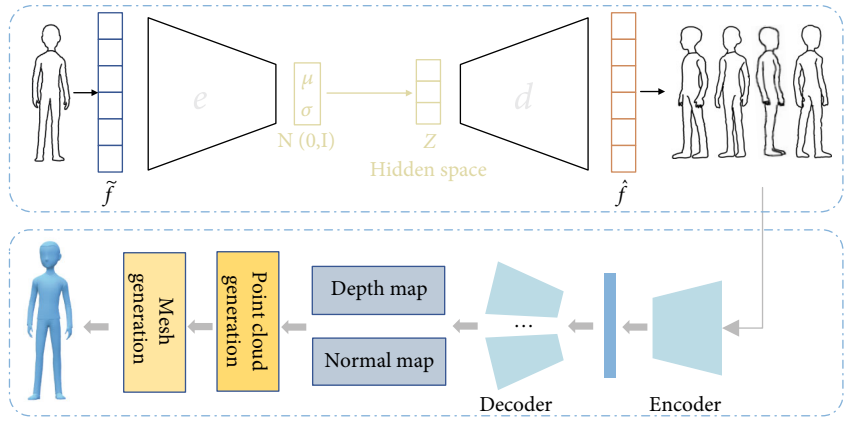


Figure 3: Deep 3D shape reconstruction using multifarious sketches.

template; the transformation gradient T_i can be obtained by minimizing the energy function below:

$$E(T_i) = \sum_{j \in N_i} c_{ij} \| e'_{ij} - T_i e_{ij} \|^2, \tag{3}$$

where N_i is the neighborhood set of v_i, $e'_{ij} = p'_i - p'_j$, $e_{ij} = p_i - p_j$, and $c_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$ is cotangent weight. The affine transformation matrix can be decomposed into rotation part and scaling part $T_i = R_i S_i$. The rotation difference from v_i to v_j is given by

$$dR_{ij} = R_i^T R_j. \tag{4}$$

Thus, the energy function is redefined as

$$E(T_i) = \sum_{j \in N_i} c_{ij} \sum_{t \in N_i} c'_i \| e'_{ij} - R_i dR_{ti} S_i e_{ij} \|^2, \tag{5}$$

where $c'_i = 1/|N_i|$ and $|N_i|$ is the number of neighborhood of v_i. Finally, the feature of the transformation template on v_i can be represented as

$$f_i^j = \{ \log dR_{ij} ; S_i \} \ (\forall i, j \in N_i). \tag{6}$$

The Sketch-VAE network is aimed at finding an encoder and a decoder, where the goal of the encoder is to map the posterior distribution of x to hidden vector z, while that of the decoder is to generate a credible x. The loss function of the Sketch-VAE model is defined as

$$L_{\text{VAE}} = \sum_{j=1}^{M} \sum_{i=1}^{K} \left(f \wedge_i^j - \tilde{f}_i^j \right)^2 + D_{KL} \left(q \left(z \mid \tilde{f} \right) \| p(z) \right), \tag{7}$$

where $\tilde{f}_i^j$ is the transformed feature after preprocessing and $\widehat{f}_i^j$ is the output of the Sketch-VAE model. z is a hidden vector; $p(z)$ and $q(z \mid \tilde{f})$ are prior and posterior probability, respectively; and D_{KL} is KL divergence.

To avoid incorrect output caused by the separation between sketch components, we add a constrain condition and define the loss function of knots as

$$L_{\text{joints}} = \sum_{i=1}^{n} \| p_i - q_i \|^2. \tag{8}$$

Besides, we also add a regularization constrain to the network optimization network to avoid distortion and define the loss function as

$$L_{\text{reg}} = \sum_{i=1}^{|V|} \sum_{j \in N_i} w_{ij} \| \left(\widehat{v}_i - \widehat{v}_j \right) - R_i \left(v_i - v_j \right) \|^2, \tag{9}$$

where v_i and $\widehat{v}_i$ are the vertexes in the original and transformed sketches, respectively.

Table 1: Segmentation accuracy (%) in terms of P-metric under different parameter settings on the SketchSeg dataset.

Loss weight λ_1	Airplane	Bicycle	Candelabra	Chair	Four legs	Human	Lamp	Rifle	Table	Vase	Average
$\lambda_1 = 0.1$	89.3	89.2	91.5	90.4	87.3	84.2	89.4	89.2	**93.4**	**95.8**	89.8
$\lambda_1 = 0.3$	**92.1**	**91.6**	**93.1**	**91.8**	**87.6**	**84.3**	**89.6**	**90.4**	88.7	92.1	90.1
$\lambda_1 = 0.5$	90.3	91.0	91.8	91.7	85.9	82.9	90.2	88.8	87.9	91.3	89.2
$\lambda_1 = 0.7$	88.7	89.6	89.4	85.3	83.6	80.6	86.9	83.5	81.6	88.5	85.8
$\lambda_1 = 0.9$	84.0	84.5	85.7	81.6	82.0	74.9	83.0	84.0	76.6	80.7	81.7

3.4. Deep 3D Shape Reconstruction Using Multifarious Sketches. The proposed deep 3D shape reconstruction framework is illustrated in Figure 3. For the sketches obtained in the last step, the encoder first extracts the feature of each sketch. It consists of a series of convolution, and all layers use ReLU as an active function. Then, the decoder transforms these features into depth and normal images, which will be fused into a 3D point cloud subsequently. The first step of fusion is to map all foreground pixels to 3D points. If the prediction probability of a pixel is larger than 50%, we consider it as a foreground pixel. Let the depth of a foreground pixel p be $p_{p.v}$, the coordinate set of graph space in the ith sketch image be $\{p_x, p_y\}$, and the position of a 3D point $q_{p,i}$ can be calculated as

$$q_{p,i} = R_v \left[\kappa p_x \kappa p_y d_{p,i}\right]^T + e_i, \tag{10}$$

where κ is a scaling coefficient, representing the distance between adjacent pixels and the center. Finally, the 3D model can be obtained by transforming it into polygon mesh with the Poisson surface reconstruction algorithm [12].

3.5. Network Optimization. To constrain the two feature vectors, we use the relative cross-entropy to evaluate their similarity and select sigmoid normalization-based cross-entropy as the loss function of the proposed sketch pixel network:

$$L_{\text{sketch pixel}} = -\frac{1}{K}\sum_{k}^{K}\left[y_k \log\left(\frac{1}{1+e^{-c^t}}\right) + (1-y_k)\log\left(\frac{e^{-c^t}}{1+e^{-c^t}}\right)\right], \tag{11}$$

where y_k indicates whether the label k exists in the obtained segmentation result. If so, we set $y_k^{cls} = 1$; otherwise, $y_k = 0$. c_k is the prediction probability that the label k appears in the results. Likewise, $L_{\text{point cloud}}$ is the feature constrain of the 2D point cloud. We can perform the feature fusion via multiple ways such as cascade and weighted sum. Meanwhile, the weights of different networks have a direct impact on the results. We will study the impact of the normalization of different network components on the results in Table 1. The objective function is formulated as

$$L = \lambda_1 L_{\text{sketch pixel}} + \lambda_2 L_{\text{point cloud}}. \tag{12}$$

4. Experiment and Discussion

4.1. Experiment Setup. To evaluate the effectiveness of the proposed method for 3D shape reconstruction, we conduct a comparative experiment on a public dataset. To train our neural network, we use the dataset presented in [11], which mainly consists of three different types of 3D models, i.e., human/humanoid, airplanes, and chairs. Among them, human/humanoid involves human models, aliens, and virtual cartoon characters, which come from *The Models Resource* dataset [13], while airplane and chair models are mainly from *3D ShapeNet* [14], which has a large variety in shape geometry. There are 120 sketch images in total in the test dataset. Among them, 90 are synthetic sketches, which are generated from test images with line painting techniques, while the rest 30 sketch images are drawn by two professional artists. They were asked to draw 10 sketch images for each category. The sketch images were scaled to a size of 800×800. We train our model for 50 epochs with an SGD optimizer. The size of minibatch, initial learning rate, and momentum was set to 5, 0.01, and 0.9, respectively. The weight parameters of cross-entropy loss are 0 for background and 0.1 for others, which can avoid the interference of the background. The experiment was conducted on a PC equipped with an Intel i7 CPU, 32 GB RAM, and GTX 2080ti GPU. For the weight parameters λ_1 and λ_2 in Equation (12), we formulate $\lambda_1 + \lambda_2 = 1$ and set λ_1 to 0.3 according to the parameter analysis described in Table 1.

Peer sketch-based 3D shape reconstruction methods selected for comparison include ShapeMVD [11], nearest retrieval, Tatarchenko et al. [15], U-net [16], volumetric decoder, and R2N2 [17], which are state-of-the-art models and widely used by existing studies for performance evaluation. For the nearest-neighbor baseline, we extract the representation of the input test sketches based on our encoder. This is used as a query representation to retrieve the training shape whose sketches have the nearest encoder representation based on the Euclidean distance. We additionally implemented a variant of Tatarchenko et al.'s decoder by adding U-net connections between the encoder and their decoder. The volumetric decoder consisted of five transpose 3D convolutions of stride 2 and kernel size $4 \times 4 \times 4$. The number of filters starts with 512 and is divided by 2 at each layer. Leaky ReLU functions and batch normalization were used

Table 2: Man-made objects (synthetic).

	Ours	ShapeMVD	Nearest retrieval	Tatarchenko et al. [15]	[15]+U-net	Volumetric decoder	R2N2
Hausdorff distance	**0.076**	0.092	0.165	0.142	0.121	0.113	0.144
Chamfer distance	**0.011**	0.015	0.025	0.022	0.017	0.021	0.026
Normal distance	**26.45**	30.66	42.57	35.58	32.32	49.40	48.78
Depth map error	**0.013**	0.026	0.049	0.039	0.030	0.038	0.045
Volumetric distance	**0.276**	0.344	0.501	0.442	0.374	0.432	0.512

Table 3: Character models (synthetic).

	Ours	ShapeMVD	Nearest retrieval	Tatarchenko et al. [15]	[15]+U-net	Volumetric decoder	R2N2
Hausdorff distance	**0.065**	0.089	0.200	0.119	0.092	0.152	0.148
Chamfer distance	**0.010**	0.015	0.036	0.025	0.016	0.026	0.032
Normal distance	**26.47**	30.61	44.93	34.98	31.00	53.84	53.13
Depth map error	**0.014**	0.018	0.040	0.030	0.019	0.031	0.036
Volumetric distance	**0.248**	0.313	0.541	0.428	0.329	0.437	0.493

Table 4: Man-made objects (human drawing).

	Ours	ShapeMVD	Nearest retrieval	Tatarchenko et al. [15]	[15]+U-net	Volumetric decoder	R2N2
Hausdorff distance	**0.094**	0.116	0.176	0.153	0.153	0.130	0.149
Chamfer distance	**0.011**	0.017	0.031	0.024	0.025	0.022	0.028
Normal distance	**21.058**	27.04	40.96	32.40	30.45	48.32	48.12
Depth map error	**0.011**	0.021	0.042	0.033	0.032	0.032	0.042
Volumetric distance	**0.202**	0.311	0.544	0.405	0.403	0.405	0.500

Table 5: Character models (human drawing).

	Ours	ShapeMVD	Nearest retrieval	Tatarchenko et al. [15]	[15]+U-net	Volumetric decoder	R2N2
Hausdorff distance	**0.102**	0.117	0.188	0.139	0.136	0.178	0.168
Chamfer distance	**0.013**	0.021	0.036	0.025	0.024	0.032	0.036
Normal distance	**28.22**	33.44	43.81	36.11	34.74	54.91	54.29
Depth map error	**0.012**	0.026	0.040	0.031	0.027	0.037	0.040
Volumetric distance	**0.217**	0.298	0.458	0.342	0.307	0.420	0.436

after each layer. We note that we did not use skip connections (U-net architecture) in the volumetric decoder because the size of the feature representations produced in the sketch image-based encoder is incompatible with the ones produced in the decoder.

4.2. Overall Reconstruction Performance Comparison against Peer Methods. Following the common practice, we also compare the similarity between the reconstructed 3D models and input sketches using the following five distance metrics: Chamfer distance, Hausdorff distance, surface normal distance, depth map error, and volumetric Jaccard distance. The comparison results are illustrated in Tables 2–5, where Tables 2 and 3 are the results of synthetic sketches while Tables 4 and 5 are those of human drawing sketches. Bold values in the tables indicate the best results among all methods. From Tables 2 and 3, we can see that the proposed method achieves smaller distances than peer ones in terms of the five evaluation metrics on both man-made objects and character models. Specifically, compared to the state-of-the-art model ShapeMVD, our model achieves a performance gain in terms of Hausdorff distance, Chamfer distance, normal distance, depth map error, and volumetric distance by 17.4%, 26.7%, 13.7%, 50.0%, and 19.8% on the man-made objects dataset and by 27.0%, 33.3%, 13.5%, 22.2%, and 20.8% on the character model dataset, respectively. Overall, the proposed method improves the performance in terms of the five distance metrics by an average of 25.5% and 23.4% on the man-made objects and character models, respectively, which demonstrates that our method can generate more accurate 3D models. We can find that ShapeMVD performs better than conventional methods like U-net. However, its performance gain is just marginally higher. Our model outperforms all of these methods, and performance gain is significant. The superiority of the proposed method can also be reflected on the human drawing sketch

Table 6: Segmentation performance of MIFNet against state-of-the-art methods with the P-metric.

Method	U-net	LinkNet	FCN	PointNet	MIFNet
Airplane	68.9	78.0	78.2	81.0	**92.9**
Bicycle	68.1	65.3	71.4	78.0	**93.5**
Candelabra	89.3	88.3	**90.8**	81.1	93.0
Chair	84.0	89.1	86.9	81.0	**88.1**
Four legs	74.1	76.7	80.3	75.5	**89.3**
Human	71.9	74.5	75.6	69.2	**85.5**
Lamp	92.2	91.2	92.8	86.2	**87.8**
Rifle	54.8	59.9	65.2	83.2	**89.7**
Table	79.6	82.5	81.4	82.0	**88.6**
Vase	89.9	93.8	**94.4**	84.8	92.8
Average	77.3	79.9	81.7	80.2	**90.1**

Table 7: Segmentation performance of MIFNet against state-of-the-art methods with the C-metric.

Method	U-net	LinkNet	FCN	PointNet	MIFNet
Airplane	52.6	67.7	66.5	67.3	**86.5**
Bicycle	49.7	55.7	**59.2**	50.9	85.5
Candelabra	90.3	89.0	**94.5**	67.9	94.8
Chair	81.9	89.2	84.8	77.6	**85.1**
Four legs	54.5	67.2	73.5	60.9	**84.2**
Human	62.6	67.9	**72.1**	56.6	81.2
Lamp	92.4	92.4	92.5	86.1	**87.0**
Rifle	38.9	44.5	54.7	59.7	**82.1**
Table	70.1	80.3	75.3	67.5	**82.6**
Vase	90.7	96.6	**98.1**	78.9	93.1
Average	68.4	75.0	77.1	67.3	**86.2**

dataset, as shown in Tables 4 and 5. Likewise, the distance reduction compared with other methods is also prominent. For example, compared to ShapeMVD, the proposed method reduces the Chamfer distance by 33.3% and 38.1% on the man-made object objects and character models, respectively. Overall, our method achieves a performance gain on these two human drawing sketch datasets by an average of 31.8% and 29.5%, respectively. These experimental results demonstrate the effectiveness of the proposed method in reconstructing 3D shape from single-view sketches.

4.3. Evaluation of the Effectiveness of the Proposed Sketch Segmentation Method. To further verify the effectiveness of the proposed sketch segmentation method, we conducted an experiment on a public sketch segmentation database, i.e., SketchSeg. The two evaluation metrics used in this paper are pixel-based accuracy (P-metric) and component-based accuracy (C-metric), which is first proposed by Huang et al. [18] and widely used by peer work. Among them, P-metric is targeted for the evaluation of a whole sketch image, i.e., the percentage of the pixels of components that are correctly segmented to the pixels of the entire sketch, while C-metric is defined as the percentage of the number of components that are correctly segmented to the total number of components of the sketch. We treat a component as a correctly segmented one if more than 75% of its pixels are correctly predicted. The comparison between multisource information fusion (MIFNet) and peer methods is shown in Table 6. We can see from the table that the performance of MIFNet is superior to other methods. More specifically, it achieves an accuracy of 90.1% on average, while those of U-Net, LinkNet, FCN, and PointNet are 77.3%, 79.9%, 81.7%, and 80.2%, respectively. In other words, the proposed method improves the performance by 12.8%, 10.2%, 8.4%, and 9.9%, respectively. Besides, the component-based accuracy of peer methods is 68.4%, 75.0%, 77.1%, and 67.3%, respectively. MIFNet outperforms the FCN by 9.1% in terms of segmentation accuracy. Particularly, we can observe that the performance improvement on an *airplane*, *chair*, *human*, *gun*, and *desk* is more significant than others. This is because the average number of pixels of these five categories is higher than other ones. As for C-metric, the proposed method also shows an advantage over peer ones. As can be seen from Table 7, MIFNet improves the accuracy by 9.1%, compared to FCN, which demonstrates that MIFNet is more effective in capturing the structural information of sketch components.

4.4. The Selection of Loss Function Weight. Table 1 illustrates the comparison of different loss function weight λ_1 where λ_1 is the weight of pixel-based network while 1-λ_1 represents that of the point cloud-based network. We can see from the table that the segmentation results are less satisfactory when λ_1 is large, and the performance gets better gradually with the decrease of λ_1, which demonstrates that the point cloud-based network plays a more important role than the pixel-based one. Particularly, when $\lambda_1 = 0.3$, the performance of the pixel-based network reaches its peak and achieves an accuracy of 90.1%. In light of this, we thus train the model with this setting in our experiment.

5. Conclusion

In this paper, we have introduced a novel 3D shape reconstruction method from a single-view sketch image using a deep neural network. Our model is general and can be easily extended to other applications, such as biomedical and intelligent computing [19, 20]. The proposed method first generates a series of sketch images from different viewpoints by analysing the semantic information of the input sketch image. Then, the obtained sketch images are fed into a deep neural network to reconstruct the 3D shapes. Compared with multiview-based approaches, the proposed method only takes a single sketch image as input, which can significantly reduce time used for drawing sketches and remarkably improve the reconstruction efficiency. Besides, using the input sketch image as visual clues to generate multiview sketch images is helpful to reconstruct 3D shapes more accurately, which is superior to conventional single sketch image-based 3D shape reconstruction methods. Extensive experiments on a public 3D shape reconstruction dataset have demonstrated the efficacy of the proposed model.

One of the most challenging issues related to sketch-based 3D shape reconstruction is that the painting skills

and styles of different users vary greatly, which makes it difficult to develop a versatile model to successfully extract meaningful features and infer semantic information from all kinds of sketches. A limitation of the proposed method is that it may fail to accurately reconstruct a high-quality 3D model if the input sketch is painted at a very abstract level or can easily cause ambiguity. Therefore, more powerful deep neural networks and machine learning techniques such as [21, 22] will be a promising way to address the challenges and further improve the reconstruction performance. Besides, the proposed method can be extended to sketch-based dynamic 3D model creation, which is used to be a time-consuming and labour-intensive task in the field of cartoon animation.

Data Availability

The source codes used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61902087 and 61902232), Natural Science Foundation of Guangxi (2018GXNSFAA294127), General Universities and Colleges Young Innovative Talents Project of Guangdong Province (2019GKQNCX120 and 2019GKQNCX121), Scientific Research Start-up Fund of Shantou University (09420021), Natural Science Foundation of Guangdong Province (2021A1515012302, S2018A030313420, and 2019A1515010943), Key Project of Basic and Applied Basic Research of Colleges and Universities in Guangdong Province (Natural Science) (2018KZDXM035), Basic and Applied Basic Research of Colleges and Universities in Guangdong Province (Special Projects in Artificial Intelligence) (2019KZDZX1030), and 2020 Li Ka Shing Foundation Cross-Disciplinary Research Grant (2020LKSFG05D and 2020LKSFG04D).

References

[1] M. Eitz, J. Haysy, and M. Alexa, "How do humans sketch objects?," *Acm Transactions on Graphics*, vol. 31, no. 4CD, pp. 1–10, 2012.

[2] F. Wang, Y. Yu, B. Zhao et al., "Deep 3D shape reconstruction from single-view sketch image," in *The 8th International Conference on Digital Home*, Dalian, China, 2020.

[3] D. Jiang, G. Tu, D. Jin et al., "A hybrid intelligent model for acute hypotensive episode prediction with large-scale data," *Information Sciences*, vol. 546, pp. 787–802, 2021.

[4] D. Jiang, K. Wu, D. Chen et al., "A probability and integrated learning based classification algorithm for high-level human emotion recognition problems," *Measurement*, vol. 150, article 107049, 2020.

[5] D. Jiang, Z. Tian, Z. He, G. Tu, and R. Huang, "A framework for designing of genetic operators automatically based on gene expression programming and differential evolution," *Natural Computing*, vol. 6, 2021.

[6] F. Wang, S. Lin, H. Wu et al., "SPFusionNet: sketch segmentation using multi-modal data fusion," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1654–1659, Shanghai, China, 2019.

[7] J. Chen and Y. Fang, "Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 605–620, Munich, Germany, 2018.

[8] F. Wang, S. Lin, H. Li et al., "Multi-column point-CNN for sketch segmentation," *Neurocomputing*, vol. 392, pp. 50–59, 2020.

[9] F. Wang, S. Lin, X. Luo, B. Zhao, and R. Wang, "Query-by-sketch image retrieval using homogeneous painting style characterization," *Journal of Electronic Imaging*, vol. 28, no. 2, article 023037, 2019.

[10] L. Wang, C. Qian, J. Wang, and Y. Fang, "Unsupervised learning of 3D model reconstruction from handdrawn sketches," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1820–1828, Seoul, Republic of Korea, 2018.

[11] Z. Lun, M. Gadelha, E. Kalogerakis, S. Maji, and R. Wang, "3D shape reconstruction from sketches via multi-view convolutional networks," in *2017 International Conference on 3D Vision (3DV)*, pp. 67–77, Qingdao, China, 2017.

[12] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.

[13] T. M. Resource, 2017, https://www.models-resource.com/.

[14] A. X. Chang, T. Funkhouser, L. Guibas et al., "ShapeNet: an information-rich 3D model repository," 2015, https://arxiv.org/abs/1512.03012.

[15] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D models from single images with a convolutional network," in *Computer Vision – ECCV 2016: 14th European Conference*, vol. 9911, pp. 322–337, Amsterdam, The Netherlands, 2016.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, Springer International Publishing, 2015.

[17] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: a unified approach for single and multi-view 3d object reconstruction," in *Computer Vision – ECCV 2016: 14th European Conference,*, vol. 9912, pp. 628–644, Amsterdam, The Netherlands, 2016.

[18] Z. Huang, H. Fu, and R. W. H. Lau, "Data-driven segmentation and labeling of freehand sketches," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–10, 2014.

[19] D. Jiang, Z. He, Y. Lin, Y. Chen, and L. Xu, "An improved unsupervised single channel speech separation algorithm for processing speech sensor signals," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6655125, 13 pages, 2021.

[20] D. Jiang, D. Jin, J. Zhuang, D. Tan, D. Chen, and Y. Liang, "A computational model of emotion based on audio-visual stimuli understanding and personalized regulation with concurrency," *Concurrency and Computation: Practice and Experience*, vol. 17, 2021.

[21] C. Ieracitano, A. Paviglianiti, M. Campolo, A. Hussain, E. Pasero, and F. C. Morabito, "A novel automatic classification system based on hybrid unsupervised and supervised machine learning for electrospun nanofibers," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 64–76, 2021.

[22] Q. Deng, L. Ma, A. Jin, H. Bi, B. H. Le, and Z. Deng, "Plausible 3D face wrinkle generation using variational autoencoders," *IEEE Transactions on Visualization & Computer Graphics*, vol. 1, 2021.

[23] F. Wang, S. Lin, X. Luo, and R. Wang, "Coupling computation of density-invariant and divergence-free for improving incompressible SPH efficiency," *IEEE Access*, vol. 8, pp. 135912–135919, 2020.

[24] L. Cai, Y. Yu, S. Zhang, Y. Song, Z. Xiong, and T. Zhou, "A sample-rebalanced outlier-rejected k -nearest neighbor regression model for short-term traffic flow forecasting," *IEEE Access*, vol. 8, pp. 22686–22696, 2020.

[25] H. Lu, D. Huang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "St-trafficnet: a spatial-temporal deep learning network for traffic forecasting," *Electronics*, vol. 9, no. 9, pp. 1474–1517, 2020.

[26] H. Lu, Z. Ge, Y. Song, D. Jiang, T. Zhou, and J. Qin, "A temporal-aware lstm enhanced by loss-switch mechanism for traffic flow forecasting," *Neurocomputing*, vol. 427, pp. 169–178, 2021.

[27] C. Li, S. Tang, H. K. Kwan, J. Yan, and T. Zhou, "Color correction based on cfa and enhancement based on retinex with dense pixels for underwater images," *IEEE Access*, vol. 8, pp. 155732–155741, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5577393, 14 pages
https://doi.org/10.1155/2021/5577393

Research Article

Multideep Feature Fusion Algorithm for Clothing Style Recognition

Yuhua Li,[1] **Zhiqiang He**,[1] **Sunan Wang**,[2] **Zicheng Wang**,[1] **and Wanwei Huang**[1]

[1]*Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450001, China*
[2]*School of Electronic & Communication Engineering, Shenzhen Polytechnic, Shenzhen 518055, China*

Correspondence should be addressed to Sunan Wang; wsntemp@163.com

Received 7 January 2021; Revised 11 March 2021; Accepted 3 April 2021; Published 17 April 2021

Academic Editor: Amr Tolba

In order to improve recognition accuracy of clothing style and fully exploit the advantages of deep learning in extracting deep semantic features from global to local features of clothing images, this paper utilizes the target detection technology and deep residual network (ResNet) to extract comprehensive clothing features, which aims at focusing on clothing itself in the process of feature extraction procedure. Based on that, we propose a multideep feature fusion algorithm for clothing image style recognition. First, we use the improved target detection model to extract the global area, main part, and part areas of clothing, which constitute the image, so as to weaken the influence of the background and other interference factors. Then, the three parts were inputted, respectively, to improve ResNet for feature extraction, which has been trained beforehand. The ResNet model is improved by optimizing the convolution layer in the residual block and adjusting the order of the batch-normalized layer and the activation layer. Finally, the multicategory fusion features were obtained by combining the overall features of the clothing image from the global area, the main part, to the part areas. The experimental results show that the proposed algorithm eliminates the influence of interference factors, makes the recognition process focus on clothing itself, greatly improves the accuracy of the clothing style recognition, and is better than the traditional deep residual network-based methods.

1. Introduction

Due to the prosperity of economy, people pursue personal spiritual value on the basis of satisfying the material life. People's aesthetics standard of clothing is also gradually improving unconsciously. They are no longer satisfied with the basic functional characteristics of covering up and heating and begin to pay attention to the aesthetics and personalized decorative characteristics of clothing [1]. Nowadays, people prefer to "look different" in their clothes and want to have a unique personal style [2]. Therefore, successful clothes always have distinct style characteristics.

With the introduction of the concept of deep learning, computer vision has been greatly developed [3, 4]. The computer vision technology completes the recognition and classification of images. The computer is also used to analyze and understand the image content, simulate the thinking mode of human, and automatically extract the image features [5–7]. At present, deep learning performs well in visual recognition, speech recognition, image recognition, and other aspects. In this background, based on deep learning and style characteristics of clothing, this paper proposes to take the advantages of object detection and improved deep residual network to automatically extract image features to recognize clothing styles.

He et al. [8] combined the needs of comfort, security, and beauty with clothing fabric, sewing quality, style, size, and other aspects to obtain the design elements of student clothing. Bengio et al. [9] established a connection between the Kansei engineering theory and fashion style design elements, analyzed fashion styles, colors, fabrics, and other elements of clothing, and applied the Kansei engineering theory to fashion style evaluation. Szegedy et al. [10] firstly used the action-movement tracking technology to find the parts that could most influence the style of clothing and ranked them according to their influence weight from high to low. Secondly, they collected the vocabulary describing the style of clothing and obtained the representative factors describing

the style by using the semantic difference method, which were made up of three words. Finally, they utilized Kansei engineering and fuzzy mathematics theory to establish a clothing style model, which is used for quantitative analysis of the relationship between clothing components and style. Bengio et al. [9] applied Kansei engineering to the field of clothing research. The research designed an evaluation scale first, combining with consumers' subjective evaluation of the dress style, and finally analyzed the style characteristics represented by each style and sorted them out.

YOLO [5, 11] proposed by Redmon is an earlier end-to-end detection method. The input image is first divided into $s \times s$ grid cells, and then, the direct input is resized to the convolution neural network structure that consisted of 24 convolutions with two full connection layers. The network output as a tensor includes the dimensions of each unit and is responsible for detecting the target frame of four coordinates, a positioning confidence level, and the probability value belonging to each category. YOLO also sets a multitask loss function that is compatible with border position coordinate prediction, confidence prediction, and target category prediction meanwhile for model training. Statistics in the paper show that YOLO can be as fast as 45 fps but YOLO still has many drawbacks. The most typical defects include the poor performance of YOLO in detecting small objects and nearby features. At the same time, fixed YOLO input leads to slow detection speed and an unstable network structure requires a lot of calculation.

Through in-depth study, this paper proposes a multifeature fusion recognition algorithm for clothing style based on the improved residual network and target detection model based on YOLOv3 [12]. In order to eliminate the interference factors such as the background of the clothing image, at the same time extract the comprehensive and detailed features, the proposed method extracts multicategory areas of the global areas, main parts, and part areas from the clothing image by our model. In order to extract the features of the areas, this paper improves the residual network (ResNet). The improved ResNet is trained by multilabel images beforehand, to enhance the ability of feature extraction for multicategory areas. By combining together the features of the multicategory parts extracted through improved ResNet of different categories, this paper uses an effective multifeature fusion method to realize the recognition of clothing style.

The main contribution of this paper includes the following aspects:

(i) A multicategory feature extraction model (MFEM) is proposed. We designed a multicategory clothing area extraction strategy to extract the three category areas from an image, namely, the global areas, main parts, and part areas, meanwhile eliminating the interference factors in the process of clothing style recognition. In this process, we used the target detection technology

(ii) An improved ResNet model is proposed, by improving the order of the " batch normalization layer with the activation layer with the convolutional layer" in the traditional residual block and adjusting the structure of the network convolutional kernel

(iii) A multifeature fusion method is proposed. The features of global areas, main parts, and part areas extracted by MFEM will play different roles in retrieval due to the different image scales they focus on. Although direct fusing can improve the effect, there will also be mutual influence and weakening. The multifeature fusion technology can effectively fuse different features of multicategory areas of the input image

The rest of this paper is organized as follows. In the next section, we give a brief review of the existing clothing style recognition algorithms. The proposed method is described in Section 3. In Section 4, we report experimental results on two different datasets. Finally, we conclude this paper in Section 5.

2. Related Work

With the number of images growing in the Internet, the clothing style recognition technology has become a hot research area for scientific researchers and internet companies. Schroff et al. [13] used a simple spatial local attribute classification method combined with a naive classifier for image style learning and recognition classification. Zheng et al. [14] proposed a clothing image classification method combining the face and hairstyle. This method first segments the input image into the face, hair, and clothing area, meanwhile applying PCA and GMM to each area, and then uses some known classification results to output a single score for every area, according to the user's face and hairstyle recommend appropriate clothing for the image. Yang et al. [15] believed that a jacket could be defined by its style elements, such as the collar, the printing style, and the existence of sleeves, especially the collar. Therefore, style elements such as the collar shape are important clues to distinguish clothing types. Noh et al. [16] designed a system that could be independent of the model pose, image background, and image resolution, realizing automatic classification from the input image to the jacket. Tola et al. [17] extracted the color texture and other factors of jacket models and then analyzed the extracted features using the random forest, so as to complete the classification of clothing.

He et al. [18] used the Kansei engineering method to decompose various components of men's shirts, extracted influencing factors of styles, studied the relationship between the style and components, and achieved the style quantification of men's shirts. Ketkar [19] analyzed the modeling factors of dresses and introduced the triangle fuzzy number to fuzzy quantify the relationship between the clothing modeling factors and clothing perceptual words, so as to achieve the quantification of clothing style characteristics. Redmon and Farhadi [20] designed a fine-grained deep model and multimedia search program. First, the property vocabulary is constructed using human annotations obtained on the new fine-grained garment dataset. Then, this vocabulary is

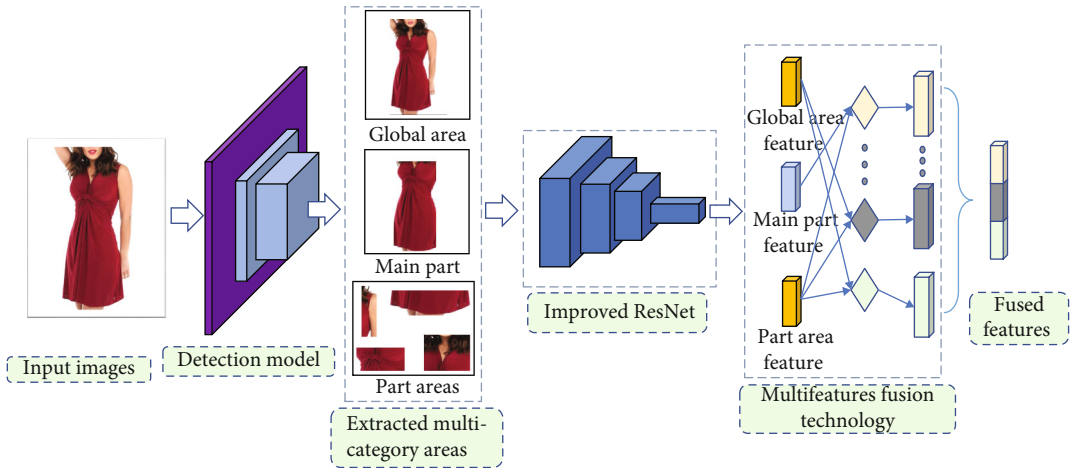


Figure 1: Multifeature extraction and fusion process. Firstly, using the improved detected method to extract the global area, main part, and part areas of the input image. Secondly, according to the results of the method, the multicategory areas is inputted to the improved residual network. And the residual network separately outputs three 128 dimensional features of global area, main part, and parts areas. Finally, the three category features are effectively fused by the multicategory feature fusion technology.

used to train a fine visual recognition system of clothing style to realize the recognition of the clothing style. Mehmood et al. [21] first found similar styles from a large database of tagged fashion images, parsed queries using these examples, and then trained the global model to implement style recognition.

However, the deep convolutional neural network is still inadequate for clothing style recognition. Khan et al. [22, 23] proposed the famous deep residual network ResNet. Compared with the traditional convolutional neural network, the deep residual network introduces a residual module into the network, which effectively alleviates the gradient disappearance of back propagation during network model training, thus solving the problems of difficult training and performance degradation in the deep network. In this paper, a kind of improved deep residual network structure and target detection model are proposed to improve the performance of recognition of clothing style.

Target detection models based on deep learning are generally divided into two categories, one is the target detection model based on candidate regions and the other is the target detection model based on the regression method [24–26].

The target detection model based on the candidate area process is divided into two steps and therefore also known as the two-phase-type (two-stage) target detection model, the first generation contains the ROI (region of interest) [27, 28]; the ROI is used to detect the target location of the candidate region and each candidate region of the generated target category of estimation and the return of border position [29]. This kind of model relies on the design of the convolutional neural network structure, but its real-time performance is poor due to the multistage characteristics.

Compared with the target detection model based on candidate regions, the target detection model based on the regression method does not need to extract the candidate box but directly completes the target border detection through convolution computation, which is called the one-stage method. In literature [30, 31], Redmon proposed the YOLOv2 model [32] and conducted BN normalization operation [33] for the input of each layer of the network. The anchor box was introduced to replace the full-connection layer, and a clustering method was used to screen the anchor box, which improved the detection accuracy of YOLO. Compared with the v2 version, YOLOv3 proposed in literature [34] has made more optimization. For example, binary cross-entropy loss [35] is used by the classification target function branch to replace the original Softmax and the underlying network with darknet-53 [36]. As a result, the detection efficiency is higher and the universality is stronger.

According to the theory of the receptive field, the deeper the convolutional layer is, the more abstract the semantics are and the local detail features of the bottom layer are blurred. Many local fine margin and shape changes become less and less obvious after multilayer convolution processing. Most of the detection models directly extract the features of the last layer of the network for analysis, which directly leads to the loss of the bottom detail features and has little impact on the accuracy of large-scale target objects, but the detection accuracy of small target objects will drop sharply [37]. Multiscale feature fusion is used to solve this problem, that is, instead of choosing the convolution output of the last layer as the feature of the image, it adopts the method of multiscale feature fusion. The above, based on deep learning and traditional feature fusion algorithms, have their own advantages in extracting the overall semantic features and specific local features of the clothing image. It is difficult to use one method alone to make the features of the clothing image more effective and comprehensive.

Therefore, we firstly propose a multicategory feature extraction model (MFEM) to extract the three category areas from an image, namely, the global areas, main parts, and part areas, meanwhile eliminating the interference factors in the

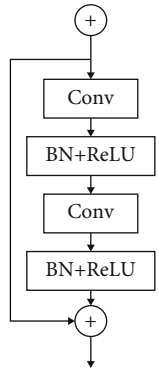


Figure 2: The sequence of the traditional residual block. The main path represents the feature diagram first through the convolution layer to BN and ReLU. The input feature diagram is not feature normalized, so the existence of the BN layer does not play a big role.

process of clothing style recognition. And then, we propose an improved ResNet model, improving the order of the batch normalization layer with the activation layer with the convolutional layer in the traditional residual block and adjusting the structure of network convolutional kernel. Finally, we designed a multifeature fusion technology to solve the problem that the single neural network cannot extract the local feature when extracting the global feature.

3. Methods

At present, the image recognition algorithm based on deep learning to extract features for the original image is widely using a single CNN [12, 17, 22]. But sometimes, the area of clothing identified is a little part of the original image and the areas that are not relevant to the identity of the clothing will have a negative impact on the recognition results. Only using a single CNN to identify the features of the clothing from the global is not comprehensive, leading to the image recognition of the clothing images not being focused on the clothing itself. In this paper, the image recognition algorithm based on improved ResNet and multifeature fusion is proposed, as shown in Figure 1. First, use the improved detected method to extract the global, main, and part areas of the image. Then, according to the results of the method, the multicategory areas are input to the improved residual network. By setting the dimension of the last layer of the residual network to 128, the residual network separately outputs three 128 dimensional features of the global, main, and parts areas of the clothing image and the three category features are effectively fused by the multicategory feature fusion technology.

3.1. Improved Residual Network. At present, most researchers choose AlexNet [38] and VGGNet to extract features for the clothing image. VGGNet has been improved on AlexNet, and the network structure is concise. The literature [39] uses VGGNet on the clothing recognition, but in the face of multiple clothing categories, the network layers of AlexNet and

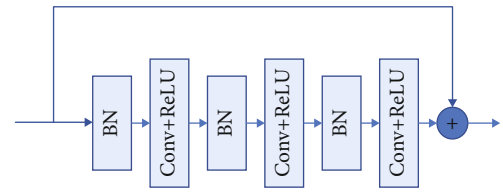


Figure 3: The sequence of the residual block in the improved residual network.

VGGNet are less, which directly affects the feature learning ability of the network.

3.1.1. Traditional Residual Network. ResNet has obtained the first place in the 2015 ImageNet Large-Scale Visual Recognition Competition [40]. The deep residual network is made up of the residual block. Each residual block can be expressed as follows:

$$y_i = h(x_i) + F(x_i, w_i), \tag{1}$$

$$x_{i+1} = f(y_i), \tag{2}$$

where the F is the residual function, f is the *ReLU* function, w_i is the power value matrix, and x_i and y_i are the input and output, respectively, of the I layer. The number h is by

$$h(x_i) = x_i. \tag{3}$$

The residual function F is defined as follows:

$$F(x_i, w_i) = w_i \cdot \sigma\left(B\left(w_i'\right) \cdot \sigma(B(x_i))\right). \tag{4}$$

$B(x_i)$ is batch normalization, "•" is convolution, and $\sigma(x) = \max(x, 0)$.

The residual units in ResNet, like the traditional CNN convolution layer, are not included in the system. Instead, the shortcut connection is introduced from the input end to the output end of each convolution layer. Using identity mapping as a shortcut connection reduces the complexity of the residual network and makes the deep network faster trained. In addition, all these shortcuts do not spread the gradient, which is the reason for the faster optimization and training of the disabled network. As the number of network layers deepens, the accuracy is not falling.

3.1.2. Improved ResNet. The weight of a certain layer of the deep convolution neural network is changed, and the output feature diagram of the layer changes, and the weight of the next layer of network needs to be studied again, and each layer of network weight will be affected. Adding activation functions to ResNet can improve the nonlinear ability of building network models. The deep residual network adopts linear modification unit ReLU [41], function $f(x) = \max(0, x)$ as activation function. The gradient of the ReLU function has been reduced by the gradient at $x0$, and the gradient dispersion is alleviated.

Table 1: Comparison of two kinds of network convolution structures.

	ResNet50	Improved ResNet
Structure	$\begin{bmatrix} C:1\times1,64 \\ C:3\times3,64 \\ C:1\times1,256 \end{bmatrix}\times3$	$\begin{bmatrix} C:1\times1,128 \\ C:3\times3,128 \\ C:1\times1,128 \end{bmatrix}\times3$
	$\begin{bmatrix} C:1\times1,128 \\ C:1\times1,128 \\ C:1\times1,512 \end{bmatrix}\times4$	$\begin{bmatrix} C:1\times1,256 \\ C:1\times1,256 \\ C:1\times1,256 \end{bmatrix}\times4$
	$\begin{bmatrix} C:1\times1,256 \\ C:1\times1,256 \\ C:1\times1,1024 \end{bmatrix}\times6$	$\begin{bmatrix} C:1\times1,512 \\ C:1\times1,512 \\ C:1\times1,512 \end{bmatrix}\times6$
	$\begin{bmatrix} C:1\times1,512 \\ C:1\times1,512 \\ C:1\times1,2048 \end{bmatrix}\times3$	$\begin{bmatrix} C:1\times1,1024 \\ C:1\times1,1024 \\ C:1\times1,1024 \end{bmatrix}\times3$

With the deepening of the convolution neural network, the speed of convergence of the network and the dispersion of the gradient are found in the course of training. This problem can be solved effectively. The specific solution is to normalize the input signal of the same layer, and the formula is as follows:

$$\widehat{x} = \frac{X - E(x)}{\sqrt{\mathrm{Var}(x) + \varepsilon}}, \tag{5}$$

where $\widehat{x}$ is the activation value of the network normalization, X is the activation value of a layer of the network, $E(x)$ is the average, $\mathrm{Var}(x)$ is the variance, and ε is the minimum. The BN algorithm formula is as follows:

$$y^{(k)} = \gamma^{(k)} x \wedge^{(k)} + \beta^{(k)}. \tag{6}$$

Each neuron x^k has a pair of γ, β. When $\gamma^{(k)} = \sqrt{\mathrm{Var}[x^{(k)}]}$, $\beta^{(k)} = E[x^{(k)}]$, the model can maintain the original learning features of a layer and can reconstruct the parameters γ, β and restore the feature distribution of the initial network learning. The BN layer is an activation method of the normalized neural network, and the algorithm of batch normalization is used to process the input signal of each layer, stabilize the distribution of the data, and set up a large learning rate in the training, so that the network converges speed and the training speed is faster. Figure 2 shows the sequence of the convolution layer with the BN layer with the ReLU layer in the traditional residual network.

The sequence of traditional residues is defective in deep convolution ResNet, such as the input of the identical blocks from two paths to the deep network. The main path represents the feature diagram first through the convolution layer to BN and ReLU. The input feature diagram is not processed first, so the existence of the BN layer does not play a big role. The method of arrangement of new residual blocks proposed in this paper is to preserve the identity of the shortcut and also maintain the learning ability of the nonlinear network path on the right, as shown in Figure 3.

Table 1 is the main structure of the convolutional layer of the original ResNet50 network and the main structure after the number of parameters has been changed. There are 3 convolution kernels, 4 convolution kernels, 4 convolution kernels, 6 convolution kernels, and 512 convolution kernels. There are 3 residual blocks containing 1024 convolution kernels and two fully connected layers. The dimensions of the model output are 8 and 10, correspond to the classification categories of the two datasets.

3.2. Extracting Multicategory Areas Based on Target Detection. In order to realize the effective extraction of the global area, main part, and parts areas, the improved target detection model is used to detect the areas. At present, in the field of target detection, there are two popular types of CNN used for feature extraction, namely, VGG and ResNet, both of which are deep network structures. ResNet is more efficient than VGG due to its efficient residual components, and the extracted image feature semantics are more abundant. Therefore, the improved ResNet is used in our model. The improved ResNet trained by simple stochastic gradient descending has fast convergence speed and the ability to use memorized information to avoid repeated computation.

First, the improved ResNet is used to extract the image features, and the region proposal network (RPN) is used to complete the recommendation of candidate boxes on the image features, and a set of candidate boxes is selected. Then, the corresponding feature area is intercepted for the candidate box, and the size of $7 \times 7 \times 512$ is inputted to the full connection layer after pooling. Finally, the classification layer and regression layer are used for target classification and border regression. Our model has been optimized in many aspects.

(1) Through our improved residual block, a total of five feature maps are generated, each of which is at a different level. Therefore, the semantic and resolution information contained vary in strength and weakness

(2) The second module is the RPN recommendation candidate box. Different from faster R-CNN, RPN of this model is a cascading structure and anchors of different scales take the feature map of the corresponding level through a selector. After the first layer RPN selects the candidate box set, the optimized non-maximum suppression (NMS) method is also used to filter the candidate box set, so as to improve the efficiency of candidate box screening

(3) For each candidate region recommended by RPN, the corresponding feature map fragment is intercepted and dimensionally reduced using the ROI Align pooling layer to form the final feature with the size of $7 \times 7 \times 512$ and the full connection layer is inputted. And the ROI Align pooling method uses bilinear

Input: $B = \{b_1, \cdots, b_2\}$, $S = \{S_1, \cdots, S_N\}$, N_t, Where B is the sequence of candidate boxes, S is the score of the candidate box, N_t is the threshold of the IOU.
Output: $D = \{d_1, \cdots, d_2\}$, $S = \{S_1, \cdots, S_k\}$, Where D is the final winning candidate box and S is the score of the output candidate box.

```
Begin:
  D ← {}
  While B! = {} do:
    m ← arg max {S}
    M ← b_m
    L ← M
    B ← B ← M
    For b_i in B
      If IOU{M, b} > N_t
        L ← L ∪ b_i
      End if
    s_i ← s_i f(IOU(M, b_i))
    End for
    M ← f_2(L)
    D ← D ∪ M
End while
```

Algorithm 1: Optimized NMS method.

Table 2: Label categories of the three level areas.

Area category	Label category
Global area	Whole body
Main part	Upper, bottom
Part areas	Collar, sleeve, skirt, trouser

interpolation to avoid precision mismatch caused by quantization

In the prediction stage, this model also makes some optimization operations in order to improve the recommendation efficiency of RPN. Firstly, an RPN module is connected after the RPN model for the refinement of the secondary border of the candidate box. In addition, an optimized NMS algorithm is introduced to suppress and screen the candidate boxes generated by the RPN in the first layer. Because there is no definite proportional relationship between the confidence of the classification result and the confidence of the rectangular box position, traditional NMS will cause many candidate boxes with different targets to be mistakenly deleted. Therefore, the NMS algorithm has been improving. For example, soft-NMS in literature [42, 43] uses a method that does not eliminate high-overlapping candidate boxes but subdivides the candidate boxes. In literature [44], soft-NMS adopts the Gaussian function weighting method to integrate high-overlapping candidate boxes and these methods have certain effects. In this paper, an optimized NMS method is obtained through integration and the pseudocode is shown as Algorithm 1.

In this approach, there are two aspects. Firstly, the soft-NMS scoring inhibition method was used and the scoring formula was shown in equation (7). Secondly, the soft-NMS weighted adjustment method is used to adjust the weight of the candidate box's optimal position coordinates according to the score. As shown in equation (8), each box whose candidate box IOU with the maximum score value is larger than the threshold value is added according to the weight of the score value to get a new box to be added to the final set of candidate boxes.

$$s_i = s_i e^{-\frac{iou(M,b_i)^2}{\sigma}}, \quad \forall b_i \notin D, \tag{7}$$

$$M'_x = \sum_i^m \frac{b_{i,\text{score}}}{\sum_j^m b_{j,\text{score}}} b_{ix}, \tag{8}$$

where b is the sequence of candidate boxes, S is the score of the candidate box, and m is the maximum value of s.

When training our model, we firstly use the annotation tool to label the three level category areas of the clothing image; the specific labeling category is shown in Table 2.

The global area, main part, and part areas are as mentioned before, where the global area is the area for removing the background and retaining the full clothing and human body. The main part is the coat or the lower part of the image, and the dress and the attachment are also part of the coat. The part area is the collar, sleeve, and other local areas. In this paper, the proposed model outputs the coordinate and category information of each area box and then extracts and generates the result according to the area box coordinates, and then, the result map is inputted to the improved residual network for feature extraction.

3.3. Multifeature Fusion. Because CNN has rich features of high-level semantic information, the fusion of features of different scales can not only retain the details of the high-level bottom but also retain the basic features of high-level semantic information. However, different fusion strategies have different effects on the test results. A more complex integration strategy will only increase the

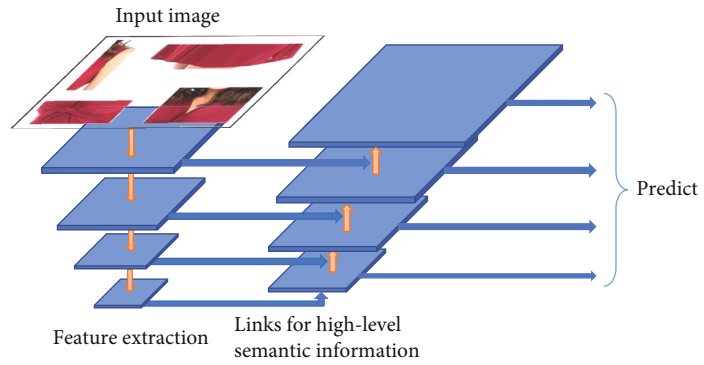


Figure 4: Structure of the feature pyramid network (FPN) model. The top-down link is used for feature extraction of the input image by the improved ResNet; a bottom-up link is used for the downward transmission of high-level semantic information.

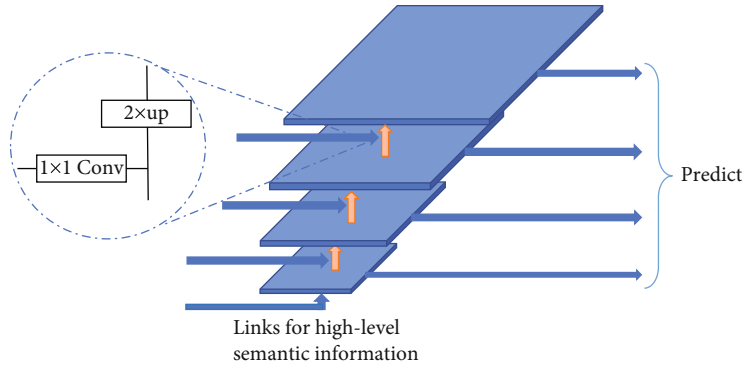


Figure 5: Internal details of the feature pyramid network (FPN) model.

Table 3: The six styles of clothing. We use the multilabel dataset to train the classification.

Attribute category	Specific label category
Sleeve of clothing	Long sleeves, short sleeves, sleeveless
Color of clothing	Pure color, color, pattern
Length of clothing	Long, ordinary, short
Type of clothing	Loose, flat, straight
Material of clothing	Cotton, hemp, cowboys, lace, mix
Collar of clothing	Circle collar, v collar, erect collar

computational complexity of the model but will have a subtle impact on the results. At present, in the target detection model based on candidate regions, the feature pyramid scheme proposed in literature [45] is a multiscale feature fusion strategy with good effect.

As shown in Figure 4, the output of each layer of the pyramid is independent and can be used as the selection of features. Such a feature formation method is also known as the feature pyramid network (FPN).

As shown in Figure 4, FPN has two links and a horizontal connection; a top-down link is used for feature extraction of the input image by the improved ResNet; a bottom-up link is used for the downward transmission of high-level semantic information; a horizontal link is used for the fusion output of features and transmitted semantic information of this layer.

As can be seen in Figure 5, there are actually three fusion links in this model. The first one is the feedforward calculation of improved ResNet, which only needs to use convolution computation to complete the feature extraction of the input image and save the features of each layer. In addition, there are two information transmission links and lateral links on both sides. As mentioned in Figure 4, the left side is the top-down information transmission link and the right lateral link. High-level semantic information is transmitted down through this link, from the third layer all the way to the first layer. The feature fusion method of the two adjacent layers is to carry out up-sampling of the upper layer features. Since the output scale of the two layers of features differs by two times in ResNet, the scale of the upper layer features can be the same as that of the lower layer features only by using deconvolution and sampling twice.

Meanwhile, the lower layer features need to be convolved through 1×1. Then, the two-layer features are added to the element to obtain the features {C1, C2, C3}. Similarly, the other two links are the right bottom-up resolution information transfer path and the left transverse connection link, from the first layer all the way to the third layer. The feature fusion method of the two adjacent layers is to pool the features of the lower layer, and the scale of the features of the upper layer can also be the same as that of the lower layer. At the same time, the upper layer features need to be convolved through 1×1. Then, the two layers of features are

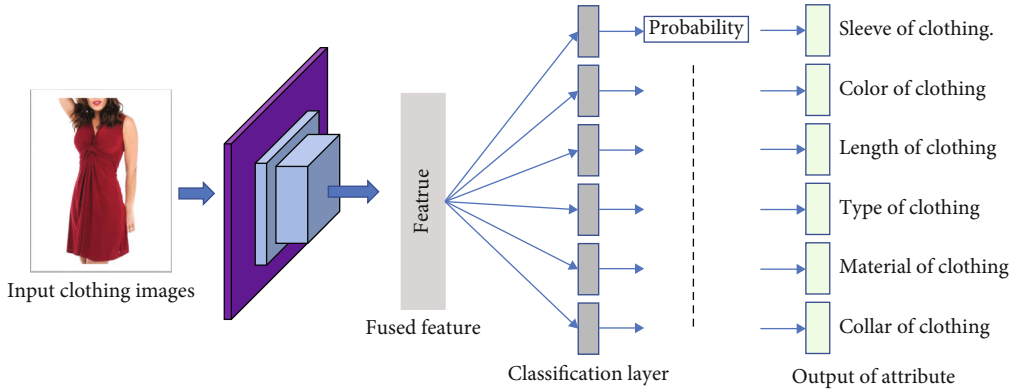


Figure 6: The clothing style classification model. Firstly, the fused feature is obtained by our model. Then, the clothing is classified by using the six Softmax classifiers in the classification layer. The number of classifiers is equal to the number of categories defined for clothing styles.

added by element to obtain features {N1, N2, N3}. Finally, the feature of the corresponding layer is added by element to obtain the final feature vector.

$$L = \left[k_0 + \log_2 \left(\frac{\sqrt{wh}}{224} \right) \right], \tag{9}$$

where 224 is the standard scale of the input of the model and the model takes it as the reference base value of the length and width of the candidate box, which represents the output of layer L that can be used. w and h are the length and width of the candidate box, where $k_0 = 4$.

Multilayer feature fusion is composed of three multiscale features, some of which are biased towards high-level semantic information and some towards low-level resolution information. For the targets with different scales, using the characteristics of different scales is more beneficial to the final result. For example, small-scale targets need rich resolution information, so they can be followed up with features that are biased towards the bottom layer. Large-scale targets are more concerned with the richness of semantic information, so they naturally tend to follow up the calculation with higher-level features.

The output is represented as L_i, 128 dimensional vectors, as the feature of the image. The features of the input multicategory area extraction are represented as $L(\text{global}), L(\text{main}), L(\text{parts})$. The fusion of the output of the system is a weighted set of 128 dimensional vectors, as shown in Figure 1. The output of the encoder contains the multicategory features of the input image, and the three multicategory areas are required to merge into the decoder. The current moment of the input image can be expressed as follows:

$$G = \sum_{i=1}^{n} \alpha_i^{(t)} L_i, \tag{10}$$

where $\alpha_i^{(t)}$ is the poutput weight of t times, $\sum_{i=0}^{n} \alpha_i^{(t)} = 1$, and $\alpha_i^{(t)}$ changes in the change of the t and dynamically adjusting the weights of different locations. And $\alpha_i^{(t)}$ is related to the visual weight of the input of the t moment and the information before the t. $\alpha_i^{(t)}$ update mechanism can be expressed as follows:

$$\begin{aligned} \beta_i^{(t)} &= w^T \varphi\left(W_h h_{t-1} + W_f f_i + b \right), \\ \alpha_i^{(t)} &= \frac{\beta_i^{(t)}}{\sum_{j=1}^{n+1} \beta_j^{(t)}}. \end{aligned} \tag{11}$$

f_i is a subset vector for I, $f_i \in \{G, L_1, L_2, \cdots, L_n\}$ and $\beta_i^{(t)}$ indicates that the corresponding visual vector f_i is weighted under the weight relative to the corresponding score weight that has been produced before. h_{t-1} is the output of a hidden layer; w, W_h, W_f and b are the weighted variables that need to be learned; $\varphi(\cdot)$ is the activation function.

3.4. Clothing Style Recognition. CNN is usually used for single label classification, and the image is the most difficult image category, with a large number of clothing features, except for the rich visual information. In the classification problem of clothing style properties, each image is represented by multiple labels, so single label learning does not apply. In this paper, the paper uses multilabel learning to conduct the classification training of clothing style properties for the improved ResNet and classifies each type of attribute and gets the model of the image classification of the clothing after training. Effectively, to solve the problems of the correlation between the different types of properties and the ability to directly put these properties in the same class set, this article defines multiple general clothing style attributes and several specific category tags, as shown in Table 3.

Based on the above definition, this paper designs the model of the clothing attribute multilabel classification, as shown in Figure 6:

The input image is first obtained by the multicategory deep features by improved detection and the improved ResNet, and then, the properties of several Softmax classifiers in the clothing style classification layer are calculated, and the number of classifiers is equal to the number of category properties of the dress style. The number of neurons in each

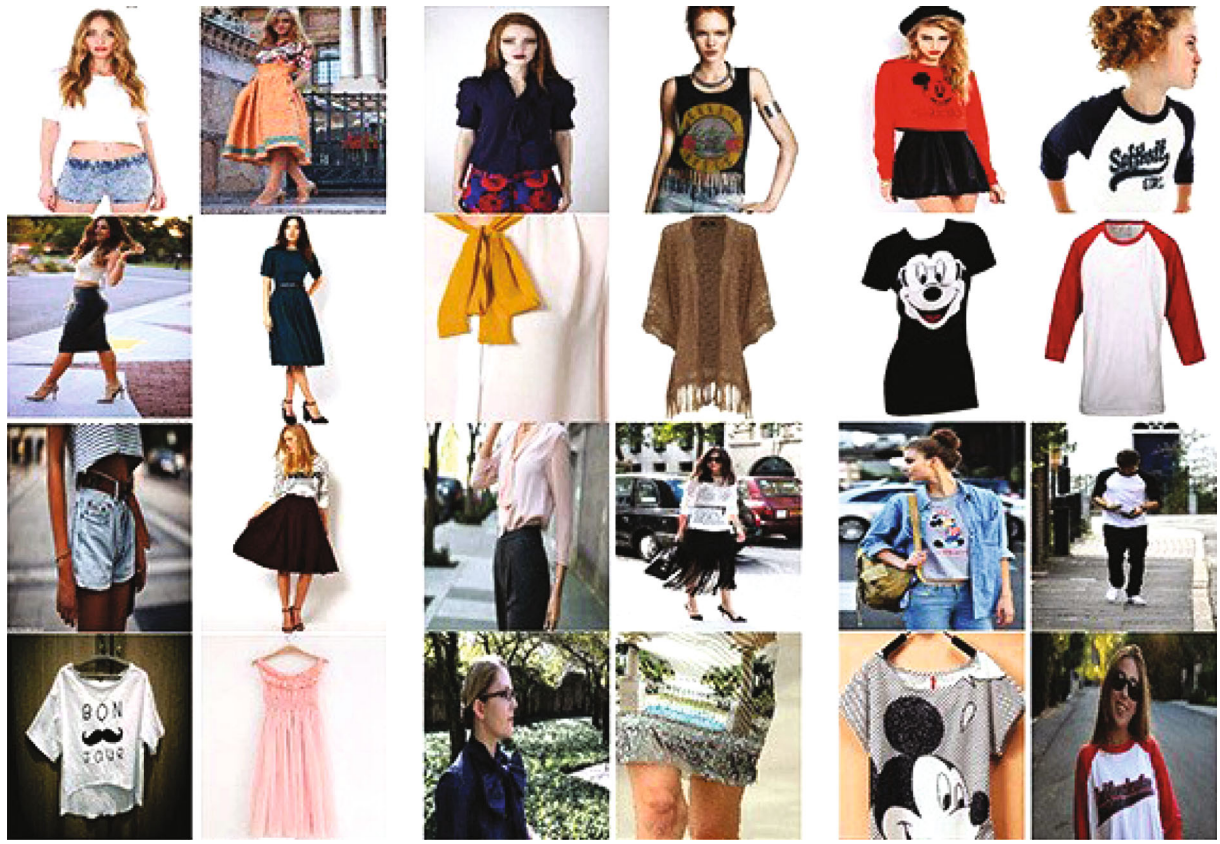


Figure 7: Sample images from the DeepFashion dataset.

classifier is equal to the number of specific labels for the clothing styles corresponding to the classifier.

3.5. Training of Model. Firstly, we randomly selected 100 styles of clothing images from the training dataset, each with a unique ID. Secondly, 3 images (a triplet) are randomly selected for each style of clothing, a total of 300 images. We only choose 3 pictures because the number of images of some clothing styles is relatively small. Then, the model with random initialization parameters is used to extract the features of each image. The retained information of each image has three categories: path, ID, and feature vector. Finally, twice loops are used for each image under each ID to select the matching positive and negative samples from the remaining 299 images according to equation (1) for training our model.

A triple consists of x_i^a (anchor), x_i^p (positive), and x_i^n (negative). x_i^a and x_i^p are the same style, while x_i^a and x_i^n are different styles. In triples, the Euclidian distance between the x_i^a and x_i^p plus the threshold should be greater than the Euclidian distance between the x_i^a and x_i^n. We use the selected triples to train the proposed improved ResNet model and then reselect the triples with the new parametric model.

$$\left\|\mathrm{Net}(x_i^a) - \mathrm{Net}\left(x_i^p\right)\right\|_2^2 + thre > \left\|\mathrm{Net}(x_i^a) - \mathrm{Net}(x_i^n)\right\|_2^2, \quad (12)$$

where i represents the i-th triple. There is the threshold value. By trying different thresholds, the triplet similarity measurement is learned. It is found that the best effect is when the global area, main part, and part area branches are set at 0.2, 0.18, and 0.15, respectively. Net $(\bullet)$ represents the feature vector extracted from the proposed model.

When using triples for training, the feature vectors Net(x_i^a), Net(x_i^p), and Net(x_i^n)) of the three samples are inputted into the triplet loss function. If it is not equal to equation (12), the parameters of the model will not be changed; otherwise, it will be calculated according to equation (13) of the loss function:

$$L = \left\|\mathrm{Net}(x_i^a) - \mathrm{Net}\left(x_i^p\right)\right\|_2^2 + thre - \left\|\mathrm{Net}(x_i^a) - \mathrm{Net}(x_i^n)\right\|_2^2. \quad (13)$$

Obtain the loss L of the model, and then, adjust the parameters of the model. The proposed model trained on the triplet similarity measure can reduce the feature distance of the same clothing image, increase the feature distance of different clothing images, and further improve the recognition ability.

4. Experiments

In this section, we will demonstrate the benefits of our approach. We start with an introduction to the dataset and then present our experimental results with performance comparison to several state-of-the-arts on the public

Short sleeves	Long sleeves	Sleeveless
Pure color	Pure color	Pure color
Ordinary	Ordinary	Long
Flat	Loose	Straight
Cotton	Hemp	Lace
Circle collar	V collar	Circle collar

Long sleeves	Sleeveless	Short sleeves
Pure color	Pattern	Pure color
Ordinary	Long	Ordinary
Loose	Straight	Flat
Mix	Cotton	Cotton
Erect collar	Circle collar	Circle collar

Figure 8: Image classification results on datasets. The black font represents the result of correct recognition, and the bold font represents the result of incorrect recognition. The first row indicates input images. The next table represents the predicted results of every category. The first row to the last row in the table represent the sleeve, color, length, type, material, and collar of the input clothing. It can be seen from the experimental results that our model has a good recognition effect for different types of clothes. For example, the second images are predicted wrongly because the hair covers the collar and the model incorrectly recognizes it as a V collar.

datasets, DeepFashion and FashionMNIST datasets. Finally, the scalability and effectiveness of our method are verified on the datasets. And the experimental device is a GTX 2080 GPU and 32 GB of RAM.

4.1. Datasets

4.1.1. DeepFashion. DeepFashion is a large-scale dataset opened by The Chinese University of Hong Kong. It contains 800000 pictures, including different angles, different scenes, and buyer shows. There are a total of four main tasks, namely, clothing category and attribute prediction, in-shop and C2S clothing retrieval, key points, and external rectangular box detection. Each image also has a wealth of annotation information, including categories, attributes, feature points, and other information. Figure 7 shows some sample images from the DeepFashion dataset.

4.1.2. FashionMNIST. The FashionMNIST dataset contains 10 categories of images, namely, T-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The training data set contains 6000 samples for each category, and the test data set contains 1000 samples for each category. There are altogether 10 categories.

4.2. Evaluation Metrics. In this article, we use mean average precision (mAP) as the measurement standard of the algorithm. mAP is the average on the basis of AP. The formula for calculating mAP is shown in equation (14).

$$\text{mAP} = \frac{1}{|Q_R|}\sum_{q\in Q_R} AP(q) \tag{14}$$

In equation (14), q means a query, which is the image to be retrieved, Q_R means the entire image collection, and $AP(q)$ means the average accuracy rate. In simple terms, AP is to calculate the average accuracy of a query image and mAP is to take the average of the accuracy of all query images. The ordering of target images in the search results is also within the consideration of mAP.

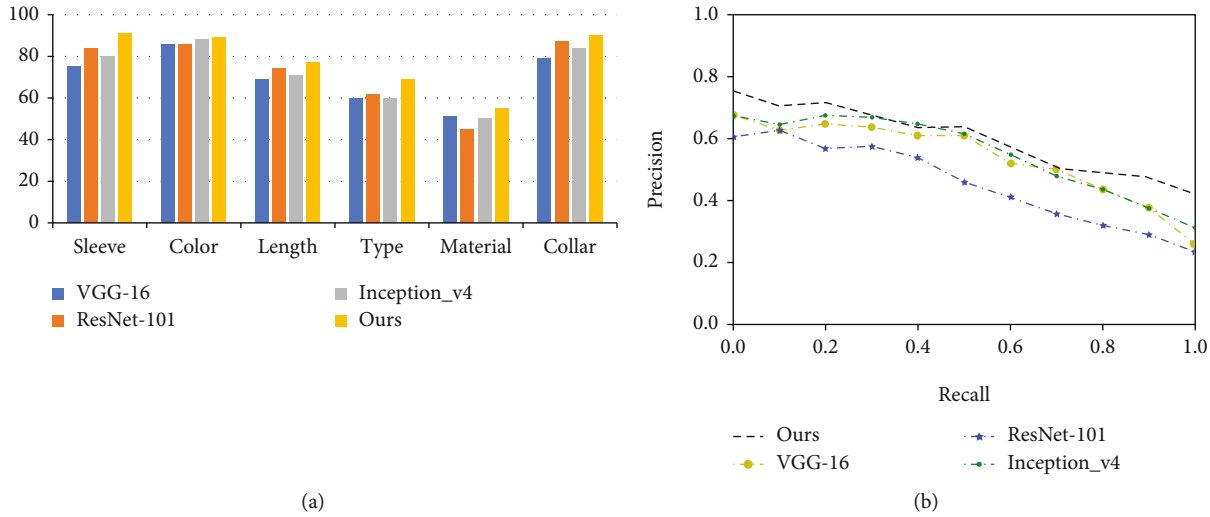
Although mAP is a statistical evaluation of the proportion of correct search results, there is a lack of evaluation of the location information of the search results. This paper uses the PR curve as the evaluation of the location information of the retrieval results. P in the PR curve represents precision, and R represents recall (recall rate).

$$\begin{aligned}\text{precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}},\\ \text{recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}.\end{aligned} \tag{15}$$

Among them, the positive examples are correctly classified as positive examples, denoted as TP (true positive), and the positive examples are incorrectly classified as negative examples, denoted as FN (false negative). Negative cases are correctly classified as negative examples, denoted as TN (true negative), and negative examples are incorrectly classified as positive examples, denoted as FP (false positive).

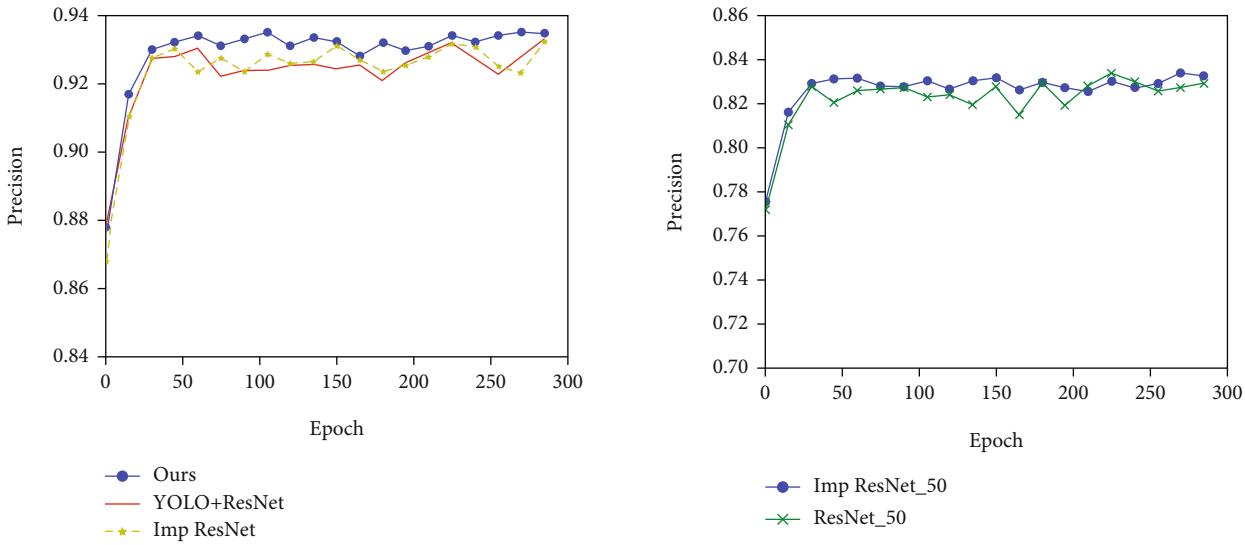
4.3. Experiment of the Proposed Method for Clothing Style Recognition. To demonstrate the scalability and effectiveness of our approach, we tested it on large-scale DeepFashion and FashionMNIST datasets. Both of these datasets are composed of a large number of clothing images, which include people with noiseless background or not. At the same time, the people have different postures. This experiment mainly reflects the classification effect of the clothing jacket. We set the number of neurons in the classification layer as 17 and h in the latent layer as 156. Then, we fine tune our network with the entire dataset. After 10000 training iterations, our proposed method achieved very high accuracy in 17 categories of clothing classification tasks (obtained by the last layer).

As shown in Figure 8, although the background of the clothing image is background free or noisy, the method presented in this paper shows good classification performance with or without people. The recognition effect of six clothes is shown in Figure 8. These 6 pieces of clothing include short sleeves, shirts, dresses, and cardigans, which, respectively, represent different genders and styles of clothing. The table under each photo shows six attributes of the clothing, including the sleeve, color, length, type, material, and collar. The



(a) (b)

Figure 9: (a) is the precision of the classification of the clothing image style and (b) represents the PR curve. As shown in (a), although the different network models are given different rates, however, the classification accuracy of the sleeves, collars, and patterns of the six types of attributes is higher and the classification accuracy of the three kinds of clothing is low. This is due to the easy distinction between the sleeve length, the collar, and the three kinds of species and the length, the plate, and the clothing are more difficult to distinguish. It can also be seen in (b) that our model has better performance than the other three models.



(a) The final model (ours) is compared with two other improved models (YOLO +ResNet: YOLOv3+ResNet_101; Imp ResNet: improved ResNet_101 without YOLOv3)

(b) The improved ResNet_50 compared with ResNet_50

Figure 10: The final model (ours) is compared with other models. As shown in (a), it can be found that no matter whether adding a YOLO model to the traditional residual network or just using improved ResNet, the best experimental results have been obtained with our proposed models. As shown in (b), the improved residual network of this paper is compared with the traditional residual network and the mAP of our model is better than that of the traditional residual network. When the epoch was around 140 and 200, MAP dropped sharply, forming two valleys and peaks.

black font represents the result of correct recognition, and the red font represents the result of incorrect recognition. For example, the second images are predicted wrongly because the hair covers the collar and the model incorrectly recognizes it as a V collar. Please note that some of the images are predicted wrongly because products can be ambiguous between certain categories. For example, as shown in Figure 8, the looseness of white short sleeves is difficult to distinguish.

4.4. Comparison of the Proposed Method with Other Methods on DeepFashion. In the course of the classification of clothing styles, 28500 images were selected from the training center, with 23000 images as a training set, 5500 images as a test

Table 4: The results of different networks.

Network model	Accuracy rate (%)	Training time (h)
VGG16	89.76	44
Inception_v3	91.06	39
ResNet_101	93.26	81
Ours	94.97	83

set, in order to find the suitable network model for the classification of clothing images, selecting the model of VGG-16, ResNet-101, Inception-v4, and ours for the comparison of the classification of clothing styles. Using the training parameters on the ImageNet to initialize each network, the classification layer parameters are randomly initialized by Gaussian distribution and then training the network using the training set.

Finally, Figure 9 shows the precision of the test set in four different networks. As Figure 9 reveals, although the different network models are given different rates, however, the classification accuracy of the sleeves, collars, and patterns of the six types of attributes is higher and the classification accuracy of the three kinds of clothing is low. This is due to the easy distinction between the sleeve length, the collar, and the three kinds of species and the length, the plate, and the clothing are more difficult to distinguish. For the length of the dress, the classification accuracy of the dress is low because of the influence of the fashion style and the height of the model. For the type, due to the angle of shooting and the position of the model, the classification of the type is not high. And clothing is harder to distinguish. The results are common logic, and in the four network models, the overall performance of the network model is the best.

4.5. Comparison of the Proposed Method with Other Methods on FashionMNIST. Standard dataset FashionMNIST has 70000 images from 10 different categories of goods. There are 60000 images as the training set and 10000 images as test set validation. The image size of the dataset is consistent with the MNIST dataset. As shown in Figure 10(a), the recognition algorithm presented in this paper is better than the other two improved networks and the ability to be more powerful in mAP and convergence. It can be found in Figure 10(a) that no matter whether it is adding a YOLO model to the traditional residual network or just using improved ResNet, the best experimental results cannot be achieved. Although our model is slightly worse than the other two models with an epoch of 170, our model is better than the other two models overall. However, the overall performance of YOLO+RseNet and Imp ResNet is basically the same. As shown in Figure 10(b), the improved residual network of this paper is compared with the traditional residual network and the mAP of our model is better than that of the traditional residual network. When the epoch was around 140 and 200, MAP dropped sharply, forming two valleys and peaks. Therefore, in conclusion, it can be found that our model has better and more stable performance in these two groups of comparative experiments. Therefore, in conclusion, our model has better and more stable performance.

As shown in Table 4, the network models are fully trained to identify accuracy and training time. The use of CNN has not resulted in the difficulty of training in the network, the inception_v3 layer is more than VGG16, but inception_v3 is more accurate and effective for image recognition classification, and the accuracy of the improved depth of the network is more accurate than VGG16. In the case of our improved ResNet, the precision is 1.32% better than the traditional residual network precision. The network precision is combined with the module, which improves the accuracy of the network by 2.21%. After the two method junctions, the final model is 0.95% more than the traditional residual network precision. The results show that the proposed model can improve the characteristic learning ability of the convolution neural network. Table 4 shows that using our method to solve the problem of clothing image recognition is very effective. The different experimental results of our method compared with other methods prove that our method of using improve ResNet has significant advantage on image recognition.

5. Conclusions

In this paper, we presented a new method for clothing style recognition, which is based on the target detection and multideep feature fusion. It first introduces and implements the improved target detection model to extract multicategory areas and the improved ResNet to extract deep features. Lastly, by feature pyramid network, the shape, soft-NMS, and multideep features fusion technology, the three multideep features are greatly fused together. In the end, an accurate and fast clothing style recognition of clothing style was achieved. By comparing the experimental results and the evaluation of recognition performance, it can be seen that the proposed algorithm has not only good efficiency but also excellent robustness in the clothing style recognition. Since the method in this paper needs to recognize every detail of clothing, the recognition rate of the proposed method will be greatly reduced if the clothing image is severely occluded. During the experiment, we found that the shirt would cover the pants in most cases, so the recognition rate was not high; so, our method did not support the multicategory recognition of pants for the time being. In the future work, we will further study the solution to this problem.

Data Availability

The DeepFashion data used to support the findings of this study have been deposited in the Google Drive or Baidu Drive repository (http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html). The FashionMNIST data used to support the findings of this study have been deposited in the Github repository (https://github.com/zalandoresearch/fashion-mnist).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is jointly supported by the National Natural Science Foundation of China (62072414 and U1504608), and the Key Scientific and Technological Project of Henan Province (212102210540, 192102210294, and 202102210383), and the Key Scientific Research Projects of Henan Higher School (20B520039).

References

[1] Y. Wang and S. Zhi-Feng, "Clothing image classification and retrieval based on metric learning," *Computer Applications and Software*, vol. 34, pp. 255–259, 2017.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, , San Diego, CA, USA, 2015.

[3] F. Afza, M. A. Khan, M. Sharif et al., "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing*, vol. 106, 2021.

[4] H. Arshad, M. A. Khan, M. I. Sharif et al., "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, vol. 27, 2020.

[5] M. A. Khan, Y. D. Zhang, S. A. Khan, M. Attique, A. Rehman, and S. Seo, "A resource conscious human action recognition framework using 26-layered deep convolutional neural network," *Multimedia Tools and Applications*, vol. 80, 2020.

[6] N. Naheed, M. Shaheen, S. A. Khan, M. Alawairdhi, and M. A. Khan, "Importance of features selection, attributes selection, challenges and future directions for medical imaging data :a review," *Computer Modeling in Engineering & Sciences*, vol. 125, no. 1, pp. 315–344, 2020.

[7] M. Rashid, M. A. Khan, M. Alhaisoni et al., "A Sustainable Deep Learning Framework for Object Recognition Using Multi-Layers Deep Features Fusion and Selection," *Sustainability*, vol. 12, no. 12, p. 5037, 2020.

[8] Z. He, Y. Li, L. Deng, P. Li, X. Shi, and X. Han, "A new two-stage image retrieval algorithm with convolutional neural network," in *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, pp. 98–102, Luoyang, Henan, China, 2019.

[9] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, 2016.

[11] D. Yarotsky, "Error bounds for approximations with deep ReLu networks," *Neural Networks*, vol. 94, pp. 103–114, 2017.

[12] D. G. Lowe, "Distinctive image features from scale-invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, Boston, MA, USA, 2015.

[14] Y. Zheng, S. Wu, D. Liu, R. Wei, S. Li, and Z. Tu, "Sleeper defect detection based on improved YOLO V3 algorithm," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 955–960, Kristiansand, Norway, 2020.

[15] X. Yang and L. J. Latecki, "Affinity learning on a tensor product graph with applications to shape and image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*, pp. 2369–2376, Colorado Springs, CO, USA, 2011.

[16] H. Noh, A. Araujo, and J. Sim, "Large-scale image retrieval with attentive deep local features," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3476–3485, Venice, Italy, 2017.

[17] E. Tola, V. Lepetit, and P. Fua, "Daisy: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[18] Y.-F. He, L. Zhou, J.-Q. Yu, T. Xu, and T. Guan, "Image retrieval based on locally features aggregating," *Chinese Journal of Computers*, vol. 34, no. 11, pp. 2224–2233, 2011.

[19] N. Ketkar, "Convolutional neural networks," in *Deep Learning with Python*, pp. 63–78, Springer, 2017.

[20] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, http://arxiv.org/abs/1804.02767.

[21] A. Mehmood, M. A. Khan, M. Sharif et al., "Prosperous human gait recognition: an end-to-end system based on pre-trained CNN features selection," *Multimedia Tools and Applications*, vol. 80, 2020.

[22] N. Hussain, M. A. Khan, M. Sharif et al., "A deep neural network and classical features based scheme for objects recognition: an application for machine inspection," *Multimedia Tools and Applications*, vol. 80, 2020.

[23] M. A. Khan, K. Javed, and T. Saba, "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimedia Tools and Applications*, vol. 80, 2020.

[24] K. Yamaguchi, M. H. Kiapour, and T. L. Berg, "Paper doll parsing: retrieving similar styles to parse clothing items," in *2013 IEEE International Conference on Computer Vision*, pp. 3519–3526, Sydney, NSW, Australia, 2013.

[25] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception - v4, inceptionresnet and the impact of residual connections on learning," 2016, http://arxiv.org/abs/1602.07261.

[26] X. Wang, T. Zhang, D. R. Tretter, and Q. Lin, "Personal clothing retrieval on photo collections by color and attributes," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2035–2045, 2013.

[27] A. Sarwar, Z. Mehmood, T. Saba, K. A. Qazi, A. Adnan, and H. Jamal, "A novel method for content-based image retrieval to improve the effectiveness of the bag-of-words model using a support vector machine," *Journal of Information Science*, vol. 45, no. 1, pp. 117–135, 2019.

[28] H. J. Kim, E. Dunn, and J.-M. Frahm, "Predicting good features for image geo-localization using per-bundle VLAD," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1170–1178, Santiago, Chile, 2015.

[29] Y. Li, H. Lei, S. Lin, and G. Luo, "A new sketch-based 3D model retrieval method by using composite features," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2921–2944, 2018.

[30] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 36–45, Boston, MA, USA, 2015.

[31] X. Han, Y. Li, Q. Zheng et al., "A Multiple Feature Fusion Based Image Retrieval Algorithm," in *Proceedings of 2019 the 8th International Conference on Networks, Communication and Computing*, pp. 104–109, Luoyang, Henan, China, 2019.

[32] L. Wei, S. Zhang, and H. Yao, "GLAD: global-local-alignment descriptor for pedestrian retrieval," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 420–428, Buenos Aires, Argentina, 2017.

[33] D. Mishkin, F. Radenovic, and J. Matas, "Repeatability is not enough: learning affine regions via discriminability," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 284–300, Munich, Germany, 2018.

[34] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, 2007.

[35] S. Gammeter, "I know what you did last summer: object-level auto-annotation of holiday snaps," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 614–621, Kyoto, Japan, 2009.

[36] S. S. Husain and M. Bober, "REMAP: multi-layer entropy-guided pooling of dense CNN features for image retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5201–5213, 2019.

[37] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2019.

[38] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting Oxford and Paris: large-scale image retrieval benchmarking," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5706–5715, Salt Lake City, UT, USA, 2018.

[39] B. Cao, J. Zhao, P. Yang, P. Yang, X. Liu, and Y. Zhang, "3-D deployment optimization for heterogeneous wireless directional sensor networks on smart city," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1798–1808, 2019.

[40] I. Jung, K. You, H. Noh et al., "Real-time object tracking via meta-learning: Efficient model adaptation and one-shot channel pruning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11205–11212, Venice, Italy, 2020.

[41] B. Cao, J. Zhao, P. Yang et al., "Multiobjective 3-D topology optimization of next-generation wireless data center network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3597–3605, 2020.

[42] Y. Hou, H. Zhang, and S. Zhou, "Evaluation of object proposals and ConvNet features for landmark-based visual place recognition," *Journal of Intelligent and Robotic Systems*, vol. 92, no. 3-4, pp. 505–520, 2018.

[43] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 754–766, 2010.

[44] H. Lu, L.-P. Nolte, and M. Reyes, "Interest points localization for brain image using landmark-annotated atlas," *International Journal of Imaging Systems & Technology*, vol. 22, no. 2, pp. 145–152, 2012.

[45] A. Mikulik, M. Perdoch, O. Chum, and J. Matas, "Learning vocabularies over a fine quantization," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 163–175, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8868781, 9 pages
https://doi.org/10.1155/2021/8868781

Research Article

Reconstruction of Generative Adversarial Networks in Cross Modal Image Generation with Canonical Polyadic Decomposition

Ruixin Ma, Junying Lou, Peng Li, and Jing Gao

School of Software, Dalian University of Technology, 116024, China

Correspondence should be addressed to Ruixin Ma; maruixin@dlut.edu.cn

Received 8 September 2020; Revised 10 October 2020; Accepted 8 March 2021; Published 9 April 2021

Academic Editor: Xiaojie Wang

Generating pictures from text is an interesting, classic, and challenging task. Benefited from the development of generative adversarial networks (GAN), the generation quality of this task has been greatly improved. Many excellent cross modal GAN models have been put forward. These models add extensive layers and constraints to get impressive generation pictures. However, complexity and computation of existing cross modal GANs are too high to be deployed in mobile terminal. To solve this problem, this paper designs a compact cross modal GAN based on canonical polyadic decomposition. We replace an original convolution layer with three small convolution layers and use an autoencoder to stabilize and speed up training. The experimental results show that our model achieves 20% times of compression in both parameters and FLOPs without loss of quality on generated images.

1. Introduction

Generating images according to the corresponding text is an important, challenging, and interesting task in computer vision. Compared with text, images are direct and easy to understand. Cross modal image generation attracts many researchers due to its great potential and value in application of computer vision, such as cross modal search, art creation, and image editing. It is conducive to reducing storage space and operating cost. Generating synthetic images from text application in art creation and criminal image calls for fast reaction and compact models. For illustrations for stories and painting for album covers, compact cross modal image generation models can instantly visualize thoughts in the mind by a few descriptive sentences. Text-to-image GANs can activate visualization application so as to promote artistic creation greatly.

In the past few years, most of generative models have applied the Markov chain learning mechanism, Monte Carlo estimation, and sequence data to learn joint distribution. These models involve too much computation and are not suitable for large-scale image generation. The Variational Autoencoder (VAE), Recurrent Neural Network (RNN), and Convolutional Neural Networks (CNN) are used to generate natural pictures according to a conditional distribution [1–3]. These models can generate pictures only by labels or feature information generated by other networks. However, images generated by these models were unreal. Driven by the proposal of the generative model of GANs, images generated from text tasks got a significant development. Reed et al. [4] firstly applied GAN to synthesize impressive and compelling pictures from character level to pixel level. More and more researchers have committed to improving the quality of generated images by adding modules and constraints. Many excellent models have been proposed, such as StackGAN++ [5], AttnGAN [6], and HDGAN [7]. These models can generate high-pixel pictures. But existing text-to-image GANs are so complex that it is hard to deploy them on the mobile end.

Low computation and response in real time are critical for cross modal search and criminal image generating tasks. With the emerging of 5G technology [8–12], the demand for mobile terminal deployment is increasing. However, existing text-to-image GAN models have a large number of

parameters and huge computation for low-end devices within the Internet of things. In order to compress and speed up text-to-image GAN, we propose a compact architecture based on canonical polyadic decomposition.

Rank decomposition has been widely applied in model compression and acceleration. Rank decomposition can represent a complex matrix as multiplication of small submatrices. It means that a few submatrices can be used to reconstruct the weight matrix. These submatrices maintain important properties of the matrix. For cross modal-generated image task, there are too many parameters and high computation in existing models. Therefore, we can use rank decomposition to reduce parameters and computation. There are two methods to apply rank decomposition: decomposing the complex matrix and replacing [13–16] and designing low-rank separable network structures [17, 18]. Canonical polyadic decomposition is an efficient and standard rank decomposition method. It has been effectively applied to compress and accelerate networks [13, 15]. So we use CP decomposition to compress text-to-image GAN.

There are three problems to decompose the complex model. First, implement rank decomposition in the original model because decomposition operations involve high computational cost. Second, text-to-image GANs are more complex than CNN. Because the training process of GAN is using zero-sum two-person game to learn the distribution of real data, the training is unstable and the decomposed model is not easy to converge. The third problem is that cross modal image generation applications have high requirements on the authenticity, clarity, diversity, and resolution of the generated images. It is hard to compress the model as much as possible under the premise of ensuring the image quality.

To solve the first problem, we use CP decomposition to reconstruct text-to-image GAN. It reduces a large number of redundant parameters and decomposition operation cost. Then, we use autoencoders to pretrain to stabilize the decomposed model. For the last problem, we explored a large number of the experiments to find the appropriate rank to guarantee the generated pictures' quality. Experimental results on representative cross modal image generation datasets show that our scheme can efficiently reduce computation complexity by CP decomposition. More importantly, our model is slightly better than the original model in FID and achieves 20% compression in FLOPs and parameters.

The contributions of this paper can be summarized as follows:

(i) To the best of our knowledge, this is the first paper to use CP decomposition to reconstruct cross modal GAN

(ii) We design a compact text-to-image GAN based on CP decomposition and use autoencoders to pretrain, reducing high computational cost

The rest of the paper is organized as follows: Section 2 presents the preliminaries related to this paper. In Section 3, the reconstruction process of compact cross modal GAN architecture is illustrated. Section 4 evaluates our proposed compact model, and Section 5 summarizes our work.

2. Related Work

The aim of this paper is to reconstruct a compact architecture for text-to-image GAN from scratch. In this section, we present the relevant research in text-to-image GAN and compressing deep neural networks by rank decomposition.

2.1. GAN in Cross Modal Image Generation. The text-to-image task extracts features from human-written descriptions to generate images, which turn low-dimensional and low-rank data into comparatively high-dimensional pictures. It is challenging to use GAN to generate high-resolution images according to text because of GAN's training instability. Reed et al. [4] first successfully used GAN to generate 64×64 high-quality images by modifying DCGAN; then, they put forward GAWWN [19] to generate high-quality 128×128 images by using text description and object location as conditions.

StackGAN [20] used stacked conditional GAN to generate 256×256 pictures for the first time. In subsequent work, StackGAN++ [5] used tree structure and multiple generators to generate images of different scales. In addition to conditional loss, it introduced unconditional loss and colour regulation. These additional conditions improved stability of the training process and quality of generated images. The third work of the team was to introduce the attention mechanism [6], which synthesized fine-grained details of different subareas of images by focusing on the relevant words in the natural language description. It was the first time to indicate that layered attention GAN can automatically select word level conditions to generate different parts of images. TAC-GAN [21] also used condition GAN to synthesize 128×128 resolution images with text. Compared with StackGAN [20], its inception score had improved by 7.8%, but resolution was not as high as that of StackGAN. Johnson et al. [22] proposed to use a scene map as an intermediate medium to generate pictures. The model of Johnson et al. [22] solved the outstanding problem of StackGAN which could not deal with complex text. HDGAN [7] designed a pyramid hierarchy structure to solve the problem that images do not match the text in StackGAN [20].

ObjGAN [23] could generate complex scenes according to text. This paper solved the problem of how to make AI understand the relationship between multiple objects in the scene. The generator in ObjGAN could use fine-grained words and object-level information to gradually refine synthetic images. StoryGAN [24] could draw stories based on the sequence condition of the GAN framework. Given a multisentence paragraph, StoryGAN could generate a series of images and each image corresponded to a sentence, completely visualizing the whole story. In order to get vivid generated images, the network has been getting deep and complex. Existing models are hard to be deployed on the mobile end. Therefore, it is necessary to compress these models.

2.2. Rank Decomposition. Rank decomposition is to extract important features of a matrix, such as Singular Value Decomposition (SVD), canonical polyadic decomposition

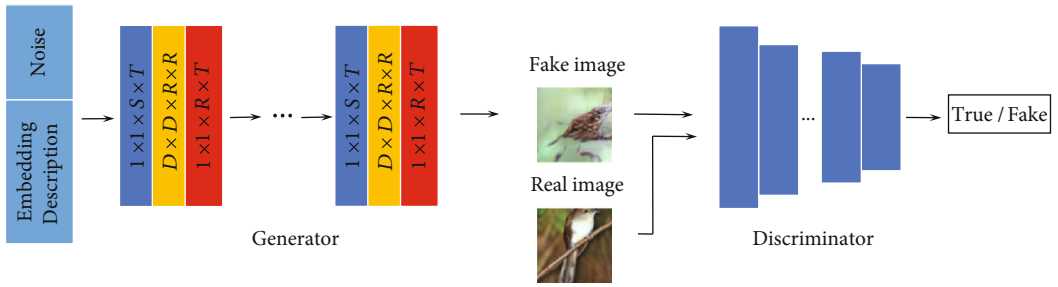


Figure 1: Overview of our cross modal image generation model based on CP decomposition.

(CP decomposition), Tucker decomposition, and tensor train decomposition (TT decomposition). It reduces redundant parameters using small and simple submatrices to represent a complex matrix. Tucker decomposition has a core tensor. Compared with Tucker, CP decomposition is a special Tucker decomposition, which is simpler and more efficient for compressing parameters. TT decomposition is suitable for sequence data and model. Therefore, this paper uses CP decomposition to compress the model.

Rigamonti et al. [25] used SVD and CPD to get a couple of separable filters to approximate an original convolution layer. It proved validity of separable convolution. Thus, many researchers paid attention to using low-rank decomposition to accelerate network. Some decomposed pretrained networks by tensor decomposition and then replaced by the original network layer [13–16, 26–29]. Some directly designed low-rank separable network structures [17, 18, 30, 31]. Lin et al. [16] decomposed CNN by GSVD and used backpropagation to decrease global reconstruction error. Based on Lin et al. [16] which only performed spatial decomposition, Jaderberg et al. [14] explored both cross channel and spatial decomposition. Then, Denton et al. [13] and Lebedev et al. [15] used CP decomposition to compress and speeded up CNN. Novikov et al. [31] used TT decomposition to compress the model. Based on the separability of convolution, compact networks were designed and trained from scratch [17, 18, 32].

It is feasible and necessary to compress models. There are a few works to compress GANs [33, 34]. Li et al. [33] and Shu et al. [34] used a pretrained network to prune to compress the model. Due to extra high computational cost of decomposing a pretrained network, we design a compact network architecture. It is the first time to use CP decomposition for text-to-image GANs. We train a compact model from scratch so as to reduce cost of decomposition computation. The reconstructed model overcomes unstable training of GANs as the model deepens. Finally, the reconstructed model achieves 20% compression while ensuring the quality of generation.

3. Method

The architecture of our model is shown in Figure 1. Description embedding is concatenated to a noise vector, and then, it is fed forward through the decomposed generator G. Generated images and real images coupled with description embedding are fed to discriminator D. During the training, D learns

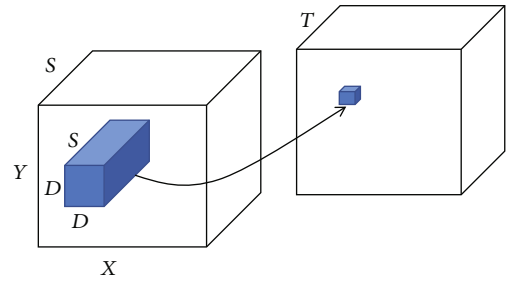


Figure 2: Original convolution layer. The filter is a tensor of size $D \times D \times S \times T$.

to distinguish whether pictures are real pictures and pair up with text. Overall, our method has three steps: the first step is to take a convolutional layer and reconstruct it using CP decomposition, the second step is to pretrain a decomposed network layer by layer, and the third step is to select an appropriate learning rate and train the network using backpropagation.

3.1. Canonical Polyadic Decomposition. Canonical polyadic decomposition was proposed by Hitchcock in 1927 [35]. An N-order tensor can be decomposed into a sum of a finite number of rank-one tensors. The finite number of components is the tensor rank R. For example, a second-order kernel tensor $\mathcal{X} \in \mathbb{N}^{i \times j}$ with rank R is given by the following form:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r, \tag{1}$$

where $\circ$ is the vector outer product, $a \in \mathbb{N}^R$, and $b \in \mathbb{N}^R$. GAN consisted of a discriminator and a generator generally. The discriminator and generator in GAN-int-cls are convolutional neural networks. The most time-consuming operation in CNNs is convolution, which maps an input tensor $\mathcal{X}(i, j, s)$ of size $X \times Y \times S$ into an output tensor $\mathcal{Y}(x, y, t)$ of size $(X - D + 1) \times (Y - D + 1) \times T$. The convolution can be represented as

$$\mathcal{Y}(x, y, t) = \sum_{i=x-\delta}^{x+\delta} \sum_{j=y-\delta}^{y+\delta} \sum_{s=1}^{S} \mathcal{K}(i - x + \delta, j - y + \delta, s, t) \mathcal{X}(i, j, s), \tag{2}$$

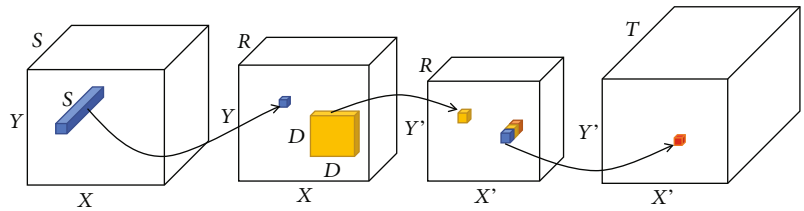


Figure 3: Decomposed convolution layers based on CP. Decomposed filters are of sizes of $1 \times 1 \times S \times R$, $D \times D \times R \times R$, and $1 \times 1 \times R \times T$. The series of three small filters approximate the original filter.

where $\mathcal{K}(i-x+\delta, j-y+\delta, s, t)$ is a $4D$ kernel tensor of size $D \times D \times S \times T$ with the first two dimensions corresponding to spatial dimensions, the third dimension corresponding to input channels, and the fourth dimension corresponding to output channels. The δ denotes half-width $(D-1)/2$. As shown in Figure 2, the convolution procedure consists of T convolutions of $D \times D \times S$.

In order to compress GAN, we use CP decomposition to reconstruct convolutional layers in a generator. Spatial dimension in the convolutional layer does not need decomposition as it is relatively small (e.g., 3×3 or 4×4).

$$\mathcal{K}(i-x+\delta, j-y+\delta, s, t) = \sum_{r=1}^{R} \mathcal{K}^{(1)}_{r,s} \mathcal{K}^{(2)}_{r,j,i} \mathcal{K}^{(3)}_{t,r}, \tag{3}$$

where $\mathcal{K}^{(1)}_{r,s}$, $\mathcal{K}^{(2)}_{r,j,i}$, and $\mathcal{K}^{(3)}_{t,r}$ are the three components of sizes $R \times S$, $R \times D \times D$, and $T \times R$, respectively.

Substituting Equation (3) into Equation (2) and performing simple manipulations give Equation (4). Equation (4) can approximate the convolution (Equation (2)) from the input tensor $\mathcal{X}$ into the output tensor $\mathcal{Y}$.

$$\mathcal{Y}(x, y, t) = \sum_{r=1}^{R} \mathcal{K}^{(3)}_{t,r} \left(\sum_{j=1}^{D} \sum_{i=1}^{D} \mathcal{K}^{(2)}_{r,j,i} \left(\sum_{r=1}^{R} \mathcal{K}^{(1)}_{r,s} \mathcal{X}(i, j, s) \right) \right) \tag{4}$$

Based on Equation (4), replacing the original convolution with a sequence of three convolutions can reduce convolutional layers' parameters. For the convenience of understanding, we call these three layers as first, second, and third:

$$\mathcal{U}^{(1)}(i, j, r) = \sum_{s=1}^{S} \mathcal{K}^{(1)}_{\mathrm{r,s}} \mathcal{X}(i, j, s), \tag{5}$$

$$\mathcal{U}^{(2)}(x, y, r) = \sum_{j=1}^{D} \sum_{i=1}^{D} \mathcal{K}^{(2)}_{r,j,i} \mathcal{U}^{(1)}(i, j, r) \tag{6}$$

$$\mathcal{Y}(x, y, t) = \sum_{r=1}^{R} \mathcal{K}^{(3)}_{t,r} \mathcal{U}^{(2)}(x, y, r), \tag{7}$$

where $\mathcal{U}^{(1)}(i, j, r)$ and $\mathcal{U}^{(2)}(x, y, r)$ are intermediate tensors of sizes $R \times X \times Y$ and $R \times (X - D + 1) \times (Y - D + 1)$, respectively. The target tensor is computed by three convolutions (see Figure 3).

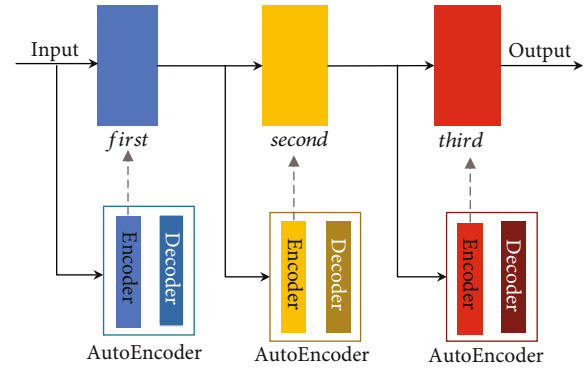


Figure 4: Layer-wise pretraining.

3.2. Layer-Wise Pretraining. In this paper, we design a new architecture based on canonical polyadic decomposition which decomposes a layer into three layers. Because the network is deeper than the original model and the training process of GAN is unstable, it is necessary to conduct layer by layer pretraining for the model. He et al. [36] proposed that the effect of random initialization was not worse than that of pretraining, but the convergence time was slower. We adopt the autoencoder to pretrain the model layer by layer.

An autoencoder consists of an encoder and decoder. The encoder is to turn input into a hidden spatial representation. It can be represented by a function $h = f(x)$. The decoder is aimed at reconstructing input from a representation of hidden space by function $x' = g(h)$. As a whole, autoencoder can be described by function $g(f(x)) = x'$, where x' is close to original input x. Autoencoder learns valuable information from original input by reconstruction.

The training process is that n autoencoders are trained in sequence. After the first autoencoder training, output of the first encoder is taken as the input of the second autoencoder. And third autoencoder takes the output of the second encoder as the input. The structure of the encoder is the same as that of decomposed layers. After training, the encoder replaces a decomposed layer. Taking the training of these three layers *first*, *second*, and *third* as an example in Figure 4, we train the *first* autoencoder taking the *first's* input and replace parameters of the *first* with those of the encoder in the *first* autoencoder. Then, after taking the output of the *first* as the *second* autoencoder's input to train the *second* autoencoder, we replace parameters of *second* with the encoder's in the *second* autoencoder. So is the *third's* training.

The training algorithm is shown in Algorithm 1.

Overall scheme of compact architecture training algorithm.
Input: mini-batch images x, matching text t, mismatching text $\hat{t}$, number of training batch steps S
Output: a compact architecture for text-to-image GAN
1: Obtain three small layers as *first*, *second*, *third* using Equation (5),(6),(7) to decompose original convolutional layer;
2: Adopt autoencoder to pre-train model layer by layer;
3: Select an appropriate learning rate for the decomposed model;
4: **for** $N = 1$ **to** S **do**
5: Encode matching text description t and mismatching text description $\hat{t}$ to description embedding h and $\hat{h}$;
6: Draw sample of random noise z;
7: Concatenate z to description embedding h and $\hat{h}$;
8: Feed forward z through generator and generate samples of {real image, right text}, {real image, wrong text} and {fake image, right text};
9: Update discriminator D using Adam;
10: Update generator G using Adam;
11: **end for**

Algorithm 1

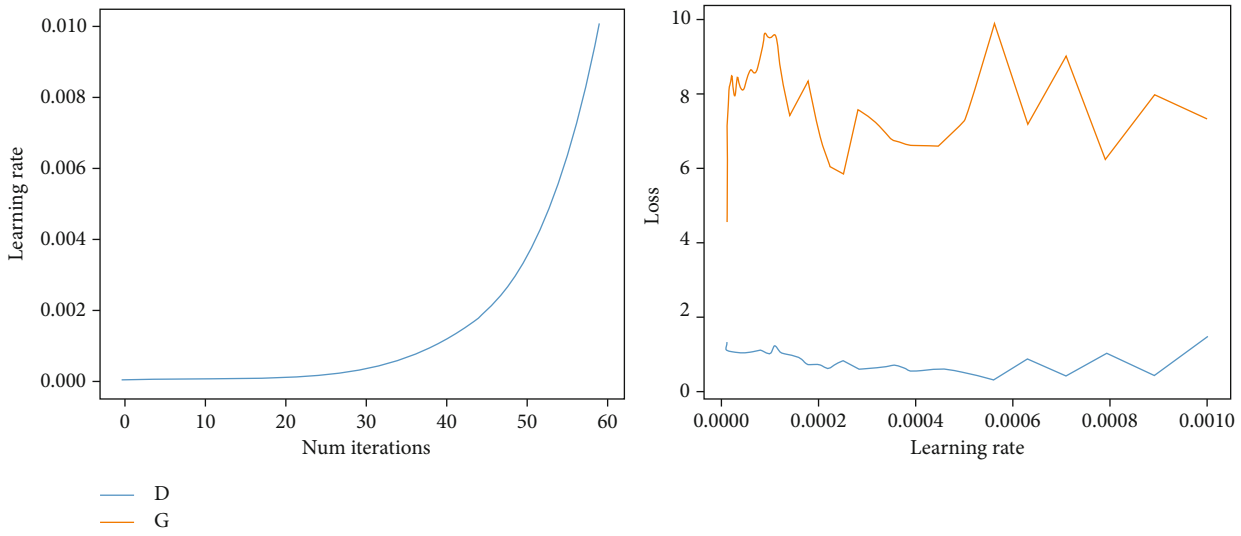


Figure 5: Selection of appropriate learning rate for the decomposition model.

3.3. Selection of the Learning Rate. The learning rate determines whether objective function converges to the local minimum and when it converges to the minimum. A suitable learning rate can make objective function converge to the local minimum in a suitable time. Due to the decomposed model getting deeper, the learning rate should be adjusted appropriately. The learning rate determines the step size of weight iteration, so it is a very sensitive parameter. Its influence on the model performance is reflected in two aspects: the first is the initial learning rate and the second is the transformation scheme of the learning rate.

Smith [37] put forward an excellent way to find the initial learning rate which was called the LR range test. The method is very simple and useful to set the learning rate. We use this method to choose the appropriate range of the learning rate. The accuracy or loss curve is obtained by using different learning rates. And we can set two inflection points of increasing and decreasing precision as the upper bound and the lower bound.

Figure 5 is the curve of the increasing learning rate and the curve of loss corresponding to the increase in the number of iterations within CUB200-2011 for the reconstructed architecture. The method LR range test has three hyperparameters: iteration, max learning rate, and min learning rate. In this paper, we set three hyperparameters as 40, 0.001, and 0, respectively. We change the learning rate once every 40 iterations. The learning rate changes according to the following formula:

$$lr = lr \times 10^{1/20}. \tag{8}$$

Figure 5 shows that loss has a minimum value when the learning rate is around 0.0002 and loss decreases sharply when learning rate is around 0.00017 and 0.00014.

Table 1: Comparison between the reconstructed model and the original model.

Model	LR	FID	IS	FLOPs	#Parameters
Original	0.0002	66.92	2.56 ± 0.03	3.0×10^{10}	5.8×10^{6}
Reconstructed	0.00017	65.05	2.38 ± 0.01	2.3×10^{10}	4.7×10^{6}

4. Experiment

4.1. Setups

4.1.1. Model. We conduct experiments on a classic and basic model in text-to-image GAN to demonstrate the generality and effectiveness of our method. Reed et al. [4] was the first to successfully apply generative adversarial networks to cross modal image generation which converted a descriptive text into images directly. The colour information obtained by GAN and GAN-cls is correct, but images look unreal. Images generated by GAN-int-cls are more reasonable, so we choose GAN-int-cls.

4.1.2. Dataset. We evaluate our decomposed architecture on the following dataset:

(i) Caltech-UCSD Birds-200-2011. There are 11788 bird images in the data set, including 200 bird subclasses, 5994 images in the training dataset, and 5794 images in the test set. Each image provides image class information, bird bounding box, key part information of the bird, and attribute information of the bird

4.1.3. Implementation Details. For the reconstructed model, the initial learning rate is 0.00017 for both the generator and discriminator during training. MultistepLR is a learning rate attenuation method in Pytorch. And we adjust learning rate by attenuation coefficient 0.85 in MultistepLR. The batch size on Caltech-UCSD Birds-200-2011 is 64 followed by the setting in the original paper [4] and trained for 1000 epochs. ADAM [38] solver with beta1 0.5 is used for all models. For the sake of comparison, we handle the dataset the same as StackGAN++ [5]. We split CUB into class-disjoint training and test sets and use char-CNN-RNN [19] to obtain text embedding of given description according to images.

4.1.4. Evaluation Metrics. We use an inception score (IS) and Fréchet inception distance (FID) to evaluate generated images quantitatively. IS is commonly used as an evaluation index of GAN. It evaluates the performance of generative models to use entropy and KL divergence by feeding a large number of generated pictures to Inception V3. The large IS score means high quality of the generated images. FID represents the distance between the feature vector of generated images and that of real images. Small FID score means small distance of images distribution, which means that generated images have high definition and rich diversity. We compute IS and FID on 30k samples randomly generated for the test set the same as StackGAN++ [5].

4.2. Results.

In the dataset Caltech-UCSD Birds-200-2011, our model is slightly better than the original structure in the Caltech-UCSD Birds-200-2011 in FID and is similar to the original model in IS. The FID and IS of the original model are 66.92 and 2.56 ± 0.03, while those of our model were 65.05 and 2.38 ± 0.01. There are many redundant parameters in the model. Our model got 19% and 23% reduction in parameters and FLOPs, respectively. Our model reduces 6.8×10^{9} FLOPs and 1.1×10^{6} parameters compared to the original model (see Table 1).

Table 2: Different choices of the rank ratio for the reconstructed model.

Ratio	FID	IS	FLOPs	#Parameters
0.1	228.37	2.36 ± 0.02	1.20×10^{10}	2.46×10^{6}
0.2	194.18	2.25 ± 0.02	1.23×10^{10}	2.56×10^{6}
0.3	130.46	2.31 ± 0.02	1.30×10^{10}	2.69×10^{6}
0.4	136.44	2.33 ± 0.02	1.39×10^{10}	2.86×10^{6}
0.5	123.24	2.32 ± 0.02	1.49×10^{10}	3.07×10^{6}
0.6	126.94	2.37 ± 0.02	1.61×10^{10}	3.31×10^{6}
0.7	102.00	2.26 ± 0.02	1.76×10^{10}	3.59×10^{6}
0.8	115.88	2.28 ± 0.02	1.93×10^{10}	3.92×10^{6}
0.9	91.54	2.32 ± 0.01	2.12×10^{10}	4.28×10^{6}
1.0	65.05	2.38 ± 0.01	2.33×10^{10}	4.68×10^{6}

It is a classic topic to balance the performance and the compression ratio. Trade-off in rank decomposed GAN is more difficult to achieve because GAN is unstable and rank selection is NP-hard in rank decomposition. In rank decomposition, rank represents the compression ratio. As shown in Table 2, we do a large number of experiments to find the balance. We explored different ranks, which are the ratios in Table 2. The ratio is the rank ratio, where 1.0 is the full-rank decomposition and 0.9 means about 0.9 times of original model's rank. Table 2 shows that with increasing rank, FLOPs and parameters grow. FID is getting smaller and smaller, while IS only has a little change. Maybe it is because FID is more sensitive to model collapse and IS is little unstable. Compared with IS, FID has better robustness. When the rank ratio is 1.0, FID and IS get the best value which is similar to the original model. Although rank is 1.0, the model is compressed by about 20%. It is effective to use CP decomposition to design a compact GAN network. Results on the Caltech-UCSD Birds-200-2011 dataset can be seen in Figure 6. Rank decomposition can reconstruct a model with less parameters from scratch without loss of generating quality.

The reconstructed model proves that there are redundant parameters in the original model at the current optimal effect

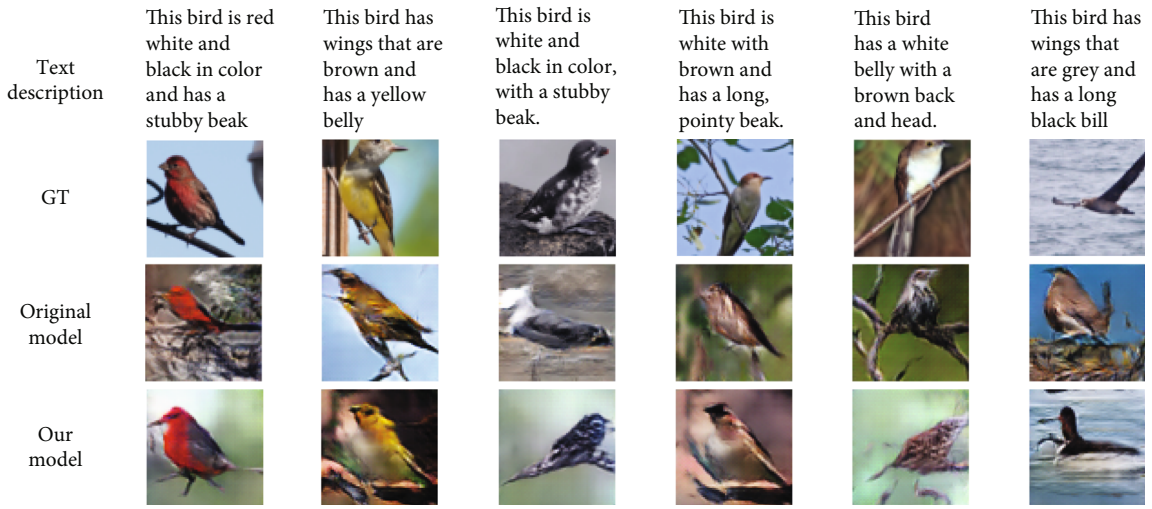


Figure 6: Example results generated by our proposed model and original model conditioned on text descriptions from the CUB test set.

Table 3: Comparison for different learning schemes of the reconstructed model.

Learning scheme	LR	FID	IS
Fixed learning rate	0.0002	99.54	2.37 ± 0.02
Fixed learning rate	0.00017	120.86	2.18 ± 0.02
Fixed learning rate	0.00014	106.45	2.29 ± 0.02
CosineAnnealingWarmRestarts	0.00025~0.00012	96.95	2.29 ± 0.02
CosineAnnealingWarmRestarts	0.0002~0.0001	119.12	2.05 ± 0.02
CosineAnnealingWarmRestarts	0.0002~0.00014	91.08	2.29 ± 0.02
CosineAnnealingWarmRestarts	0.0002~0.00017	92.88	2.30 ± 0.02
CosineAnnealingWarmRestarts	0.0017~0.00014	98.09	2.36 ± 0.02
MultistepLR	0.0002	131.57	1.96 ± 0.03
MultistepLR	0.00017	65.05	2.38 ± 0.01
MultistepLR	0.00014	84.27	2.36 ± 0.03

point. The full-rank decomposition result which is little better than the original model in FID may be because the model is small so that it is easier to find the area where the global optimization point is located. In this paper, we also do quantities of comparison experiments to find the global optimization. As shown in Table 3, we adopt three schemes to explore the optimization point. The LR range test proves that 0.00017 and 0.00014 maybe is the better learning rates. We used three transformation schemes of learning rate which is fixed learning rate, CosineAnnealing with WarmRestarts [39], and MultistepLR. The initial learning rates are around 0.00017 and 0.00014. The result proves that the MultistepLR transformation scheme helps to find the global optimization.

5. Conclusion

Cross modal GAN have a wide range of applications in computer vision. However, these models have too high computation and many parameters to be deployed on the mobile end. In this paper, we developed a compact model for text-to-image GAN based on CP decomposition. We replace a complex convolution layer with three small convolutions. Due to unstable training of GAN and uncontrollable generating, we pretrained the decomposed network layer by layer and explored a considerable amount of experiments to select an appropriate learning rate. We demonstrated that cross modal GAN can be reconstructed with less parameters without quality falling. GAN-int-cls is the most classic and basic model of cross modal GAN. CP decomposition is a standard and efficient tensor decomposition method. Our method has proven that CP decomposition is an efficient decomposition method for GAN in the general evaluation index FID and IS. It is applicable for other cross modal GAN to use CP decomposition. In future work, we aim to further study a more compact and stable network architecture of cross modal GAN.

Data Availability

The datasets used in this paper are public datasets which can be accessed through the following website: http://www.vision.caltech.edu/visipedia/CUB-200-2011.html.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (No. DUT20LAB136).

References

[1] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Generating images from captions with attention," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, 2016.

[2] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, vol. 48, pp. 1747–1756, NY, USA, 2016.

[3] A. van den Oord, N. Kalchbrenner, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," *Advances in Neural Information Processing Systems*, vol. 29, pp. 4790–4798, 2016.

[4] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee,, "Generative adversarial text to image synthesis," *Proceedings of Machine Learning Research*, vol. 48, pp. 1060–1069, 2016.

[5] H. Zhang, T. Xu, H. Li et al., "Stackgan++: realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1947–1962, 2019.

[6] T. Xu, P. Zhang, Q. Huang et al., "Attngan: fine-grained text to image generation with attentional generative adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1316–1324, Salt Lake City, UT, USA, June 2018.

[7] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6199–6208, Salt Lake City, UT, USA, June 2018.

[8] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An incremental deep convolutional computation model for feature learning on industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1341–1349, 2019.

[9] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.

[10] J. Gao, P. Li, and Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in Internet of things," *Future Generation Computer Systems*, vol. 99, pp. 508–516, 2019.

[11] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.

[12] H. Huang, M. Lin, L. T. Yang, and Q. Zhang, "Autonomous power management with double-Q reinforcement learning method," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1938–1946, 2020.

[13] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1269–1277, 2014.

[14] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Proceedings of the British Machine Vision Conference 2014*, Nottingham, UK, 2014.

[15] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned CP-decomposition," 2014, https://arxiv.org/abs/1412.6553.

[16] S. Lin, R. Ji, X. Guo, and X. Li, "Towards convolutional neural networks compression via global error reconstruction," in *International Joint Conference on Artificial Intelligence*, pp. 1753–1759, NY, USA, 2016.

[17] F. Chollet, "Xception: deep learning with depthwise separable convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800–1807, 2017.

[18] A. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[19] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," *Advances in Neural Information Processing Systems*, vol. 29, pp. 217–225, 2016.

[20] H. Zhang, T. Xu, H. Li et al., "StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915, Venice, Italy, 2017.

[21] A. Dash, J. C. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "TAC-GAN - text conditioned auxiliary classifier generative adversarial network," 2017, https://arxiv.org/abs/1703.06412.

[22] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1219–1228, 2018.

[23] W. Li, P. Zhang, L. Zhang et al., "Object-driven text-to-image synthesis via adversarial training," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12174–12182, 2019.

[24] Y. Li, Z. Gan, Y. Shen et al., "StoryGAN: a sequential conditional GAN for story visualization," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6329–6338, Long Beach, CA, USA, June 2019.

[25] R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua, "Learning separable filters," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2754–2761, 2013.

[26] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, "Predicting parameters in deep learning," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2148–2156, 2013.

[27] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Penksy, "Sparse convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 806–814, Boston, MA, USA, June 2015.

[28] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep

neural network training with high-dimensional output targets," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6655–6659, Vancouver, Canada, May 2013.

[29] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," *Interspeech*, pp. 2365–2369, 2013.

[30] T. Garipov, D. Podoprikhin, A. Novikov, and D. Vetrov, "Ultimate tensorization: compressing convolutional and FC layers alike," 2016, https://arxiv.org/abs/1611.03214.

[31] A. Novikov, D. Podoprikhin, A. Osokin, and D. Vetrov, "Tensorizing neural networks," *Neural Information Processing Systems*, vol. 28, pp. 442–450, 2015.

[32] Y. Ioannou, D. Robertson, J. Shotton, and R. Cipolla, "Training CNNs with low-rank filters for efficient image classification," *Journal of Asian Studies*, vol. 62, no. 3, pp. 952-953, 2015.

[33] M. Li, J. Lin, Y. Ding, Z. Liu, J.-Y. Zhu, and S. Han, "GAN compression: efficient architectures for interactive conditional GANs," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5283–5293, Seattle, WA, USA, June 2020.

[34] H. Shu, Y. Wang, X. Jia et al., "Co-evolutionary compression for unpaired image translation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3234–3243, Seoul, Korea (South), October 2019.

[35] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[36] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pretraining," in *Proceedings of the IEEE international conference on computer vision*, pp. 4918–4927, Seoul, Korea, 2019.

[37] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proceedings of the 2017 IEEE Winter Conference On Applications Of Computer Vision (WACV)*, pp. 464–472, Santa Rosa, CA, USA, 2017.

[38] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference On Learning Representations*, San Diego, CA, USA, 2015.

[39] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6612346, 8 pages
https://doi.org/10.1155/2021/6612346

Research Article

Noise Attenuation of Seismic Data via Deep Multiscale Fusion Network

Yu Sang,[1] **Jinguang Sun**,[1] **Dacheng Gao**,[2] **and Hao Wu**[2]

[1]*School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China*
[2]*Exploration and Development Research Institute, Liaohe Oilfield of CNPC, Panjin 124010, China*

Correspondence should be addressed to Yu Sang; sangyu2008bj@sina.com

Received 10 December 2020; Revised 2 February 2021; Accepted 5 March 2021; Published 25 March 2021

Academic Editor: Xiaojie Wang

Convolutional neural network- (CNN-) based deep learning (DL) architectures have achieved great success in many fields such as remote sensing, medical image processing, and computer vision. Recently, CNN-based models have also been attempted to solve geophysical problems. This paper presents a noise attenuation method of seismic data via a novel deep learning (DL) architecture, namely, deep multiscale fusion network (MSFN). Firstly, we integrate multiscale fusion (MSF) block to adaptively exploit local signal features at different scales from seismic data. And then, a series of stacked MSF blocks are formed into MSFN, which can restore the noisy seismic data effectively and preserve more useful signal information. Furthermore, a comparative study of our method and other leading edge ones is conducted by using synthetic seismic records and the SEG/EAGE salt and overthrust models. The results qualitatively and quantitatively show the capability of our method of achieving higher peak signal-to-noise ratios (PSNRs) while preserving much more useful information, comparing with other methods. Finally, our method is utilized in the real seismic data processing, obtaining satisfactory results.

1. Introduction

It is crucial to depict the underlying geological structures using the information contained in the seismic data acquired through the use of various sensing equipment and networks [1–7]. However, the reliability of seismic analysis is degenerated due to the random noise in seismic data. Hence, noise attenuation plays a critical role in improving signal-to-noise ratio (SNR) for geological interpretation based on seismic data.

In recent years, with the gradual extension of the field of seismic exploration, the deepening of exploration depth, and the increasingly complex exploration environment, the noise also increases significantly and can be more complex. This will hinder the realization of high-precision seismic exploration. So, remarkably improving the SNR becomes the most important and basic task. However, conventional seismic data denoising methods are difficult to satisfy the demands of high-precision seismic exploration. Therefore, it is urgent to develop a more effective new technique.

Up to now, many noise attenuation methods have been developed. The popular sparse representation of seismic data exploits the domain transform technology to denoise [8–15]. Learning-based methods [16, 17] are another type of effective methods, wherein a set of examples are used to generate an overcomplete dictionary, generally an explicit matrix. Recently, deep learning (DL) has become a research focus due to its advantages compared with traditional learning. DL-based convolutional neutral networks (CNNs) [18–29] are intensively adapted to process tremendous multimedia problems with impressive results.

In the present work, a noise attenuation method of seismic data via a novel deep learning architecture is proposed. Our contributions in this paper are threefold:

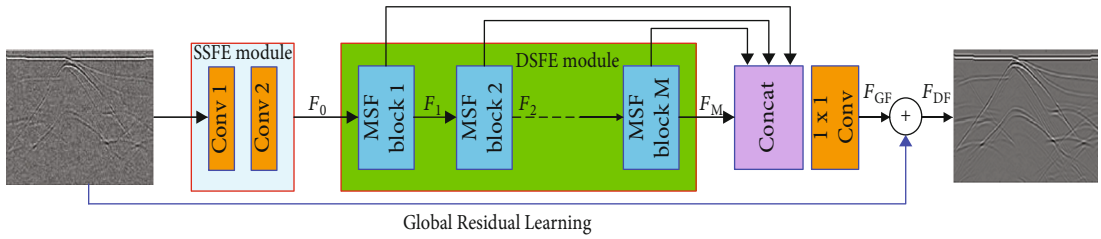


Figure 1: The architecture of our proposed MSFN.

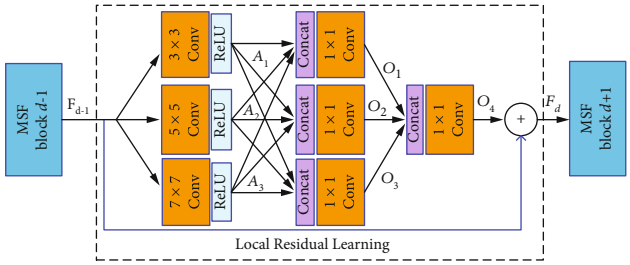


Figure 2: The architecture of our proposed MSF block.

(i) We propose MSF block to adaptively exploit local signal features at different scales from seismic data

(ii) A series of stacked MSF blocks are formed into MSFN, which can restore the noisy seismic data effectively and preserve more useful signal information

(iii) The superior of our method over other leading-edge methods is demonstrated with the synthetic seismic records, SEG/EAGE salt and overthrust models, and real seismic data

The remainder of the paper is structured as below. Section 2 reviews related work. Section 4 presents a detailed description of the suggested scheme, and Section 5 validates the proposed method. Finally, the conclusions of this paper are summarized in Section 5.

2. Related Work

At present, numerous seismic denoising approaches [8–17, 30–33] including some new methods [10, 13, 29] have been suggested. Actually, seismic random noise, which penetrates the whole time domain, is the most common in all types of noise for seismic data. And it can seriously interfere with effective seismic signals, thus resulting signal perturbation. Various effective random noise attenuation approaches, e.g., the empirical mode decomposition- (EMD-) based methods and the sparse transform-based approaches have been proposed on the basis of the initial denoising method developed by Canales [26]. Chen and Ma [32] proposed to use f-x EMD predictive filtering to remove the random noise. Liu et al. [33] presented a random noise attenuation method based on variational mode decomposition to perform seismic time-frequency analysis. Chen and Fomel [12] suggested a novel random noise attenuation method based on an EMD-seislet transform. Neelamani et al. [9] presented a coherent and random noise attenuation method based on the curvelet transform. Zhang and Lu [8] proposed a wavelet transform-based denoising approach and achieved improved resolution of seismic data. Subsequently, some improved and/or combined transform domain based methods were proposed [8–15], which achieves good results.

Compared with conventional superresolution (SR) methods, the CNN-based schemes from the first SRCNN [18] to the latest feedback network [29] can remarkably improve the SR quality. The shallow structure of SRCNN limits its performance. To overcome this drawback, deepening structures were adopted in networks. For example, a deeper structure was used in the VDSR model proposed by Kim et al. [20]. Several new very deep models, e.g., RCAN [21], achieved outstanding SR performance. Besides, dense connections integrated SR models, e.g., SRDenseNet [23] and MemNet [25], displayed a better resolution. Moreover, by connecting all the same signal feature extraction (SFE) modules in the entire network, the efficiency of the constructed SR methods based on CNNs, e.g., RDN [26], IDN [27], MSRN [28], and SRFBN [29], could be increased, indicating each block was crucial.

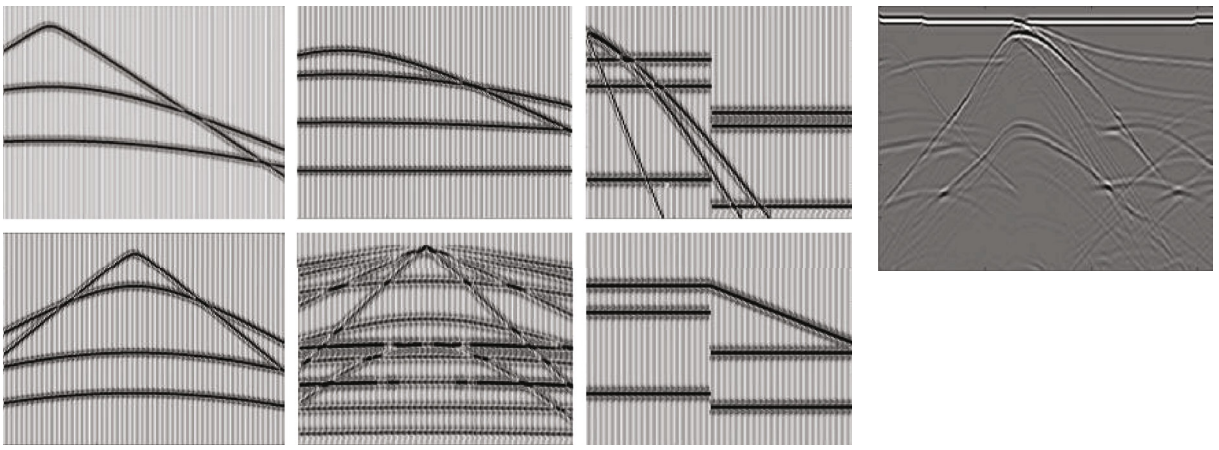
3. Proposed Method

This section presents the network architecture of a novel seismic data denoising method (MSFN). The structure has two parts, namely, a shallow signal feature extraction (SSFE) and a deep signal feature extraction (DSFE) module, as shown in Figure 1. Let us denote the clean data and the noised data by I^H and I^L, respectively, by solving the problem:

$$\widehat{\theta} = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L\left(I_i^L, I_i^H\right), \tag{1}$$

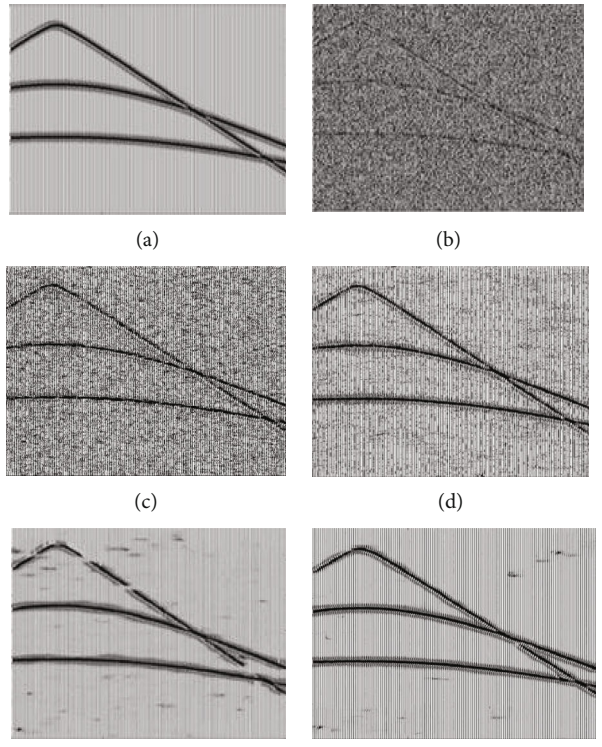
where L denotes the loss function which can minimize the discrepancy between the clean data I^H and the noised data I^L, N denotes the number of training samples, and $\theta = \{W^1, W^2, \cdots, W^p, b^1, b^2, \cdots, b^p\}$ is a set of weights and biases of the pth convolutional layer.

The mean square error (MSE) function [26] and L2 function are the two most popular objective optimization



(a) (b)

Figure 3: Seismic data. (a) Partial synthetic seismic records. (b) Stacked profile acquired by SEG/EAGE salt and overthrust model.



(a) (b) (c) (d) (e) (f)

Figure 4: Synthetic seismic data denoising. (a) Clear seismic data. (b) Seismic data with added strong random noise (PSNR: 68.1429 dB). (c, d) Denoised seismic data by curvelet-based threshold denoising (PSNR: 83.1584 dB) and shearlet-based threshold denoising (PSNR: 85.6618 dB). (e, f) Denoised seismic data by IDN (PSNR: 89.8992 dB) and our method (PSNR: 91.6515 dB).

functions in image SR. Due to the excessively smooth textures in the solutions of the MSE and L2 optimization problems, we found marginal performance improvement could be obtained, except for their high PSNR/SSIM. Besides, training with MSE loss was not a good option according to the experiment by Lim et al. [23]. To reduce computations and avoid introducing unnecessary training tricks, as a better alternative, a mean absolute error (MAE) function L_1 are used and given by

$$L_1 = \frac{1}{N}\sum_{i=1}^{N} \left\| I_i^L - I_i^H \right\|_1, \tag{2}$$

where $\|\cdot\|$ denotes L1 norm. So, the shallow feature F_0 can be

Table 1: Comparison of PSNR on synthetic seismic records for different methods.

Noisy data (dB)	Traditional methods			Deep learning-based methods	
	Wavelet threshold	Curvelet threshold	Shearlet threshold	IDN	MSFN (ours)
78.1486	87.2569	90.3568	92.1864	95.0346	96.8461
73.6958	81.9565	86.6764	88.5431	92.6243	94.5628
68.1429	76.3467	83.1584	85.6681	89.8992	91.6515
64.6281	71.4122	74.5964	76.1624	82.8568	84.1436
Average	79.2431	83.6970	85.6400	90.1037	91.8010

Table 2: Comparison of PSNR on synthetic seismic records for 1-3 scale fusion networks.

Noisy data (dB)	1 scale (baseline)	2 scale (ours)	3 scale (ours)
78.1486	94.8629	95.9651	96.8461
73.6958	91.5431	93.4957	94.5628
68.1429	88.7264	90.7647	91.6515
64.6281	81.6358	83.3652	84.1436
Average	89.1921	90.8977	91.8010

Table 3: Comparison of PSNR on SEG/EAGE salt and overthrust model for different methods.

Noise level	Traditional methods			Deep learning-based methods	
	Wavelet threshold	Curvelet threshold	Shearlet threshold	IDN	MSFN (ours)
0.05	91.25	92.35	93.18	95.03	95.84
0.10	83.95	85.66	86.54	87.62	89.25
0.20	77.34	78.25	79.96	82.35	84.45
0.30	72.41	72.59	73.94	75.85	78.36
Average	81.24	82.21	83.41	85.21	83.41

extracted by two convolution layers as follows:

$$F_0 = H_{SSFE1}\left(H_{SSFE2}\left(I^L\right)\right), \tag{3}$$

where and are the SSFE convolution operations of two layers. After that, F_0 is employed in DSFE module, containing a cascaded MSF block set. Then, the output information is adaptively controlled by using a 1×1 convolutional layer as follows (named as feature fusion):

$$F_{GF} = H_{GFF}([F_1, F_2, \cdots, F_M]), \tag{4}$$

where denotes the composite function of a 1×1 convolutional layer and $[F_1, F_2, \cdots, F_M]$ is the feature map set produced by all MSF blocks. By using the global residual learning, we get the feature maps F_{DF} by

$$F_{DF} = F_{GF} + F_0. \tag{5}$$

Table 4: Comparison of PSNR on SEG/EAGE salt and overthrust model for 1-3 scale fusion networks.

Noise level	1 scale (baseline)	2 scale (ours)	3 scale (ours)
0.05	94.71	95.42	95.84
0.10	87.33	88.23	89.25
0.20	81.95	83.04	84.45
0.30	75.24	76.95	78.36
Average	84.81	85.91	83.41

Figure 2 shows the proposed MSF block. A three-bypass network with various convolutional kernels for each pass is constructed in each MSF block. So that, the signal features at different scales can be detected because the information can be shared between these bypasses. According to [18], we define the operation as follows:

$$\begin{aligned} A_1 &= \sigma\left(W^1_{3\times3} \cdot F_{d\text{-}1} + b^1\right), \\ A_2 &= \sigma\left(W^1_{5\times5} \cdot F_{d\text{-}1} + b^2\right), \\ A_3 &= \sigma\left(W^1_{7\times7} \cdot F_{d\text{-}1} + b^3\right), \end{aligned} \tag{6}$$

$$\begin{aligned} O_1 &= \sigma\left(W^2_{1\times1} \cdot [A_1, A_2, A_3] + b^4\right), \\ O_2 &= \sigma\left(W^2_{1\times1} \cdot [A_1, A_2, A_3] + b^5\right), \\ O_3 &= \sigma\left(W^2_{1\times1} \cdot [A_1, A_2, A_3] + b^6\right), \end{aligned} \tag{7}$$

$$O_4 = W^3_{1\times1} \cdot [O_1, O_2, O_3] + b^7, \tag{8}$$

$$F_d = F_{d-1} + O_4, \tag{9}$$

where $W^1_{3\times3}$, $W^1_{5\times5}$, and $W^1_{7\times7}$ refer to the weights of 3×3, 5×5, and 7×7 convolutional layers in Figure 2, respectively; $W^2_{1\times1}$ and $W^3_{1\times1}$ refer to the 1×1 convolutional weights of the second and third layers, respectively; b denotes the bias and $[A_1, A_2, A_3]$ denote the feature map set produced by A_1, A_2, and A_3. F_{d-1} and F_d are the input and output of the dth MSF block, respectively. $\sigma(\cdot)$ denotes the ReLU function [34].

4. Experimental Results

The qualitative and quantitative experiments are conducted to evaluate the performance of our method. In this work, three traditional seismic denoising methods (wavelet-based threshold denoising (WTD), curvelet-based threshold denoising (CTD), and shearlet-based threshold denoising (STD)) and one deep learning-based method (information distillation network (IDN) [27]) are selected for the comparative study.

The basic data can be synthesized with 24 seismic records including linear, curvilinear, fault, and various dip angle events. The trace number is 150 and the sampling frequency is 1000 Hz. The selected seismic wavelet is Ricker wavelet,

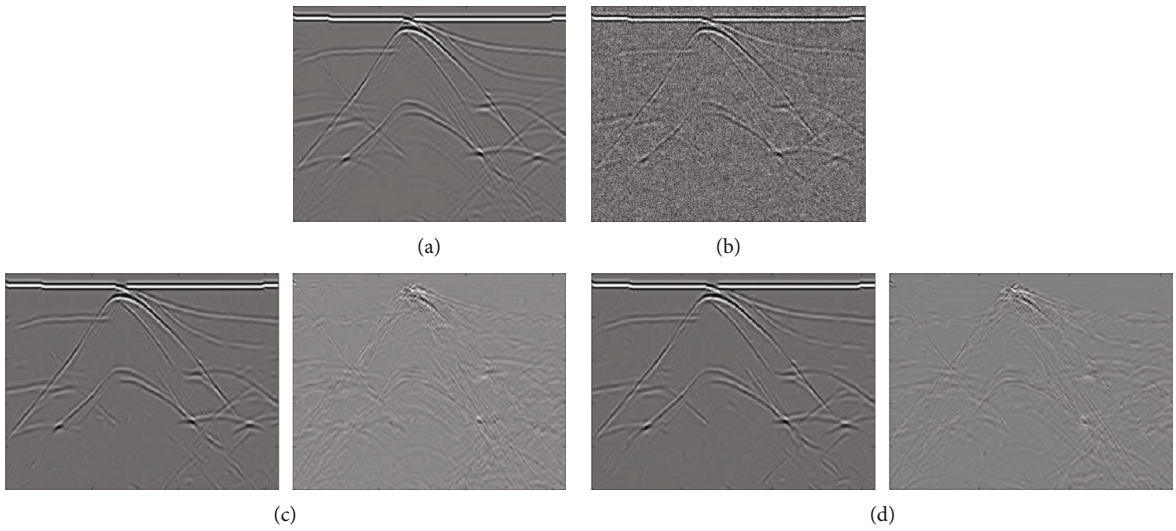


FIGURE 5: Visual results of seismic data denoising. (a) Clear data. (b) Noisy data. (c) Left panel: denoised data by IDN; right panel: difference between the clean data and the denoised data on the left. (d) Left panel: denoised data by our method; right panel: difference between the clean data and the denoised data on the left.

which can be expressed as

$$x(t) = \left(1 - 2\pi^2 f^2 t^2\right) \cdot e^{-\pi^2 f^2 t^2}, \tag{10}$$

where f denotes the sampling frequency and t denotes time. Figure 3(a) presents partial synthetic seismic records. The SEG/EAGE salt and overthrust models [35] are used to obtain the immigrated stack profile (Figure 3(b)). At the same time, these two types of seismic data are rotated by 45°, 90°, 135°, 180°, 270°, and 360°, respectively, following [25]. To obtain additional expanded versions, random noises of various levels are added to the original and rotated data, and training sets are the 80% versions, with the rest as test sets.

Our MSFN contains 12 MSF blocks and all convolutional layers have 32 filters. There are 24 pixels overlapping for training in the cropped training seismic data of 48×48 patches. The batch size is set as 64. The leaning rate reduces by half for every 50 epochs and the initial value of 10^{-4} for all layers. Our model is trained with Tesla k80 GPUs, and time is about 14 hours.

The denoising performance of our method is evaluated as below. All models are trained with the same training set for fair comparison. And the codes of contrastive methods are publicly released. The reconstruction results are justified by a quantitative evaluation metric of the PSNR [36], which can be calculated as follows:

$$\mathrm{PSNR}\left(X', X\right) = 10 \log_{10} \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathrm{MAX}_I{}^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} \left(X'(i,j) - X(i,j)\right)^2}, \tag{11}$$

where X denotes the clear data of size $M \times N$, X' denotes the denoised seismic data, $X(i,j)$ and $X'(i,j)$ are the values of element (i,j) of X and X', respectively, and MAX_I denotes the maximal signal intensity can be possibly achieved.

Firstly, synthetic seismic records are used in the comparing study of our method and the traditional WTD, CTD, and STD methods and deep learning IDN model, and the results are presented in Figure 4. A better result was achieved by our method with higher PSNR value, compared with other methods. In addition, the performance of our method is also quantitatively evaluated on synthetic seismic records. Table 1 shows the PSNRs (dB) with bolded optimal values. The comparison indicates much higher PSNRs of our method than that of others. Table 2 presents the PSNRs (dB) on synthetic seismic records 1-3 scale fusion networks. It can be seen that 3 scale fusion network achieve the best results.

Secondly, the SEG/EAGE salt and overthrust models are used for evaluating our method. Table 3 shows the PSNRs (dB) with bolded optimal values. The significantly higher PSNR values of our method are obtained again, comparing with other methods. Particularly, our method shows a more considerable performance when the level of noise in the seismic data increases. In Table 3, the higher the noise level is, the lower the SNR is. Table 4 presents the PSNRs (dB) on SEG/EAGE salt and overthrust model for 1-3 scale fusion networks. It can be seen that 3 scale fusion network achieve the best results. Besides, a qualitative comparison between our model and a deep learning-based one and the results are presented in Figure 5. We have Figure 5(b) by adding random noise to the clean seismic data (Figure 5(a)). Figures 5(c) and 5(d) are the obtained denoised results. Obviously, our method is an ideal denoising method for removing random noises while keeping coherent details.

Furthermore, we select the field data examples (noisy seismic data of Liaohe depression, China) in the same data acquisition work area with the same way of excitation and reception to validate the processing result of the proposed method. We utilize traditional random noise reduction

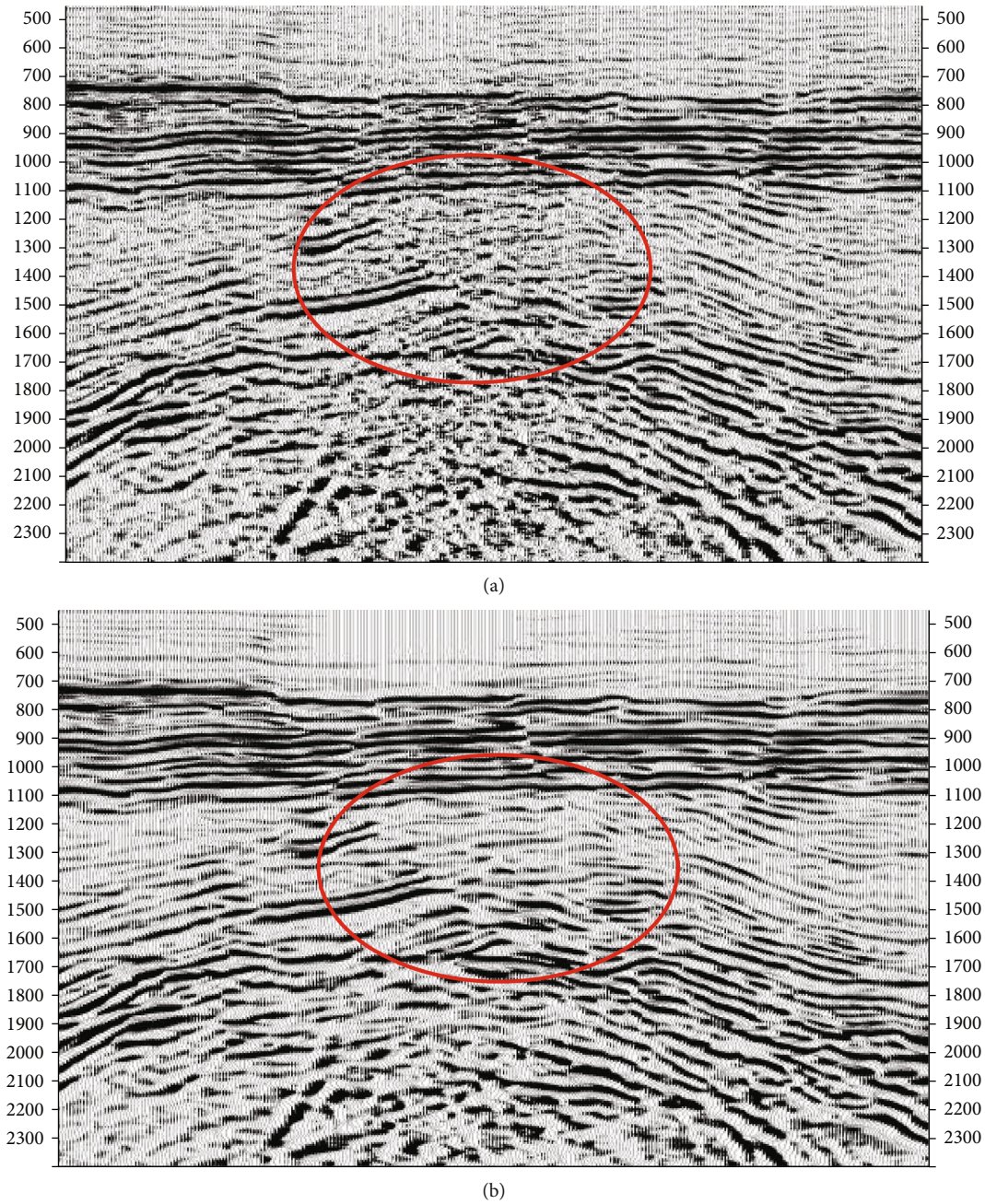


Figure 6: Migration profiles. (a) Original noisy section. (b) Denoised section by our method.

modular of large processing system to roughly denoise these data, guaranteeing no loss of valid information. The denoised data are view as targeted clear data. Due to the generalization ability of deep learning, we add random noise of various levels to the targeted data with the aim to learn and recognize noise and effective signals. Similarly, to obtain additional expanded versions, we rotate these real seismic data by 45°, 90°, 135°, 180°, 270°, and 360°, respectively. 80% versions are selected as training sets; the rest is as test sets. Figures 6(a) and 6(b) present the original noisy data and the denoised result by our method, respectively. Some effective signals highlight, especially the region in the red ellipse; the interlayer structure is clearer; and the continuity of the events is also enhanced, as shown in Figure 6.

5. Conclusions

We propose a novel network MSFN based on CNNs to denoise seismic data, wherein a cascaded MSF block set and seismic data features are exploited to perform noise attenuation. The results qualitatively and quantitatively demonstrate our scheme is much superior to other leading edge ones especially in promoting the seismic data restoration ability.

Data Availability

The data used to support the study can be available in the link: https://s3.amazonaws.com/open.source.geoscience/open_data/seg_eage_models_cd/salt_and_overthrust_models.tar.gz

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 of China (NSFC) under Grant No. 61602226, in part by the PhD Startup Foundation of Liaoning Technical University of China under Grant No. 18-1021, and in part by the Project supported by Discipline Innovation Team of Liaoning Technical University of China under Grant No. LNTU20TD-22.

References

[1] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1028–1042, 2019.

[2] N. Chen, T. Qiu, X. Zhou, K. Li, and M. Atiquzzaman, "An intelligent robust networking mechanism for the internet of things," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 91–95, 2019.

[3] C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, "Delay-aware grid-based geographic routing in urban VANETs: a backbone approach," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2324–2337, 2019.

[4] Z. L. Ning, P. R. Dong, X. J. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2021.

[5] Z. L. Ning, P. R. Dong, X. J. Wang et al., "Partial computation offloading and adaptive task scheduling for 5g-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 19, p. 1, 2020.

[6] Z. L. Ning, P. R. Dong, X. J. Wang et al., "Distributed and dynamic service placement in pervasive edge computing networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, 2021.

[7] X. J. Wang, Z. L. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, p. 1, 2020.

[8] J. H. Zhang and J. M. Lu, "Application of wavelet transform in removing noise and improving resolution of seismic data," *Journal of University of Petroleum: Edition of Natural Science*, vol. 31, no. 12, pp. 1975–1981, 1997.

[9] R. Neelamani, A. I. Baumstein, D. G. Gillard, M. T. Hadidi, and W. L. Soroka, "Coherent and random noise attenuation using the curvelet transform," *The Leading Edge*, vol. 27, no. 2, pp. 240–248, 2008.

[10] J. Xu, W. Wang, J. Gao, and W. Chen, "Monochromatic noise removal via sparsity-enabled signal decomposition method," *IEEE Geoscience Remote Sensing Letter*, vol. 10, no. 3, pp. 533–537, 2013.

[11] L. Chengming, W. Deli, W. Tong, F. Fei, C. Hao, and M. Gege, "Random seismic noise attenuation based on the shearlet transform," *Acta Petrolei Sinica*, vol. 35, no. 4, pp. 692–699, 2014.

[12] Y. Chen and S. Fomel, "EMD-seislet transform," *85th SEG Annual International Meeting, Expanded Abstracts*, pp. 4775–4778, 2015.

[13] B. Wang, R. S. Wu, X. Chen, and J. Li, "Simultaneous seismic data interpolation and denoising with a new adaptive method based on dreamlet transform," *Geophysical Journal International*, vol. 201, no. 2, pp. 1182–1194, 2015.

[14] W. Liu, S. Cao, Y. Chen, and S. Zu, "An effective approach to attenuate random noise based on compressive sensing and curvelet transform," *Journal of Geophysics and Engineering*, vol. 13, no. 2, pp. 135–145, 2016.

[15] Y. H. Yuan, Y. B. Wang, Y. K. Liu, and X. Chang, "Non-dyadic curvelet transform and its application in seismic noise elimination," *Chinese Journal of Geophysics*, vol. 56, no. 3, pp. 1023–1032, 2013.

[16] S. Beckouche and J. W. Ma, "Simultaneous dictionary learning and denoising for seismic data," *Geophysics*, vol. 79, no. 3, pp. A27–A31, 2014.

[17] Y. Chen, "Fast dictionary learning for noise attenuation of multidimensional seismic data," *Geophysical Journal International*, vol. 209, no. 1, pp. 21–31, 2017.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 184–199, Cham, 2014.

[19] C. Dong, C. C. Loy, and X. O. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 391–407, Cham, 2016.

[20] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, Las Vegas, NV, USA, 2016.

[21] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 294–310, Salt Lake City, UT, USA, 2018.

[22] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proceedings of the IEEE Conference on Computer Vision (ICCV)*, pp. 4809–4817, Venice, Italy, 2017.

[23] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pp. 136–144, Honolulu, HI, USA, 2017.

[24] Y. Tai, J. Yang, and X. M. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3147–3155, Honolulu, HI, USA, 2017.

[25] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: a persistent memory network for image restoration," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4539–4547, Honolulu, HI, USA, 2017, 2017.

[26] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018.

[27] Z. Hui, X. M. Wang, and X. B. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 723–731, Salt Lake City, UT, USA, 2018.

[28] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 527–542, Munich, Germany, 2018.

[29] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3867–3876, Long Beach, CA, USA, 2019.

[30] L. L. Canales, "Random noise reduction," *54th Annual International Meeting of SEG Technical Program Expanded Abstracts*, pp. 525–527, 1984.

[31] D. Bonar and M. Sacchi, "Denoising seismic data using the nonlocal means algorithm," *Geophysics*, vol. 77, no. 1, pp. A5–A8, 2012.

[32] Y. Chen and J. Ma, "Random noise attenuation by fx empirical-mode decomposition predictive filtering," *Geophysics*, vol. 79, no. 3, pp. V81–V91, 2014.

[33] W. Liu, S. Cao, and Y. Chen, "Application of variational mode decomposition in random noise attenuation and time frequency analysis of seismic data," in *In EAGE 78th Annual International Conference and Exhibition, Extended Abstracts*, pp. 1–5, Vienna, Austria, 2016.

[34] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, Fort Lauderdale, FL, USA, 2011.

[35] F. Aminzadeh, N. Burkhard, T. Kunz, L. Nicoletis, and F. Rocca, "3-D modeling project: 3rd report," *The Leading Edge*, vol. 14, no. 2, pp. 125–128, 1995.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5531222, 9 pages
https://doi.org/10.1155/2021/5531222

Research Article

Robust Visual Tracking Based on Convolutional Sparse Coding

Yun Liang, Dong Wang, Yijin Chen, Lei Xiao, and Caixing Liu

College of Mathematics and Informatics, South China Agricultural University, Guangzhou 501642, China

Correspondence should be addressed to Dong Wang; wdngng@163.com

Received 6 January 2021; Revised 2 February 2021; Accepted 28 February 2021; Published 24 March 2021

Academic Editor: Xiaojie Wang

This paper proposes a new visual tracking method by constructing the robust appearance model of the target with convolutional sparse coding. First, our method uses convolutional sparse coding to divide the interest region of the target into a smooth image and four detail images with different fitting degrees. Second, we compute the initial target region by tracking the smooth image with the kernel correlation filtering. We define an appearance model to describe the details of the target based on the initial target region and the combination of four detail images. Third, we propose a matching method by the overlap rate and Euclidean distance to evaluate candidates and the appearance model to compute the tracking results based on detail images. Finally, the two tracking results are separately computed by the smooth image, and the detail images are combined to produce the final target rectangle. Many experiments on videos from Tracking Benchmark 2015 demonstrate that our method produces much better results than most of the present visual tracking methods.

1. Introduction

Visual tracking is a hot topic in computer vision and graphics. The changes in background and object bring many tracking challenges such as deformation, occlusion, rotation, and so on. Now, it is still under well solved to produce accurate tracking results. Many tracking methods have been proposed recently. They are divided into two categories: generative tracking methods and discriminative tracking methods. The generative methods usually describe and identify the target with the maximum likelihood probability or posterior probability. The discriminative tracking methods often train a classification model to separate target and background.

The generative tracking methods find the candidate most similar to the target object as the tracking result. For example, Black and Jepson [1] proposed a subspace-based method to calculate the radiation transformation between the current frame and the image reconstructed with the feature vector of target. Later, Ross et al. [2] improved it by updating the basis of feature space online. Mei and Ling [3] solved the sparsity between the target template and the subspace composed of positive and negative trivial templates through the l_1 regularized least squares, which performed well in dealing with illumination variations and occlusions. Subsequently, many tracking methods [4, 5] have proposed to improve the tracking results by optimizing the l_1 algorithm. Although the generative tracking methods made a great breakthrough, they are limited by how to accurately separate the background and target, especially when dealing with clutter background and great deformation.

The discriminative methods often learn to distinguish target and background based on the cues from previous frames. For example, the tracker based on support vector machine (SVM) [6] distinguishes the target and background by learning positive and negative samples. Following the SVM, Hare et al. [7] proposed the structured support vector machine (SSVM) to further enhance its discriminating ability in dealing with deformation and occlusion challenges. Later, Ning et al. [8] proposed the dual linear structured support vector machine (DLSSVM) based on SSVM to produce efficient high-dimensional features of target and candidates.

The recent discriminative trackers are often defined by correlation filtering [9–15]. For example, Bolme et al. [9] proposed minimum output sum of squared error (MOSSE) tracking algorithm, which first applied correlation filtering in visual tracking. The MOSSE performs a convolution calculation on the Fourier domain between the target template and

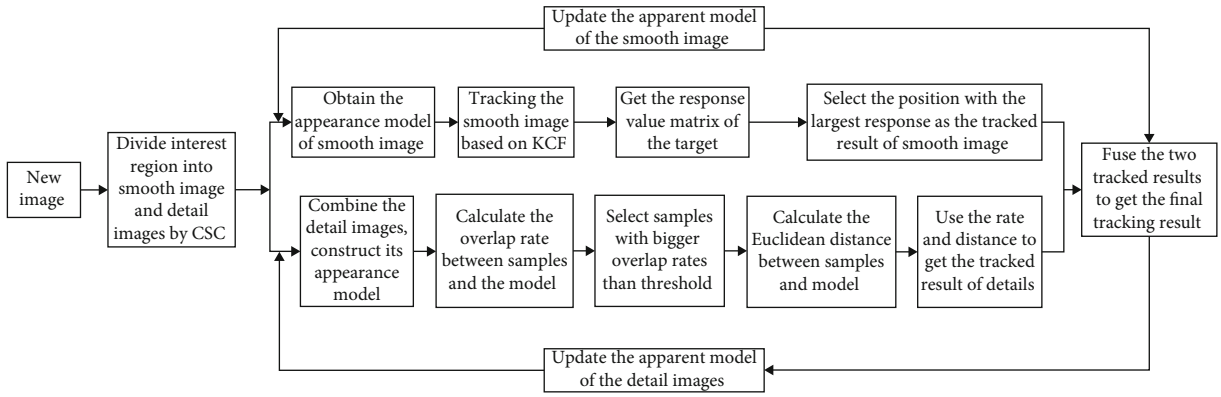


Figure 1: The flowchart of the proposed tracking method.

the interest region of target. Based on MOSSE, Henriques et al. [10] introduced the cyclic matrix and kernel method of tracking and convolving dense samples with the cyclic matrix formed by the target template in the Fourier domain. Then, they proposed the circulant structure kernel (CSK) tracking algorithm. Based on the CSK, Henriques et al. [11] introduced histogram of oriented gradient (HOG) feature to convert a single channel to multiple channels to improve the tracking results without increasing the time cost. Bertinetto et al. [12] integrated the features from both HOG and color cues to further improve tracking accuracy. Danelljan et al. [13] used the depth feature of single-layer convolution in CNN to replace the HOG feature in spatially regularized correlation filters to deal with tracking challenges. Danelljan et al. [14] improved the speed and stability of the algorithm based on the continuous convolution operators by reducing the model parameters and adopting a sparse update strategy. Valmadre et al. [15] used an end-to-end learning method to treat correlation filtering as a layer in CNN to reduce tracking drift and failure.

Recently, the tracking methods with deep features [16–19] have become very popular for their good performance in describing target and background. Li et al. [16] proposed a method to learn target perception features and integrate these features with Siamese matching network. Wang et al. [17] proposed a SiamFC-based tracker using "rough matching" and "fine matching." They enhanced tracking robustness through training in rough matching and improved discrimination through distance learning network in fine matching. Du et al. [18] proposed a tracker to detect target corners. It first uses the Siamese network to roughly distinguish the foreground and background to get the interest region of target. Then, they used the relationship between the target template and interest region to highlight the corner regions and enhance the features of the corner to produce a more accurate bounding box. Guo et al. [19] proposed a fully convolutional Siamese network for tracking. Chen et al. [20] proposed a Siamese box adaptive network structure named SiamBAN. The network views tracking as a parallel classification and regression problem, then classifies the objects, and regresses their bounding boxes in a fully convolutional network. Danelljan et al. [21] proposed a probabilistic regression model for tracking. Yang et al. [22] defined a tracking model by an offline recurrent neural optimizer to update the tracking model in a meta-learning setting. Li et al. [23] improved the tracking by integrating alignment data with deep features, then used ConditionNet to bridge the gap between the preconditioning and learning process.

The above tracking methods are mostly defined by establishing an appearance model of target. Therefore, when the target appearance model becomes not robust or inaccurate, it is very difficult to correct the model to improve the performance in tracking the following frames. Especially, for the updated target appearance model, if the online update ability of the model is too strong, it is easy to take the surrounding background as target information and introduce overfitting. However, if the online update ability of the appearance model is too weak, it leads to underfitting and tracking drift. This paper divides the target into a smooth image and four detail images based on convolutional sparse coding (CSC) and separately establishes target appearance models for them to track targets. The proposed tracker is achieved by combining the tracking results of the two parts to cope with challenges and improve the tracking performance.

2. Our Tracking Framework

This paper first extracts the interest region by expanding the target rectangle area by 2.5 times. Then, we divide the interest region into a smooth image and four detail images with different fitting degrees by CSC. For the smooth image, the model is initialized and tracked based on the KCF. For the detail images, we first combine the four detail images with different fitting degrees and construct the appearance model of the combined image to represent the target details. Then, we evaluate the candidates by measuring the overlap rate and Euclidean distance between the candidates and the appearance model. This evaluation describes that how much a candidate matches the appearance model, and the best-matched candidate is the tracked result based on detail images. Finally, the tracked results of the details and the smooth image are combined to produce the final tracking result. To suit the changes of the target, we update the appearance model with the tracked result frame by frame. The flowchart of our method is shown in Figure 1. It includes the target model initialization phase, target tracking, and model update.

2.1. Initialize the Appearance Model. As shown in Figure 1, we first extract the interest region based on the target rectangle of the last frame. Then, we divide the interest region into a smooth image and four detail images with different fitting degrees by the CSC. For the smooth image, we initialize its appearance model based on the KCF method. For the detail images, we combine the detail images with different fitting degrees and establish an appearance model for it to describe the target details.

2.2. Tracking Target Object. After producing the smooth image and detail images of the target, we track the two parts separately with different approaches. As described in the top row of Figure 1, for the smooth image, we use the KCF method to construct its appearance model and get the response value matrix. The tracking result based on a smooth image is selected as the candidate with the largest response value. Similarly, as shown in the bottom row of Figure 1, for the detail images, we first combine the four details and construct the appearance model. Then, we compute the overlap rates between samples and the appearance model and select the samples with bigger rate values. Later, we compute the Euclidean distance values between the selected samples and the appearance model. Then, we select the sample with the biggest overlap rate and minimum distance as the tracking result based on detail images. Finally, we fuse the two tracked results based on the smooth image and detail images to get the final tracking result.

2.3. Update the Appearance Model. The model update includes the appearance model update of smooth image and detail images. For the smooth image, an appearance model is first established according to the tracked result. Then, it is combined with the old one to form the updated appearance model. For the detail images, according to the new tracked result, four new detail models are extracted and then combined with different fitting degrees to replace the appearance model about target details to achieve model update.

3. Target Tracking Based on the CSC

In this section, we first describe how to divide an interest region into a smooth image and four detail images based on the CSC and how to establish the appearance models of smooth image and detail images, respectively, as shown in Figure 2. Second, we separately describe how to track the smooth image and the detail images in Section 3.2. Finally, we describe the details of the appearance model update in Section 3.3.

3.1. Initialize the Appearance Model for Smooth Image. Following the method proposed in [24], we divide the interest region of the target into the smooth part and the detail parts using N filters as shown in Figure 2. The smooth part contains the cues about the color and shape features of target. The detail parts describe the features about the image edge and texture structure. Equation (1) describes the separation

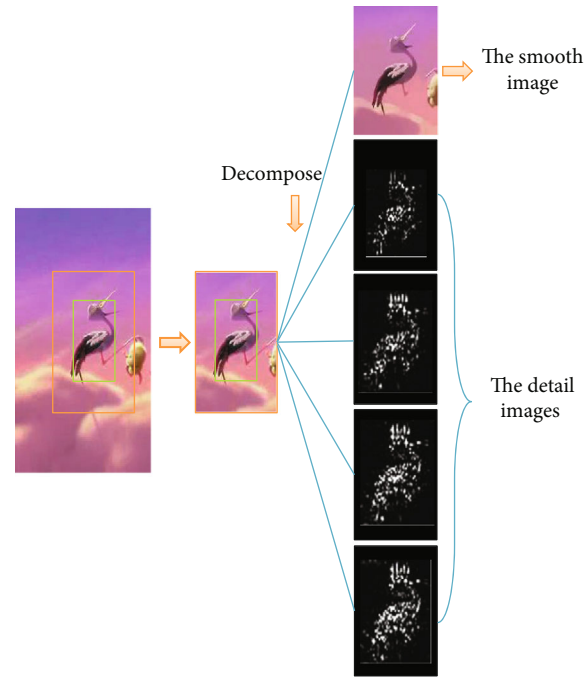


Figure 2: Divide the interest region into a smooth image and four detail images.

of smooth part and detail parts:

$$y = f^s \otimes Z_y^s + Y, \tag{1}$$

where y represents the original image, $f^s \otimes Z_y^s$ describes the smooth part, and Y describes the detail part. f^s is a low-pass filter and Z_y^s is a characteristic diagram.

As shown in Figure 2, the green rectangle describes the target region, and the red rectangle shows the interest region of target. As shown in the third column, it is divided into a smooth image and four detail images with different fitting degrees based on the CSC.

3.1.1. Initialize the Target Appearance Model of the Smooth Image. We construct the appearance model about the smooth image based on the KCF method. Therefore, the first step to initialize the tracking target is to construct a cyclic matrix $C(x)$ by extracting the target features. Then we diagonalize the cyclic matrix to obtain the diagonal cyclic feature matrix X using the Discrete Fourier transform, as shown in Equation (2).

$$X = F \operatorname{diag}\left(\widehat{x}\right) F^H, \tag{2}$$

where F describes the Constant Fourier matrix and satisfies $F F^H = 1$. $\widehat{x}$ is the generated vector after Fourier transform. We solve the least squares in the Fourier domain based on X to train the target detector $\widehat{\alpha}$, as described in Equation (3).

$$\widehat{\alpha} = \frac{\widehat{y}}{k\wedge^{xx} + \lambda}, \tag{3}$$

where $k\wedge^{xx}$ is the first row of the kernel matrix in the Fourier

domain, and $\hat{y}$ is the regression target training based on the Gaussian kernel function.

3.1.2. Initialize the Appearance Model for the Detail Images. As shown in Figure 3, this paper constructs the appearance model for the details of the target based on the detail images. This paper employs 400 filters to implement CSC. Therefore, it can get 400 detailed feature maps about tracking targets $(r_1 \sim r_{400})$. To prevent underfitting or overfitting, this paper combines every 100 feature maps to form a detail image. Therefore, we obtain four detail images by $R_1 = \sum_{i=1}^{100} r_i, R_2 = \sum_{i=101}^{200} r_i, R_3 = \sum_{i=301}^{300} r_i, R_4 = \sum_{i=401}^{400} r_i$, then four detailed models $(R_1 \sim R_4)$ with different fitting degrees are constructed to describe the details of target. Finally, we combine $R_1 R_1$, R_2, R_3, and R_4 to get the final appearance model of detail images named as R, as shown in Equation (4) where $\sum_{i=1}^{4} \lambda_i = 1$.

$$R = \sum_{i=1}^{4} \lambda_i R_i. \tag{4}$$

3.2. Tracking Target Based on CSC. This section introduces our tracking method based on the CSC in detail. For each frame of a video, we first use the CSC to divide it into the smooth image and four detail images. Second, we use KCF to track the target region based on the smooth image. Then, we predict the target region based on the appearance model of image details. Finally, we compute the final tracking result by combining the target regions based on both the smooth image and detail images.

3.2.1. Tracking Target Based on Smooth Image. We expand the target region from the last frame by 2.5 times to form the sampling area Z. We extract the features of the sampling area to form a feature matrix Z. Then, we calculate the kernel correlation values of the cyclic feature matrix X and Z using the kernel functions. It means that $K^x = C(k^{xz})$, where k^{xz} is a row vector from X and Z through the kernel function operation, and C is a function that constructs a cyclic matrix. K^{xz} is a circled function. After that, we use target detectors $\widehat{\alpha}$ and K^{xz} to do dot multiplication to get the response value matrix $\widehat{f}(z)$ of each position in the sampling area by Equation (5).

$$\widehat{f}(z) = \widehat{\alpha} \odot k\wedge^{xz}. \tag{5}$$

The sample with a larger response value matrix has more possibility to be used as the target. The position of the max response value is taken as the target center based on the smooth image.

3.2.2. Tracking Target Based on Detail Images. To predict the target based on detail images, we first randomly select 400 samples $(s_1 \sim s_{400})$ which has the same size as target in the interest region Z. Then, we calculate the overlap rate (OR) between each sample and the appearance model of detail images by Equation (6).

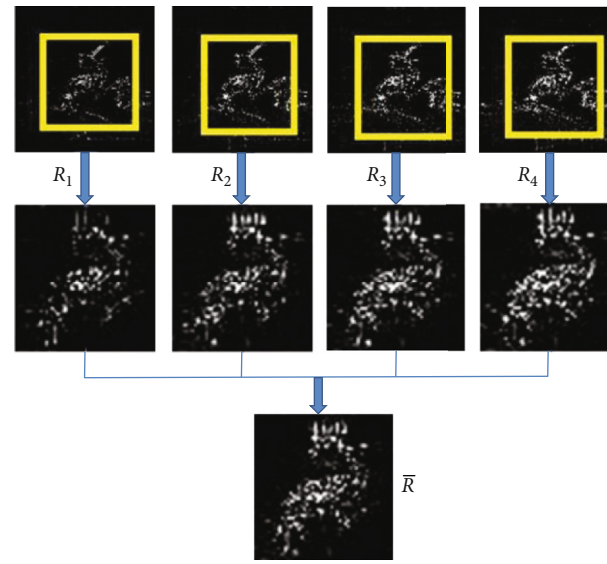


Figure 3: Constructs the appearance model for the detail images to describe the target details.

$$\begin{cases} \text{Width} = (W_s + W_t) - (\max(x_{sr}, x_{tr}) - \min(x_{sl}, x_{tl})), \\ \text{High} = (H_s + H_t) - (\max(y_{su}, y_{tu}) - \min(y_{sd}, y_{td})), \\ \text{OR} = \dfrac{\text{Width} \bullet \text{High}}{W_t * H_t} \times 100\%. \end{cases} \tag{6}$$

The Width is the width of the overlapping part, and the High is the high of the overlapping part. W_t and H_t are the width and height of the target model. W_s and H_s are the width and height of samples. (x_{tr}, y_{td}) and (x_{tl}, y_{tu}) represent the coordinate values of the down right point and the top left point of the target model. (x_{sr}, y_{sd}) and (x_{sl}, y_{su}) represent the coordinate values of the down right point and the top left point of the sample. When the overlap rate is greater than the setting threshold, the sample is retained; otherwise, the sample is rejected directly.

For the retained samples, we calculate the Euclidean distance between them and the appearance model of the detail images. First, for each sample, we extract four detail images with different fitting degrees accumulated by every 100 detail features. Then, we combine the four detail images to obtain the detail description of the sample, denoted by S. Finally, we calculate the Euclidean distance as EU between S and the appearance model of the detail part R by Equation (7).

$$\text{EU} = \sqrt{(R - S)^2}. \tag{7}$$

The sample with the minimum distance is taken as the final tracked result based on detail images.

3.2.3. Computing the Final Tracking Result. We use pos_s to describe the center of the tracked result based on the smooth image and use pos_r to describe the center of the tracked result

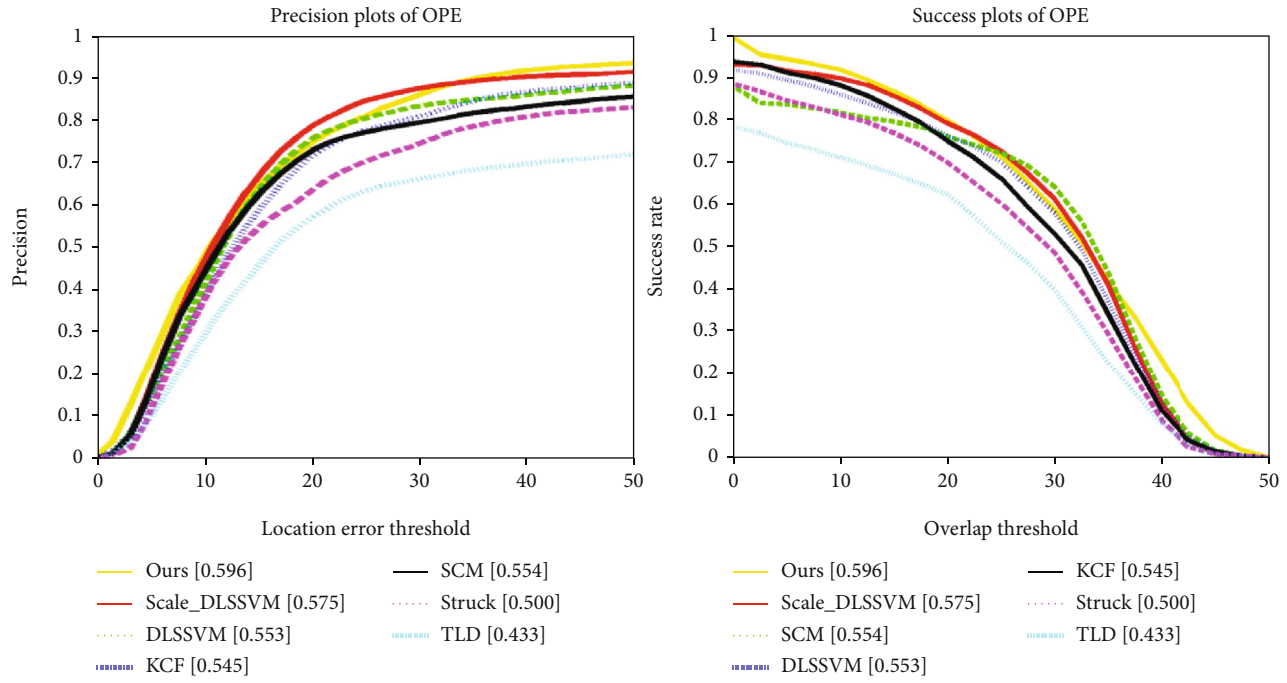


Figure 4: The overall rate of precision for center position error and success (for success rate).

Table 1: Tracking accuracy of 11 challenges where * is the first, ** is the second, and *** is the third.

Trackers	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Ours	0.658	0.781*	0.652	0.681**	0.712***	0.745**	0.636***	0.677**	0.713**	0.648**	0.708***
KCF	0.665***	0.757**	0.662***	0.676***	0.726*	0.710***	0.610	0.657***	0.700***	0.606***	0.712**
DLSSVM	0.705*	0.708***	0.752*	0.703*	0.714**	0.747*	0.747**	0.749*	0.755*	0.714*	0.736*
Struck	0.684**	0.596	0.719**	0.574	0.626	0.640	0.841*	0.639	0.649	0.606	0.672
TLD	0.558	0.537	0.616	0.469	0.581	0.604	0.515	0.542	0.590	0.537	0.608
VTD	0.346	0.449	0.306	0.451	0.482	0.583	0.559	0.509	0.594	0.434	0.584
CT	0.345	0.411	0.324	0.443	0.394	0.448	0.327	0.400	0.430	0.260	0.459

Table 2: The coverage of the success rate value for 11 challenges.

Trackers	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Ours	0.520***	0.605*	0.523	0.494***	0.523*	0.555*	0.369***	0.504**	0.520**	0.566**	0.462**
KCF	0.520***	0.588**	0.527***	0.497**	0.509***	0.525***	0.369***	0.486***	0.501***	0.530***	0.461***
DLSSVM	0.569*	0.560***	0.592*	0.526*	0.518**	0.545**	0.452**	0.563*	0.545*	0.609*	0.484*
Struck	0.549**	0.483	0.566**	0.430	0.458	0.474	0.482*	0.479	0.468	0.522	0.442
TLD	0.481	0.429	0.524	0.345	0.427	0.458	0.335	0.424	0.440	0.486	0.439
VTD	0.294	0.341	0.252	0.345	0.361	0.428	0.350	0.381	0.443	0.417	0.410
CT	0.287	0.316	0.233	0.352	0.282	0.323	0.126	0.292	0.308	0.280	0.310

based on detail images. The final tracked result is computed by combining pos_s and pos_r based on Equation (8).

$$pos_{tar} = \eta_1 pos_s + \eta_2 pos_r, \tag{8}$$

where $\eta_1 + \eta_2 = 1$. In our experiments, we set η_1 and η_2 both to be 0.5. The size of the tracked result also follows the same process of the center points of tracking results.

3.3. Update the Appearance Model of Target. Obtaining the tracking result on the current frame, we need to update the appearance model of the target to adapt to the changes of the target. This update is achieved by separately updating

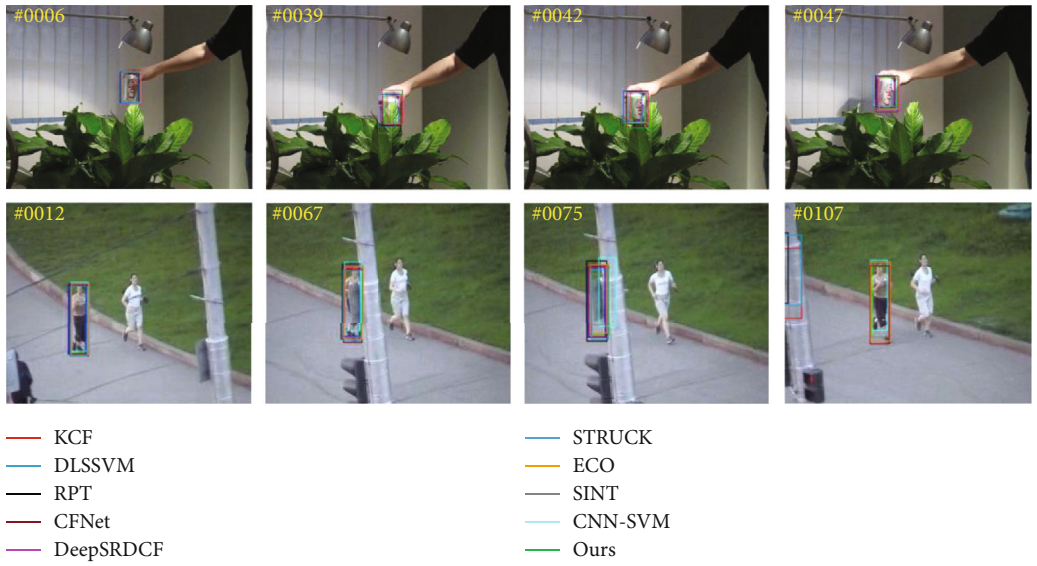


Figure 5: Comparison with occlusion challenge.

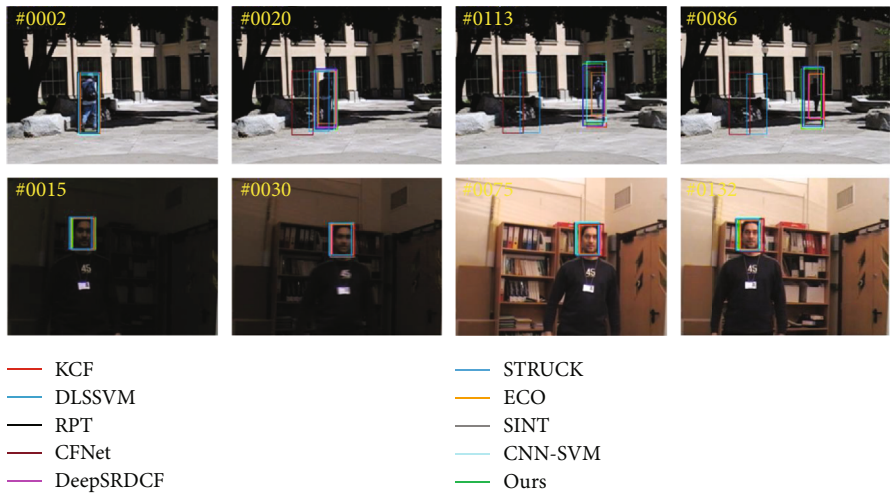


Figure 6: Comparison with illumination variation challenge.

the appearance model of the smooth image and the appearance model of the detail images.

3.3.1. Update the Appearance Model of the Smooth Image. The update of the model about the smooth image is achieved by updating the cyclic feature matrix X and the target detector $\widehat{\alpha}$. After getting the center position of target, the matrix X_t and the detector $\widehat{\alpha}_t$ on frame t are obtained. Then, we combine them with the cyclic feature matrix X_{T-1} and the target detector $\widehat{\alpha}_{T-1}$ at frame $t-1$ to get the updated X_t and $\widehat{\alpha}_t$. The equation to achieve the update is defined by:

$$\begin{cases} X_T = (1-\eta)X_{T-1} + \eta X_t, \\ \widehat{\alpha}_T = (1-\eta)\widehat{\alpha}_{T-1} + \eta\widehat{\alpha}_t, \end{cases} \tag{9}$$

where X_T and $\widehat{\alpha}_T$ are the updated cyclic feature matrix and the target detector on frame t, and η is the learning parameter. X_{T-1} and $\widehat{\alpha}_{T-1}$ are the cyclic feature matrix and target detector from the last frame, which greatly preserve the stable target feature from previous frames.

3.3.2. Update the Appearance Model of the Detail Images. The update of the appearance model of the detail images is achieved by updating the four models of detail images with different fitting degrees. First, we extract four new detail images with different fitting degrees based on the interest region of the tracked result. Then, we use the method to initialize the appearance model of detail images proposed in Section 3.1 to construct the new appearance model of detail images for the current tracked result. Finally, we update the appearance model of detail images by replacing the present model with the new one.

4. Results

Our method is implemented with MATLAB 2014b on the PC with Windows 7 system, Intel i7-6700 3.4GHz processor,

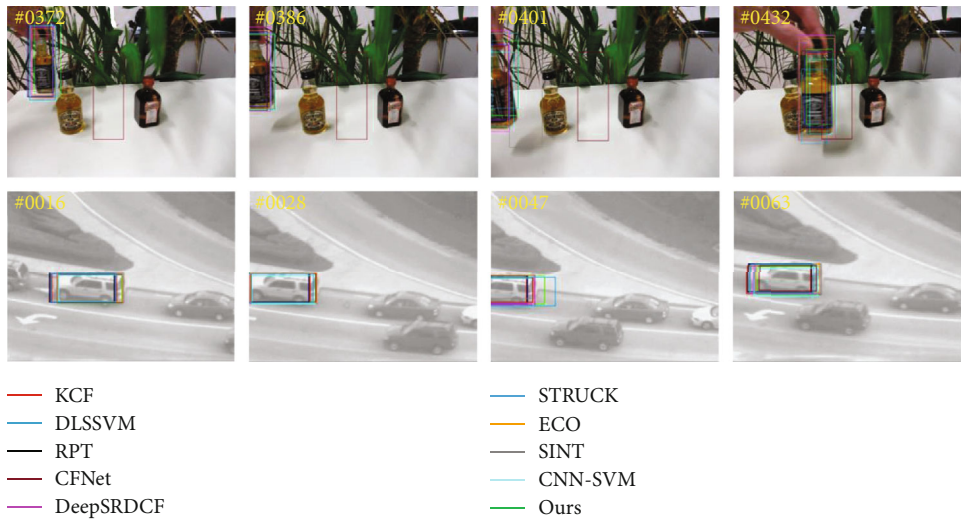


Figure 7: Comparison with out-of-plane rotation challenge.

12G video memory, and 12G memory. 61 video sequences from the Visual tracking Benchmark 2015 were used for experiments. They include several challenges such as complex background, illumination variation, and rotation. We do quantitatively and qualitatively evaluations on the famous present 13 trackers such as DLSSVM [8], KCF [11], Staple [12], ECO [24], CNN-SVM [25], CFNet [15], SINT [26], Struct [7], DeepSRDCF [14], RPT [27], TLD [28], VTD [29], and CT [30]. The results show that our method is more effective for in-plane rotation, complex background, and illumination variations.

4.1. Quantitative Evaluation. We evaluate the tracking results based on the accuracy of center position error and the success rate. The center accuracy is obtained by the center distance between the tracked result and the ideal target region, and the success rate is computed by the overlap rate of them. Compared with six target trackers, the center accuracy and the success rate of our method perform better, as shown in Figure 4. Tables 1 and 2 separately describe the evaluations for dealing with different challenges. For complex backgrounds, the center accuracy and the success rate value are ranked first with 0.781 and 0.605. For in-plane rotation, the center accuracy is 0.745 and the success rate value is ranked first with 0.555. For illumination variations, the center accuracy is 0.712 and ranks as the third, but relative to the first place KCF and the second DLSSVM differs only by 0.014 and 0.002. In addition, our success rate value is ranked first with 0.523. More details about the comparisons can be reviewed in Tables 1 and 2.

4.2. Qualitative Evaluation. This section qualitatively evaluates our method for occlusion, illumination variations, out-of-plane rotation, and so on. We compare the proposed method with nine famous algorithms.

4.2.1. Occlusion. Figure 5 shows the comparisons about target occlusion by video Coke and video jogging.1. The target is occluded by the surroundings, such as the leaves in Coke from frame 39 and the pole in Jogging.1 from frame 75. Many present trackers easily lose targets, such as Struct [7] and DLSSVM [8]. If the target is occluded completely, the background is taken as the target and finally leads to tracking drift or failure. We construct the appearance models of smooth image and detail images to describe the target, which greatly improve our performance in occlusion.

4.2.2. Illumination Variation. Figure 6 shows the comparisons among several trackers with illumination variation. We use the video Human8 (top row in Figure 6) and the video Man (bottom row in Figure 6) as examples. For video Human8, the illumination undergoes great changes. When people pass the shadow, the present trackers such as CFNet [15] and Struck [7] fail in identifying the blurred target. Using the proposed appearance model update scheme, our method performs much better than the present trackers by efficiently adapting to the changes of target appearance.

4.2.3. Out-of-Plane Rotation (OPR). Figure 7 shows the tracking results of various trackers under the OPR challenge. We use the video Liquor (top row in Figure 7) and video SUV (bottom row in Figure 7) as examples. From frame 386 to frame 401 in video Liquor, the target bottle rotates out of the plane for several times. From frame 28 and frame 47 in the video SUV, the fast movement of the car makes the camera unable to keep up, so part of target exceeds the image range. As shown in frame 386 on the top row and frame 47 on the bottom row, our method accurately detects the target region. However, some present trackers such as ECO [24] introduce track drifting. The main reason is our method using much detail information of the target to effectively separate the target and its surroundings.

5. Conclusion

This paper defines a new visual tracking method based on convolutional sparse coding. First, it extracts an interest region of the target in the current frame. Then by the

convolutional sparse coding, we divide the interest region into a smooth image and four detail images with different fitting degrees. For the smooth image, we initialise its appearance model and compute the general tracking result by kernel correlation filter. For the detail images, we first extract four detail images and combine them to initialize the appearance model for target details. We randomly sample 400 candidates in the interest region and calculate the overlap rate and Euclidean distance between each candidate and the appearance model of details to determine the tracking result based on the target details. By combining the tracking results of the smooth image and the detail images, we get the final tracking result of target. By introducing the appearance models of both the smooth image and detail images, the proposed method performs favourably in dealing with the tracking drift and failure introduced by the deformation and occlusion of target. We do quantitative and qualitative evaluations on some famous trackers on the Tracking Benchmark 2015. Many experiments demonstrate that our method produces better results in dealing with many challenges such as illumination variations, occlusion, and complex background.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of China (61772209), Science and technology planning project of Guangdong province (2019A050510034, 2019B020219001), the Production Project of Ministry Education China (201901240030), and the College Students Innovations Special Project of China under Grant 201910564037, 202010564026.

References

[1] M. J. Black and A. D. Jepson, "EigenTracking: robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[2] D. A. Ross, J. Lim, R. S. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[3] X. Mei and H. Ling, "Robust visual tracking using ℓ1 minimization," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1436–1443, Kyoto, Japan, September-October 2009.

[4] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient ℓ1 tracker with occlusion detection," in *CVPR 2011*, pp. 1257–1264, Colorado Springs, CO, USA, June 2011.

[5] D. Wang, H. Lu, and M. H. Yang, "Online object tracking with sparse prototypes," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 314–325, 2013.

[6] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[7] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[8] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4266–4274, Las Vegas, NV, USA, June 2016.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and L. Yui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, June 2010.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7575*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., pp. 702–715, Springer, Berlin, Heidelberg, 2012.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with Kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[12] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: complementary learners for real-time tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1401–1409, Las Vegas, NV, USA, June 2016.

[13] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 621–629, Santiago, Chile, December 2015.

[14] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9909*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 472–488, Springer, Cham, 2016.

[15] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5000–5008, Honolulu, HI, USA, July 2017.

[16] X. Li, C. Ma, B. Wu, Z. He, and M. Yang, "Target-aware deep tracking," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1369–1378, Long Beach, CA, USA, June 2019.

[17] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-tracker: series-parallel matching for real-time visual object tracking," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3638–3647, Long Beach, CA, USA, June 2019.

[18] F. Du, P. Liu, W. Zhao, and X. Tang, "Correlation-guided attention for corner detection based visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6836–6845, Seattle, WA, USA, June 2020.

[19] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for

visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6269–6277, Seattle, WA, USA, June 2020.

[20] Z. Chen, B. Zhong, G. Li, and S. Zhang, "Siamese box adaptive network for visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6668–6677, Seattle, WA, USA, June 2020.

[21] M. Danelljan, L. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7183–7192, Seattle, WA, USA, June 2020.

[22] T. Yang, P. Xu, R. Hu, H. Chai, and A. B. Chan, "ROAM: recurrently optimizing tracking model," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6718–6727, Seattle, WA, USA, June 2020.

[23] Y. Li, A. Bozic, T. Zhang, Y. Ji, T. Harada, and M. NieBner, "Learning to optimize non-rigid tracking," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4910–4918, Seattle, WA, USA, June 2020.

[24] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: efficient convolution operators for tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6931–6939, Honolulu, HI, USA, July 2017.

[25] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 597–606, Lille, France, July 2015.

[26] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1420–1429, Las Vegas, NV, USA, July 2016.

[27] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: robust visual tracking by exploiting reliable patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 353–361, Boston, MA, USA, June 2015.

[28] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: bootstrapping binary classifiers by structural constraints," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 49–56, San Francisco, CA, USA, June 2010.

[29] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, San Francisco, CA, USA, June 2010.

[30] K. Zhang, L. Zhang, and M. Yang, "Real-time compressive tracking," in *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7574*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., pp. 864–877, Springer, Berlin, Heidelberg, 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8856271, 10 pages
https://doi.org/10.1155/2020/8856271

Research Article

Q-Learning-Based High Credibility and Stability Routing Algorithm for Internet of Medical Things

Kefeng Wei[1,2] **Lincong Zhang**[3] **Xin Jiang,**[4] **and Yi Guo**[4]

[1]*School of Computer Science and Engineering, Northeastern University, Shenyang, China*
[2]*Shen Kan Engineering and Technology Corporation, MCC., Shenyang, China*
[3]*School of Information Science and Engineering, Shenyang Ligong University, Shenyang, China*
[4]*The Second Clinical Medical College of Jinan University, Shenzhen People's Hospital, China*

Correspondence should be addressed to Lincong Zhang; lincongz@foxmail.com

Received 26 September 2020; Revised 7 November 2020; Accepted 11 December 2020; Published 28 December 2020

Academic Editor: Xiaojie Wang

With the outbreak of COVID-19, people's demand for using the Internet of Medical Things (IoMT) for physical health monitoring has increased dramatically. The considerable amount of data requires stable, reliable, and real-time transmission, which has become an urgent problem to be solved. This paper constructs a health monitoring-enabled IoMT network which is composed of several users carrying wearable devices and a coordinator. One of the important problems for the proposed network is the unstable and inefficient transmission of data packets caused by node congestion and link breakage in the routing process. Based on these, we propose a *Q*-learning-based dynamic routing selection (QDRS) algorithm. First, a mathematical model of path optimization and a solution named Global Routing selection with high Credibility and Stability (GRCS) is proposed to select the optimal path globally. However, during the data transmission through the optimal path, the node and link status may change, causing packet loss or retransmission. This is a problem not considered by standard routing algorithms. Therefore, this paper proposes a local link dynamic adjustment scheme based on GRCS, using the *Q*-learning algorithm to select the optimal next-hop node for each intermediate forwarding node. If the selected node is not the same as the original path, the chosen node replaces the downstream node in the original path and so corrects the optimal path in time. This paper considers the congestion state, remaining energy, and mobility of the node when selecting the path and considers the network state changes during packet transmission, which is the most significant innovation of this paper. The simulation results show that compared with other similar algorithms, the proposed algorithm can significantly improve the packet forwarding rate without seriously affecting the network energy consumption and delay.

1. Introduction

In recent years, there are more and more kinds of diseases, which cause significant trouble for human beings and make people pay more attention to their health. Traditional medical treatment requires patients to go to the hospital and always takes a long time. The medical test results are usually time-consuming and inefficient. Many diseases need continuous monitoring for patients, but traditional medical treatment cannot achieve real-time observation and doctors' decision-making.

The emergence of wearable devices solves these problems. It allows people to monitor their health anytime and anywhere, thus promoting the Internet of Medical Things (IoMT) [1, 2]. The wearable device-based IoMT has attracted more and more attention and will become a trend that human beings pay attention to their health in the future. The IoMT can not only continuously monitor the physiological information of the human body through wearable devices but also transmit the detection results to the remote monitoring center or family doctor and even realize the emergency alarm. It is worth mentioning that the IoMT can help doctors propose treatment plans through decision support systems [3, 4]. This medical method can significantly reduce medical examination time, improve detection efficiency, and save human resources.

The outbreak of COVID-19 in 2019 makes people worldwide pay more attention to their health. The demand for monitoring, early warning, and transmission to doctors and family members using the IoMT is growing explosively. The mobility of users leads to the continuous change of network topology and also challenges the data transmission. Frequent user mobility will lead to link breakage and degrade network performance. At present, some scholars have studied the routing algorithms for IoMT [5–9]. However, current routing algorithms mainly consider user mobility's impact on algorithm performance, such as delay, network energy consumption, and network lifetime. If the link is not reliable, it is easy for data loss, retransmission, and other situations to occur, which pose a serious threat to the monitored personnel. Therefore, medical data need a stable and reliable transmission. Some scholars have studied the stability of link transmission to minimize the probability of link breakage [10]. Nowadays, the IoMT monitors not only patients but also the whole society with a wide range of user groups, different roles, and behaviors. Hence, the security of the multihop transmission of medical data becomes a great challenge. The author in [11] introduced the node activity in the routing algorithm, preferring to select the node with more connection times as the next hop to ensure the safe and reliable data transmission. However, the algorithm does not consider reducing the link break probability.

We proposed a routing algorithm based on comprehensive link stability, which can find the most stable link between the source node and the destination node, and provide reliable and durable communication between wearable users [12]. However, some problems have not been solved yet. First of all, the comprehensive link stability only considers the link connection duration. Then, the current congestion degree and the residual energy of nodes will also affect the link stability while not being considered. Finally, to pursue link stability, the algorithm allows too many hops.

The most important thing is that although we have established a reliable and stable path from the source node to the destination node, in the process of data forwarding, the state of intermediate forwarding nodes may change. For example, the new forwarding data from other nodes may lead to congestion for the node in the selected path. The moving trajectory of the node changes suddenly, which may bring a bad link even an interrupted link for the previously selected path. Therefore, it is necessary to adjust the path dynamically to adapt to the changing network environment. This paper proposes a Q-learning-based dynamic routing selection (QDRS) algorithm. Firstly, we establish the mathematical optimization model for routing selection. According to the connection duration, the credibility between the current node and its neighbor nodes, the residual energy, and the congestion degree of the neighbor nodes, the GRCS algorithm is proposed to select the optimal path. Node credibility is the number of times nodes communicate with each other. The higher the credibility, the more reliable the node and the more likely it is to provide reliable and stable forwarding. After that, this paper proposes a Q-learning-based local link dynamic (QLLD) algorithm to solve the congestion and link breakage in the path. The Q-learning algorithm is used to select the optimal next-hop node for each intermediate node and to modify the original optimal path in time to ensure the stability and reliability of the path.

The main contributions of this paper are as follows.

(1) This paper builds a wireless network named IoMT based on wearable devices and describes the problems of high credibility and stability in data transmission

(2) We formulate a mathematical model to maximize the credibility and stability of path and propose a global routing optimization routing algorithm with the constraints of node congestion degree, residual energy rate, credibility between nodes, and connection duration of link and hops

(3) To meet the requirement of a high packet forwarding rate under user mobility, we propose a local link dynamic adjustment method based on the Q-learning algorithm to locally select the optimal next hop for each intermediate node in the selected path. And the results are used to update the selected path. Thus, the waiting delay and transmission interruption because of node congestion and link breakage can be alleviated

The rest of this paper is organized as follows. The system model illustrates the network construction and related parameters in Section 2. Section 3 provides the problem formulation and the optimal path selection algorithm GRCS. Section 4 specifies a local link adjustment method using the Q-learning algorithm. The simulation and performance evaluation are described in Section 5. Finally, Section 6 concludes this paper.

2. System Model

Figure 1 shows that the Internet of Medical Things includes several users carrying a coordinator node and several wearable devices equipped with wireless sensors. These sensors can monitor the physiological information of different parts of the user's body (such as electroencephalograph (EEG), electrocardiograph (ECG), blood pressure, and body temperature) and the user's movement information (motion, including speed, direction, and acceleration) and surrounding environment information (temperature, humidity, toxic gas content, etc.). Each sensor periodically or suddenly sends data to the coordinator node according to the data characteristics by itself. The users' coordinator nodes can exchange or transmit information to the gateway node for remote transmission via the Internet. Therefore, real-time monitoring and early warning notifications of the user's physical health can be completed between family members or between the users and their family doctor or the hospital monitoring center. In this paper, we only consider the communication between the coordinators. We assume that the network includes N users (i.e., coordinators), and each user wears M sensor nodes and one coordinator. The gateway is randomly placed in the IoMT. Typically, to reduce energy consumption, the coordinator will send data to the nearest gateway node.

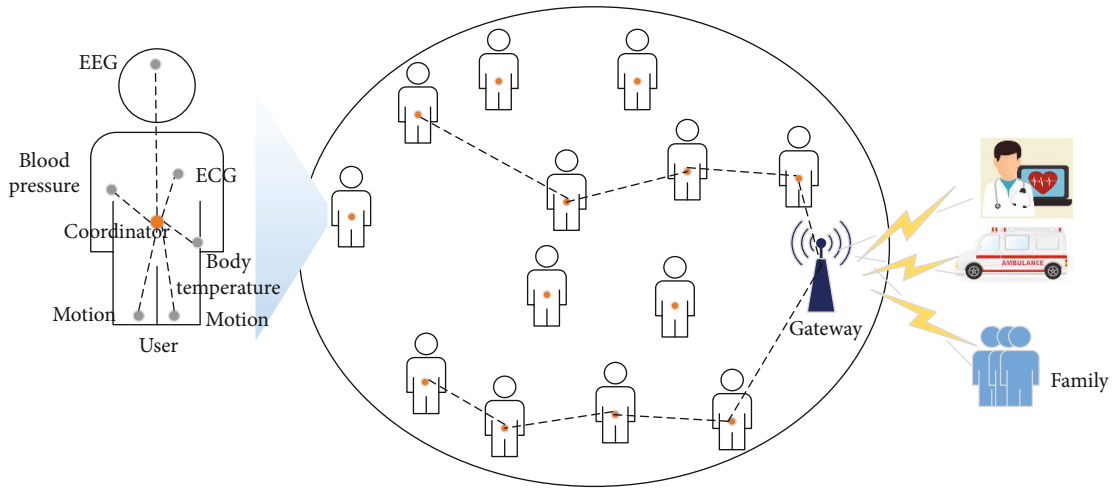


Figure 1: Internet of Medical Things.

We assume that there are H hops in the routing path between source node s and destination node d. To simplify the description of the problem, we introduce the following notations:

h:The index of hops,

i, j:Represented any two adjacent nodes of the path from s to d and $j = i + 1$,

(i, j):Described any link of the path from s to d,

$\tau_{ij}(h)$:The connection duration of the h_{th} link (i, j),

$\tau_{ij(\min)}$:In the path from s to d, the connection duration of the link with the shortest connection duration,

η:Weight coefficient,

m:The number of packets in the buffer of the h_{th} node,

k:The index of packets in the buffer of the h_{th} node,

p^k_{length}:The length of the k_{th} packet in the node buffer,

$Q_{\max}$:The maximum length of buffer queue,

E_{h_remain}:The residual energy of the h_{th} node,

$E_{\max}$:The initial energy of the h_{th} node,

N_{ij}:The number of connections between the node i and j,

N_{th}:The maximum value of the credibility,

$\alpha, \beta, \gamma, \omega$:Different weight factors.

To ensure the routing path's stability, the connection duration of the link denoted by $\tau_{ij}(h)$ is an important factor [12]. Hence, we define the link maintenance L_{sd} from s to d to measure the path's strength, as shown by

$$L_{sd} = \frac{(1-\eta)\tau_{ij(\min)} + \eta\left(\sum_{i=1}^{H}\tau_{ij}(h)/H\right)}{\sum_{i=1}^{H}\tau_{ij}(h)}. \quad (1)$$

The node load state is also a vital influence factor for link stability as the link may break when the packet waits for a too long time due to the high congestion of the node. The node congestion degree of the h_{th} node is represented by λ_h and computed by

$$\lambda_h = \frac{\sum_{k=1}^{m} p^k_{\text{length}}}{Q_{\max}}. \quad (2)$$

There is no doubt that the node's residual energy is not a negligible factor for link stability because the low-powered node may not finish the packet forwarding. The residual rate of power of the h_{th} node is represented by E_h and computed by

$$\xi_h = \frac{E_{h\_\text{remain}}}{E_{\max}}. \quad (3)$$

To ensure the safety of the packets, the forwarding node should be trustworthy and will not leak any information significant to medical knowledge. Therefore, we define the credibility to measure the safety of the forwarding node. The credibility of the node i and j denoted by R_{ij} can be computed by

$$\rho_{ij} = \frac{N_{ij}}{N_{\text{th}}}. \quad (4)$$

In conclusion, the credibility and stability of the path from the source node s to destination node d are denoted by CS_{sd} and can be computed by

$$\text{CS}_{sd} = L_{sd}{}^{\alpha} \frac{\left(\sum_{h=1}^{H-1}\xi_h\right)^{\gamma} \bullet \left(\sum_{i=1}^{H+1}\rho_{ij}\right)^{\omega}}{\left(\sum_{h=1}^{H-1}\lambda_h\right)^{\beta}}. \quad (5)$$

3. Path Selection with High Credibility and Stability

3.1. Problem Formulation. For the sake of better modeling the credible and stable routing path, we state the problem as follows:

$$\text{Maximize } \text{CS}_{sd}. \quad (6)$$

if $i == d$ **then**
  compute L_{sd} and CS_{sd};
else
  compute $d_{i,d}$, $\tau_{ij}(h)$, λ_h, ξ_h, ρ_{ij};
  if $d_{i,d} > d_{s,d} || \lambda_h \le \lambda_{th} || \xi_h < \xi_{th} || \rho_{ij} < \rho_{th} || H_{sd} > H_{th}$ **then**
    Reject forwarding;
  else if $\tau_{ij(\min)} > \tau_{ij}(h)$ **then**
    $\tau_{ij(\min)} \longleftarrow \tau_{ij}(h)$;
end if
    $h \longleftarrow h + 1$;
    $H_{sd} \longleftarrow H_{sd} + 1$
    add $\tau_{ij(\min)}$, $T_{ij}(h)$, λ_h, ξ_h, ρ_{ij} and H_{sd} into the RREQ packet and broadcast it.
  end if
end if

Algorithm 1: Judgment for the available node.

Subject to

$$d_{i,j} \le D_{\text{th}}, \tag{7}$$

$$d_{i,d} \le d_{s,d}, \tag{8}$$

$$\lambda_h \le \lambda_{\text{th}}, \tag{9}$$

$$\xi_h \ge \xi_{\text{th}}, \tag{10}$$

$$\rho_{ij} \ge \rho_{\text{th}}, \tag{11}$$

$$H_{sd} \le H_{\text{th}}. \tag{12}$$

The objective in (6) is to find the maximum CS by computing (5). The constraint in (7) states that any two communicable nodes should be in the transmission range D_{th}. Equation (8) indicates that the available node should be closer to the destination node than the source node. Equation (9) states that the single node is not very busy. Equation (10) implies that the available node has enough energy to forward packets. Equation (11) indicates the intimate and trustable relationship between two nodes, which should be larger than ρ_{th}. Equation (12) ensures that the path length is no longer than H_{th}.

3.2. GRCS Algorithm. To address the above problem, we propose a traditional algorithm named GRCS, which mainly focuses on selecting available nodes for each hop and delivering node information for each hop. The detail of the GRCS algorithm is as follows.

Step 1. Initialize the related parameters and add them to an RREQ packet. The source node broadcasts the RREQ packet to neighbor nodes in the transmission range. The destination node will reply an RRER packet to the source node after receiving this RREQ packet. Otherwise, go to Step 2.

Step 2. After the neighbor node j receives the RREQ packet from the upstream node i, it will check the information in the header fields of the RREQ packet. The detailed process for determination is shown in Algorithm 1.

Step 3. After receiving all RREQ packets in a period, the destination node computes L_{sd} and CS_{sd} for each RREQ packet, chooses the path with the largest CS_{sd} as the optimal path, and sends an RREP packet to the source node back to the way the RREQ came.

4. Q-Learning-Based Local Link Dynamic Adapting Algorithm

The data transmission starts after the path $p^*(s, d)$ has been established. However, the intermediate forward node receives the data not only from its upstream node in the path $p^*(s, d)$ but also from other nodes not in the path $p^*(s, d)$, i.e., it may congest. Meanwhile, the users may change their mind to go to another place and result in another motion trail. This sudden change will lead to link breakage. Therefore, we propose the QLLD algorithm to select an optimal node as the new forward node to improve the performance of the transmission.

The Q-learning algorithm is one of the frequently used methods of machine learning and has been used to solve the optimization problem in VANET [13–18], opportunistic networks [19–23], wireless sensor network [24–26], etc. In this paper, we adopt the Q-learning algorithm to select the optimal next-hop selection for the real-time correction for the path selected by the GRCS algorithm. We assume that the coordinator worn by each user is an agent. The cumulative revenue of the agent is affected by the next hop selected by other coordinators. In order to obtain the location, moving speed, direction, residual energy, link state, and additional information of other users, it is necessary to broadcast hello packets periodically between network coordinators for information exchange. The coordinator does not need to know the information of all the coordinators in the network but only needs to ensure that it can receive the information from its neighbors.

We define the neighbor nodes of coordinator x as those nodes which are closer to the destination node than node x and in its communication range. The set of the neighbor nodes of coordinator x is represented by $\mathfrak{N}_x$; y is one of the neighbor nodes of coordinator x:

$$\mathfrak{N}_x = \left\{ y \mid d_{x,y} < \min\left(d_{xd}, d_{\text{th}}\right), d_{yd} < d_{xd} \right\}. \tag{13}$$

Each coordinator maintains a neighbor node table, and each neighbor node is identified by the node congestion degree, residual energy rate, credibility, and connection duration.

System state: we define that the state $s_x(t)$ is decided by the location $\ell_x(t)$, represented by

$$s_x(t) = \left(\ell_x(t) \mid y \in \mathfrak{V}_x\right). \tag{14}$$

Action: the current node x selects the next-hop coordinator denoted by $a_x(t) \in \mathscr{A}_x$.

Reward: node x observes the system status $s_x(t)$; the direct reward obtained by implementing reflection $a_x(t) = \ell$ is $\mathscr{F}_x(t)$, represented by

$$\mathscr{F}_x(t) = \alpha_1 \lambda_y(t) + \beta_1 \xi_y(t) + \delta \rho_y(t) + \sigma \tau_{xy}(t). \tag{15}$$

Among them, α_1, β_1, δ, and σ are weighting coefficients and $\alpha_1 + \beta_1 + \delta + \sigma = 1$. The above reward function is defined as the sum of the congestion degree and energy residual rate of node y and the credibility and connection duration between node y and the current node x.

The long-term reward $\mathscr{R}_n^{\pi}(s)$ obtained by each coordinator is the expected value of the cumulative discount's direct reward, as shown in the following formula:

$$\mathscr{R}_x^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t \mathscr{F}_x(s(t), a(t)) \mid s(0) = s, a(0) = a_i \right]. \tag{16}$$

Among them, γ represents the discount rate, which determines the proportion of the direct reward and long-term reward, $0 \le \gamma < 1$; the greater the γ, the more significant the proportion of the direct reward.

The coordinator as an agent selects action based on the strategy π. Given any coordinator n, in the state s_t, the Q value obtained by selecting a_t according to a specific strategy $\pi(s_t, a_t)$ is defined as $Q_x^{\pi}(s_t, a_t)$. The strategy is evaluated by Q-learning, in which the Bellman equation is used to obtain the optimal Q value function, expressed in $Q_x^{\pi^*}(s_t, a_t)$, and the calculation is as follows:

$$Q_x^{\pi^*}(s_t, a_t) = \mathbb{E}\left[\left(R_x(s_t, a_t)\right) + \gamma \sum_{s_{t+1}} P_{s_t}^{s_{t+1}}(a_{x,t}) \max_{\pi^*} Q_x^{\pi^*}(s_{t+1}, a_{t+1}) \right]. \tag{17}$$

Among them, $P_{s_t}^{s_{t+1}}(a_{x,t})$ is the transition probability from state s_t to state s_{t+1}. The optimal strategy is defined as

$$\pi_x^*(s) = \max_{a_{t+1} \in \mathscr{A}_x} Q_x^{\pi^*}(s, a_{t+1}). \tag{18}$$

Q-learning iteration formula:

$$Q_x(s_t, a_t) = (1 - \alpha) Q_x(s_t, a_t) + \alpha \left[R_{nx}(s_t, a_t) + \gamma \max_{a_{t+1}} Q_x(s_{t+1}, a_{t+1}) \right]. \tag{19}$$

Among them, α is the learning rate and reflects the convergence speed of the iterative process.

For each intermediate forwarding node x, after the neighbor node y with maximum Q-value is selected, we will compare it with the downstream node of x in the original path. If they are not the same, then the downstream node of x is replaced with the selected neighbor node y. In each selected path, the intermediate coordinator node executes the QLLD algorithm and is denoted as node x. The QLLD algorithm is realized by Algorithm 2.

5. Performance Evaluation

The MATLAB software is employed as a simulation platform to verify the effectiveness of the proposed algorithm. In the simulated network, we deploy 80 randomly distributed wearable users in an 80 m × 80 m area. We set 5 sensor nodes and one coordinator on each user's body, and each user is viewed as a whole and represented by a dot in the network topology. Each user moves in any direction at a speed of 1 m/s. The transmission range D_{th} is set as 20 m. The initial energy of each node $E_{\max}$ is set as 100 J, and the threshold ξ_{th} is set as 20 J. The thresholds of node congestion degree λ_{th}, the credibility ρ_{th}, and the maximum hop H_{th} are 0.9, 1, and 7, respectively. The maximum length of the buffer queue $Q_{\max}$ is set as 2×10^5. Other parameters are the same as Ref. [12]. This paper compares the proposed algorithm with the traditional AODV algorithm, the RRLS algorithm [12], and our proposed GRCS algorithm.

First, we show a simulated routing path selected by the algorithms, as shown in Figure 2. From node 16 to node 25, the routing paths of AODV, RRLS, and GRCS are 16-22-25, 16-67-27-50-25, and 16-67-27-25, respectively. We can see that the approach of AODV is the shortest, and that of RRLS is with the most hops. The path of GRCS is similar to that of RRLS with a shorter length. The path selected by the proposed QDRS algorithm is the same as that of GRCS because QLLD do not have any effect on the path from node 16 to node 25, which is not marked in Figure 2. However, during the network running process, the advantages of QDRS can be shown according to the following results.

Figure3 illustrates the performance of the packet forwarding rate varying with running time. After the network begins to run, the data packet amount gradually increases. The congestion starts to occur in some intermediate nodes. When the congestion is serious, some packets may be discarded. And along with the mobility of users, the distances among users change varying time so the link state also

Initialize node list $\mathscr{L}$ for $p^*(s, d)$;
Initialize the location of nodes in $\mathscr{L}$;
for$i = 1; i < |\mathscr{L}|; i++$**do**
  $x = \mathscr{L}(i)$;
  $y^* = \mathscr{L}(i+1)$;
  $s_x(t) \longleftarrow \ell_x(t)$;
  Search for the neighbor nodes in communication range and put into $\mathfrak{N}_x(t)$;
  $s_x(t) \longleftarrow \ell_x(t)$;
  ifrand < ε**then**
    $a_x(t) \longleftarrow \text{argmax}[Q_x(s_t, a_t)]$;
  else
    $a_x(t) \longleftarrow argrand[Q_x(s_t, a_t)]$;
  endif
  if$a_x(t) \neq y^*$**then**
    $y^* \longleftarrow a_x(t)$;
  endif
endfor
for$i = 1; i < |\mathscr{L}|; i++$**do**
  $x = \mathscr{L}$(i);
  Update the network state and node state $s_x(t+1)$;
  Update $\mathfrak{N}_x(t+1)$;
  Update $Q_x(s_t, a_t)$ using (19);
endfor

Algorithm 2:Q-learning-based local link dynamic algorithm.

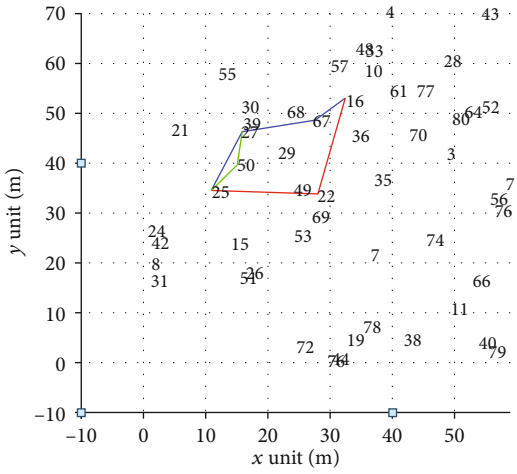


Figure 2: The routing path from node 16 to node 25.

becomes uncertain. By selecting the path in the shortest way, the AODV has the lowest packet forwarding rate. The proposed GRCS algorithm and QDRS algorithm obtain a higher packet forwarding rate because of the comprehensive consideration for node congestion, residual energy, credibility, and connection duration. In particular, on account of additional real-time correction for the selected path, the QDRS algorithm can keep the credibility and stability of the path. Therefore, the QDRS algorithm outperforms by 6.8%, 10%, and 64% compared with GRCS, RRLS, and AODV in terms of the packet forwarding rate.

Figure 4 illustrates the performance of the average path delay for each algorithm. The proposed QDRS algorithm updates the optimal path selected by the GRCS algorithm to reduce the waiting time brought by node congestion and break period caused by link failure. However, it still consumes some nonnegligible time for computing. Therefore, it just performs a little better than the GRCS algorithm for delay. What is more, we can see that QDRS and GRCS provide more stable path delay than AODV and RRLS.

As shown in Figure 5, the network energy consumption of the QDRS algorithm and GRCS algorithm is obviously lower than that of the AODV algorithm and a little more than that of the RRLS algorithm. This is because the residual energy of nodes is not the only consideration and is also not the optimization objective. Additional computing for Q -learning leads to higher energy consumption for the QDRS algorithm than the GRCS algorithm. However, without sacrificing too much energy, we promote the packet forwarding rate greatly.

In order to better verify the advantages of the proposed algorithm, we also simulate and compare the four algorithms changing with communication radius. This is because, in wireless networks, the communication radius is one of the key factors effecting the network performances.

Figure 6 shows the relationship between the packet arrival rate and the communication radius of the four algorithms. It can be seen from Figure 7 that the packet arrival rate of the four algorithms increases before the communication radius reaches 20 m. When the communication radius reaches 20 m, the packet arrival rate of the four algorithms is the maximum. Compared with AODV, RRLS, GRCS, and QDRS consider the residual energy of nodes and other factors, so the link is more stable, so they have better performance in the packet arrival rate. In addition, the performance of the GRCS and QDRS algorithms is better than that of the RRLS algorithm. When the communication radius

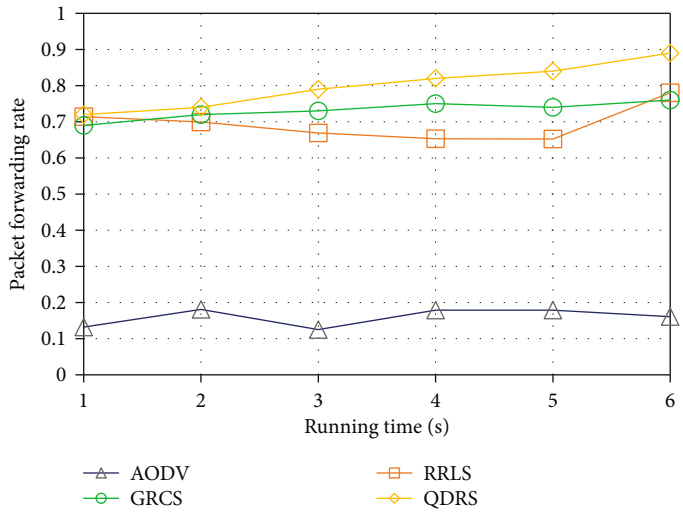


Figure 3: The packet forwarding rate.

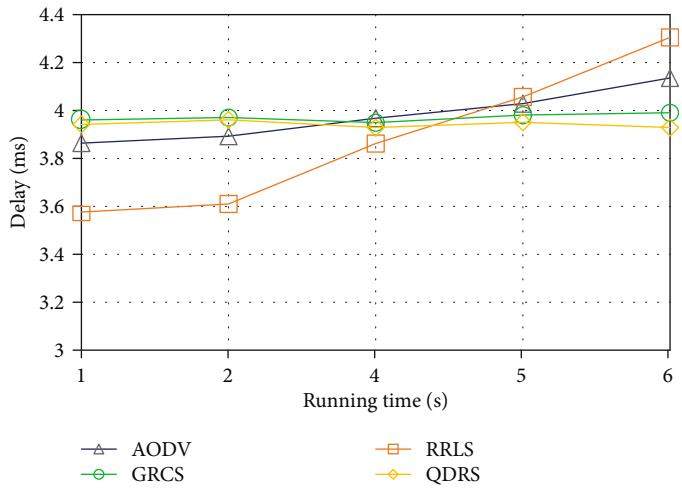


Figure 4: Comparison of path delay.

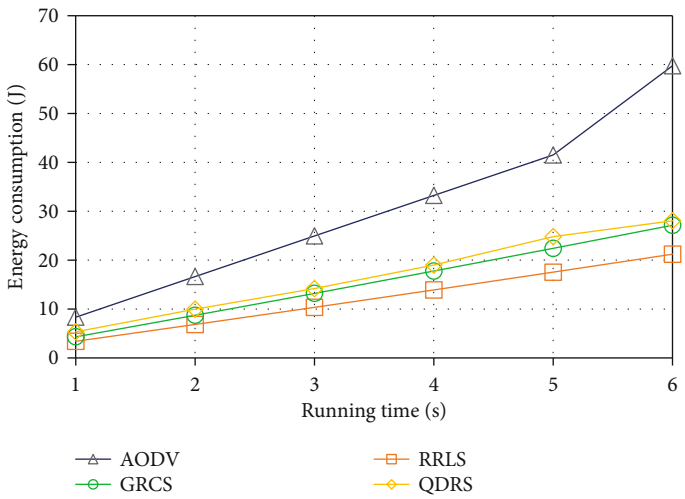


Figure 5: Energy consumption.

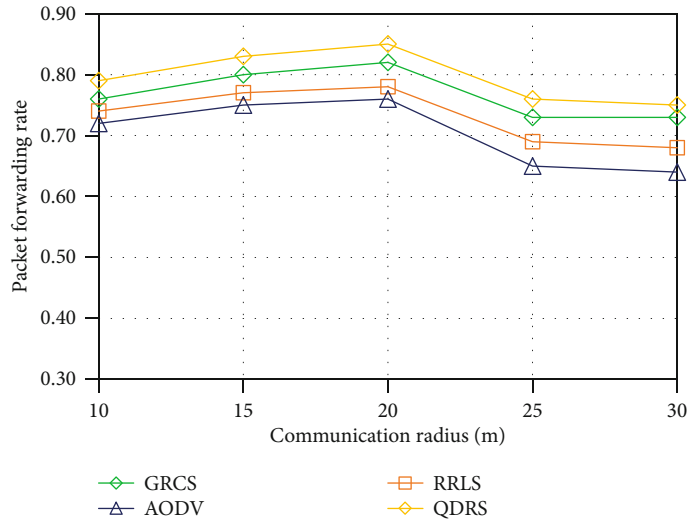


Figure 6: Packet forwarding rate varying with communication radius.

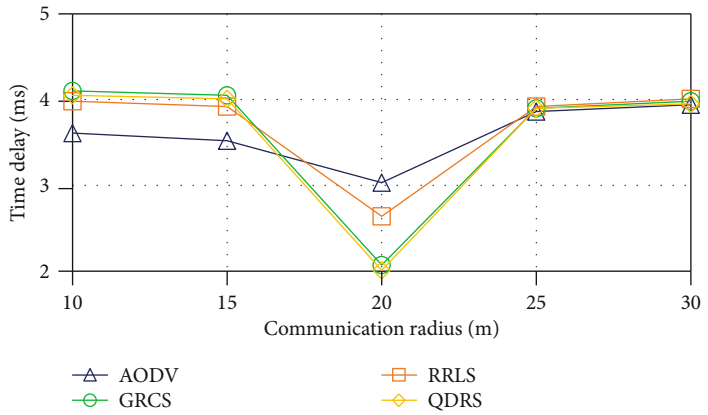


Figure 7: Delay varying with communication radius.

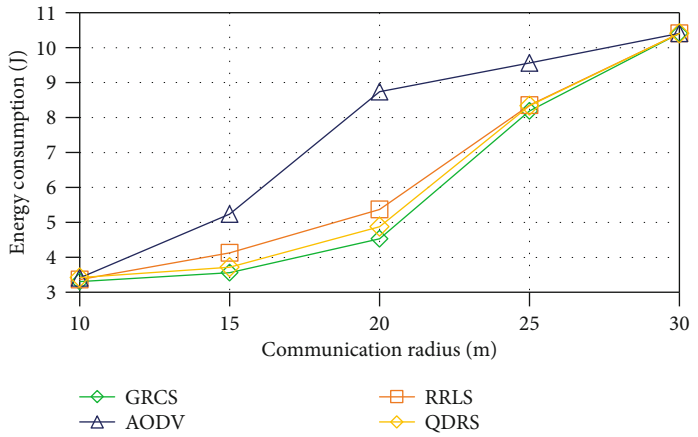


Figure 8: Energy consumption varying with communication radius.

continues to increase, it can be seen that the packet arrival rate of the three algorithms has decreased. The main reason is that with the increase of the communication radius, the number of nodes in the communication range increases, which makes the communication traffic of each node rise and cannot guarantee accurate transmission. Therefore, the packet arrival rates of the algorithms begin to decline, while the decline range is not very large. In general, our proposed QDRS still performs best among the four algorithms due to the path adjustment.

Figure 7 shows the relationship between the delay and the communication radius of the four algorithms. We can see that AODV has better delay performance when the communication radius is small because of the shortest path. With the increase of communication radius, the delays of four algorithms decrease before the communication radius reaches 20 m. When the communication radius is between 10 m and 15 m, the delays decline and the descent speeds of four algorithms are relatively gentle; while the communication radius reaches 15-20 m, the delays of the four algorithms decrease largely. When the communication radius reaches about 20 m, four algorithms achieve the optimal delay performance. Among them, the delays of GRCS and QDRS algorithms are almost the same and lower than the other two algorithms. When the communication radius continues to increase, the number of nodes in the communication range increases, so the traffic loads of nodes increase and then bring the increasing delay. In summary, the best communication radius for the four simulated algorithms is about 20 m.

Figure 8 shows the performance of energy consumption of four algorithms varying with the communication radius. It can be seen that with the increase of the communication radius, the energy consumptions of the four algorithms increase due to the increase of transmission power of nodes. Compared with the AODV algorithm, the growth trend of energy consumption of the other three algorithms is more gentle, where QDRS and GRCS have better performance than RRLS. Due to the modification of the path, some nodes need to establish a connection with the newly selected next hop, so the energy consumption of QDRS is slightly higher than that of GRCS. When the communication radius reaches 20 m, the energy consumption gap between the three algorithms and the AODV algorithm reaches the maximum. When the communication radius reaches 30 m, the energy consumption of the three algorithms tends to be consistent. This is because with the increase of the communication radius and no variation of node density, there is no big difference in the selection of the next-hop node among the four algorithms. Meanwhile, in order to maintain the communication range, more energy consumption is needed, so the energy consumption increases. In addition, due to the large communication range, the link breakage between nodes decreases; QDRS and GCRS have similar performance in energy consumption.

6. Conclusion

This paper studied the routing algorithm for IoMT and proposes a two-step solution to ensure the reliability and stability of network transmission. First, the credibility and stability of the path is the optimization goal, and the communication distance, node congestion, node residual energy rate, internode credibility, and internode hops are constrained to construct a mathematical optimization model, and the GRCS algorithm is proposed to find the optimal path. On this basis, the QLLD algorithm based on Q-learning is used to find the optimal next-hop node for the intermediate node to update the optimal path in time. Hence, it prevents the deterioration of the link status during the packet transmission and ensures the credibility and stability of the path. We use MATLAB to simulate the proposed algorithm, and the simulation results show the effectiveness of the proposed algorithm.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the National Natural Science Foundation of China, Grant/Award Number: 61501308, and the Postdoctoral Research Station project of Shenyang Ligong University.

References

[1] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for Internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, pp. 1–16, 2020.

[2] S. Huang, B. Guo, and Y. Liu, "5G-oriented optical underlay network slicing technology and challenges," *IEEE Communications Magazine*, vol. 58, no. 2, pp. 13–19, 2020.

[3] S. Durga, R. Nag, and E. Daniel, "Survey on machine learning and deep learning algorithms used in internet of things (IoT) healthcare," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1018–1022, Erode, India, 2019.

[4] T. Gao, X. Li, Y. Wu et al., "Cost-efficient VNF placement and scheduling in public cloud networks," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4946–4959, 2020.

[5] B. P. Santos, O. Goussevskaia, L. F. M. Vieira, M. A. M. Vieira, and A. A. F. Loureiro, "Mobile matrix: routing under mobility in IoT, IoMT, and social IoT," *Ad Hoc Networks*, vol. 78, pp. 84–98, 2018.

[6] W. Almobaideen, R. Krayshan, M. Allan, and M. Saadeh, "Internet of things: geographical routing based on healthcare centers vicinity for mobile smart tourism destination," *Technological Forecasting and Social Change*, vol. 123, pp. 342–350, 2017.

[7] S. Huang, C. Yang, S. Yin, Z. Zhang, and Y. Chu, "Latency-aware task peer offloading on overloaded server in multi-access edge computing system interconnected by metro optical networks," *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5949–5961.

[8] A. M. Fathollahi-Fard, A. Ahmadi, F. Goodarzian, and N. Cheikhrouhou, "A bi-objective home healthcare routing and scheduling problem considering patients' satisfaction in a fuzzy environment," *Applied Soft Computing*, vol. 93, article 106385, 2020.

[9] S. Yin, S. Huang, B. Guo et al., "Shared-protection survivable multipath scheme in flexible-grid optical networks against multiple failures," *IEEE/OSA Journal of Lightwave Technology*, vol. 35, no. 2, pp. 201–211, 2017.

[10] A. Serhani, N. Naja, and A. Jamali, "AQ-routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET-IoT systems," *Cluster Computing*, vol. 23, no. 1, pp. 13–27, 2020.

[11] Y. Liu, *Research on the Key Technology for Wireless Body Area Networks*, Beijing University of Posts and Telecommunications, 2017.

[12] L. Zhang, X. Chen, K. Wei, W. Zhang, and Y. Feng, "Body-to-body network routing algorithm based on link comprehensive stability," in *2019 28th Wireless and Optical Communications Conference (WOCC)*, pp. 1–5, Beijing, China, 2019.

[13] Z. Ning, P. Dong, X. Wang et al., "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[14] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2021.

[15] F. Khan and S. K. Nguang, "Location-based data delivery between vehicles and infrastructure," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 288–296, 2020.

[16] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing [J]," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2020.

[17] Z. Ning, K. Zhang, X. Wang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–12, 2020.

[18] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–14, 2020.

[19] B. Guo, S. Huang, Y. Shang et al., "Timeslot switching-based optical bypass in data center for intra-rack elephant flow with an ultrafast DPDK-enabled timeslot allocator," *IEEE/OSA Journal of Lightwave Technology*, vol. 37, no. 10, pp. 2253–2260, 2019.

[20] V. Vashishth, A. Chhabra, and D. K. Sharma, "GMMR: a Gaussian mixture model based unsupervised machine learning approach for optimal routing in opportunistic IoT networks," *Computer Communications*, vol. 134, pp. 138–148, 2019.

[21] D. K. Sharma, J. J. P. C. Rodrigues, V. Vashishth, A. Khanna, and A. Chhabra, "RLProph: a dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks," *Wireless Networks*, vol. 26, no. 6, pp. 4319–4338, 2020.

[22] D. K. Sharma, S. K. Dhurandher, I. Woungang, R. K. Srivastava, A. Mohananey, and J. J. P. C. Rodrigues, "A machine learning-based protocol for efficient routing in opportunistic networks," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2207–2213, 2016.

[23] D. K. Sharma, S. K. Dhurandher, D. Agarwal, and K. Arora, "kROp: k-means clustering based routing protocol for opportunistic networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1289–1306, 2019.

[24] G. Künzel, L. S. Indrusiak, and C. E. Pereira, "Latency and lifetime enhancements in industrial wireless sensor networks: a Q-learning approach for graph routing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5617–5625, 2019.

[25] Z. Zhuang, J. Wang, Q. Qi, H. Sun, and J. Liao, "Toward greater intelligence in route planning: a graph-aware deep learning approach," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1658–1669, 2019.

[26] W. Jin, R. Gu, and Y. Ji, "Reward function learning for q-learning-based geographic routing protocol," *IEEE Communications Letters*, vol. 23, no. 7, pp. 1236–1239, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6681391, 19 pages
https://doi.org/10.1155/2020/6681391

Research Article

MFCFSiam: A Correlation-Filter-Guided Siamese Network with Multifeature for Visual Tracking

Chenpu Li, Qianjian Xing, Zhenguo Ma, and Ke Zang

College of Biomedical Engineering & Instrument Science, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Zhenguo Ma; 850501@zju.edu.cn

Received 21 October 2020; Revised 12 November 2020; Accepted 12 December 2020; Published 24 December 2020

Academic Editor: Amr Tolba

With the development of deep learning, trackers based on convolutional neural networks (CNNs) have made significant achievements in visual tracking over the years. The fully connected Siamese network (SiamFC) is a typical representation of those trackers. SiamFC designs a two-branch architecture of a CNN and models' visual tracking as a general similarity-learning problem. However, the feature maps it uses for visual tracking are only from the last layer of the CNN. Those features contain high-level semantic information but lack sufficiently detailed texture information. This means that the SiamFC tracker tends to drift when there are other same-category objects or when the contrast between the target and the background is very low. Focusing on addressing this problem, we design a novel tracking algorithm that combines a correlation filter tracker and the SiamFC tracker into one framework. In this framework, the correlation filter tracker can use the Histograms of Oriented Gradients (HOG) and color name (CN) features to guide the SiamFC tracker. This framework also contains an evaluation criterion which we design to evaluate the tracking result of the two trackers. If this criterion finds the SiamFC tracker fails in some cases, our framework will use the tracking result from the correlation filter tracker to correct the SiamFC. In this way, the defects of SiamFC's high-level semantic features are remedied by the HOG and CN features. So, our algorithm provides a framework which combines two trackers together and makes them complement each other in visual tracking. And to the best of our knowledge, our algorithm is also the first one which designs an evaluation criterion using correlation filter and zero padding to evaluate the tracking result. Comprehensive experiments are conducted on the Online Tracking Benchmark (OTB), Temple Color (TC128), Benchmark for UAV Tracking (UAV-123), and Visual Object Tracking (VOT) Benchmark. The results show that our algorithm achieves quite a competitive performance when compared with the baseline tracker and several other state-of-the-art trackers.

1. Introduction

Visual tracking is a very fundamental and important research topic in computer vision. It is widely used in video surveillance [1], automonitoring [2], motion-based recognition [3], and many other fields. The main purpose of visual tracking is to solve the problem of target recognition and localization in a series of video image frames. Typically, given the labeled bounding box of the target in the first frame of a video, an ideal tracker should come up with this target's accurate position coordinates and mark it with a properly sized bounding box in each following frame of the video [4]. However, this seemingly simple task involves many difficulties that will lead to tracking failure if not properly addressed, such as obstacle occlusion [5, 6] illumination changing [7–9], deformation [10], size scale variations [11], and complex background clutter [12, 13].

To solve the problems listed above, a large variety of tracking approaches have been proposed over the years. Roughly, those approaches can be divided into two main categories: discriminative methods and generative methods. Discriminative methods [14–18] usually model the visual tracking task as a binary classification problem and train a robust classifier to distinguish the target from the background in every video frame. For example, in [11, 19, 20], all those three trackers use a support vector machine (SVM) as their main component in the visual tracking framework, and the SVM is a typical and classic discriminative

model which is widely used in machine-learning-related tasks. In [21], the authors design a tracker which combines local sparse descriptors into a boosting-based strong classifier using a discriminative appearance model. However, the main purpose of generative methods is to build up several appearance models of the target as templates and then search the video frame to find which region is most similar to the target's templates; this region is marked as the final tracking result. In [22], a consistent low-rank sparse tracker (CLRST) is designed on the basis of particle filter framework. The particle filter is a classic and typical generative model which is widely used in visual tracking. What is more, in the tracker from [23], the particle filter is used to build a redetection model, and this model is combined with a kernel correlation filter tracker to make it more robust. Other examples of generative models in visual tracking include trackers based on matrix decomposition [21, 22, 24] and those based on subspace learning [11, 19, 20]. In [24], an incremental nonnegative matrix factorization (INMF) method is used to address the visual tracking task. In [21], the authors combine the holistic and part-based representations with nonnegative matrix factorization (NMF) and model the target by a nonnegative combination of nonnegative components. In [11], the authors design a tracker which can efficiently adapt online information of the target's appearance by learning a low-dimensional subspace representation incrementally. Both the generative model and discriminative model are used in the tracking framework proposed by [19]. In [20], a subspace learning algorithm is used to impose joint row-wise sparsity structure on the target subspace. By this method, distractive information can be adaptively excluded.

Both the generative tracking models and the discriminative tracking models share the same key step: extracting powerful features from the target to represent it as distinctly as possible; those features are then used as references for tracking. Some traditional features include Histograms of Oriented Gradients (HOG), scale-invariant feature transform (SIFT), and color name (CN) [25, 26]. Most recently, with the boom in convolutional neural networks (CNNs) [27], many computer vision-related tasks have benefited from this and have achieved state-of-the-art performance [28–33]. CNN is a typical deep-learning architecture. Trained with a large set of image data, the CNN can learn to capture different levels of features owing to its multiple layers of convolution filters. Each filter can act as a specific feature pattern extractor, and combining them results in very powerful feature models. Recently, researchers have begun to integrate CNN into a visual tracking framework to try to explore the potential of deep features in this field [34–37]. In [34], the authors design a tracker using a single CNN to learn effective feature representations of the target object in an online manner. The tracker in [35] pretrains a CNN on a large set of videos to make sure the CNN can learn a generic target representation of the target. In [36], the authors adopt a tree structure to manage multiple target appearance models. They use multiple CNNs to estimate target states and determine the desirable paths for model update during tracking. Typically, Bertinetto et al. utilize the SiamFC [38] architecture and treat visual tracking as a general similarity-learning model that achieves the end-to-end workflow of tracking. This architecture achieves state-of-the-art performance and runs at about 80 fps on graphics processing units (GPUs), showing its significant potential in visual tracking.

However, in SiamFC's tracking process, only the features from the last CNN layer are used for visual tracking. The advantage of those high-level features is that they contain semantic information of the target that is very robust to appearance deformation. However, a drawback is that the semantic information lacks enough detailed texture information to distinguish the target from other same-category objects. That is to say, if there are other objects of the same type as the target in the search region, those objects would distract the tracker and cause the tracking to fail. What is more, when the contrast between the target and the background is very low—for example, as shown in Figure 1—the SiamFC also tends to drift. We can see that in the first 3 columns, when there exists other same-category objects in the search region, the score maps of SiamFC will produce large response on all these objects and this may lead the tracker to failure. And the fourth column shows that when the contrast between the target and the background is very low, the score maps of SiamFC will produce large response on the background. In other words, the deep features of this sequence even cannot provide effective information to distinguish the target from the background.

During our research, we found that the defect above is better addressed by correlation filter-based trackers. Some traditional handcrafted features such as HOG and CN are used in those correlation filter trackers, and their performances showed that these two features are very robust and effective when dealing with some complicated tracking environments. In our opinion, human's eyes are powerful trackers and we believe that the effective way to design a robust tracker is to follow the tracking logic of human eyes. When we use our eyes to track object, the main information we use contain two aspects: the target's contour information and color information. The HOG feature can represent the distribution of gradient and edge information in each local part of an object; thus, it is an ideal tool to describe the target's contour information. And the CN features are an ideal tool to describe the target's color information. What is more, the calculation of the HOG and CN features is very fast so they meet the requirement of real-time tracking. So, we believe that the HOG and CN features are the ideal instruments to compensate for SiamFC's shortcomings. Those features can express the detailed texture information of a target, and the information is usually the more visible traits that distinguish the right target from other objects in the search region. What is more, in each frame of a correlation filter tracker, a large number of samples are produced by cyclic sampling to train a robust classifier. This procedure guarantees that the correlation filter tracker is discriminative enough to distinguish the right target from other same-category distractors or a complex background. One another characteristic is that the search region's size in the correlation filter tracker is usually smaller than that in the SiamFC tracker. The benefit is that a smaller search region contains fewer objects so the tracker will not be likely to drift.

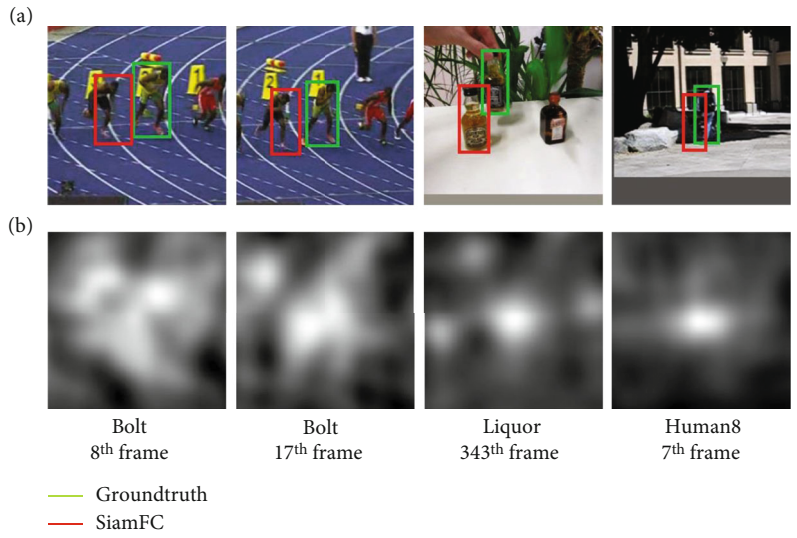


FIGURE 1: Several typical sequences of SiamFC's tracking failure from OTB100 dataset. (a) The original search region of SiamFC. (b) Their corresponded score maps, and the whiter areas on the score maps represent higher responses.

However, a smaller search region is more likely to lose the target if it moves fast.

So, to a certain extent, the advantages of the SiamFC tracker and CF trackers are complementary to each other. This provides the motivation for our research question: can we design a framework to make a correlation filter tracker and a SiamFC tracker work together? In this paper, we design a tracking framework that uses a correlation filter tracker to guide the tracking process of a SiamFC tracker. This framework is called a correlation-filter-guided Siamese network with multiple features (MFCFSiam). The main contributions of our work are summarized as follows:

(1) We design an effective criterion based on a correlation filter and the zero-padding method to judge which tracking results are more credible between a SiamFC tracker and a CF tracker. To the best of our knowledge, our algorithm is the first one which designs an evaluation criterion using correlation filter and zero padding to evaluate the tracking result

(2) We design a novel tracking framework that combines the advantages of a SiamFC tracker and correlation filter trackers. This framework works on the basis of the evaluation criterion in (1) and can effectively utilize both the semantic features from CNN and detailed texture features from traditional handcrafted feature extractors such as HOG and CN. Each kind of feature can make up for the disadvantages of other features, so the tracker can be more robust when faced with complicated tracking environments. Our framework shows an example of how to combine two trackers that share mutual advantages and make them complement each other in visual tracking

(3) We conducted a number of experiments to evaluate our proposed tracker on the dataset of Online Tracking Benchmark (OTB), Temple Color (TC128), the Benchmark for UAV Tracking (UAV-123), and the Visual Object Tracking (VOT) Benchmark. These datasets are very classic and typical in visual tracking, and the experiment results showed that our tracker achieved competitive performance when compared with baseline trackers and other state-of-the-art trackers

The rest of the paper is organized as follows. In Section 2, we introduce some related works in visual tracking. Then, we present our tracking framework in Section 3. In Section 4, we evaluate the performance of our tracker on the mainstream dataset and compare it with other representative trackers. Section 5 presents a summary of our work.

2. Related Work

2.1. Correlation Filter-Based Trackers. Recently, correlation filter-based trackers have attracted a great deal of attention due to their computational efficiency and competitive performance. These trackers [39] mainly focus on constructing a robust yet efficient appearance model of the target, which is called a correlation filter. Then, they sample several candidates around the search region and use them as inputs of the correlation filter. The filter will output each candidate's correlation score, and the one that gets the maximum response score is labeled as the final tracking result. Bolme et al. [40] first designed a correlation filter- (CF-) based tracker with a minimum output sum of squared error (MOSSE) filter, using raw pixels of images as inputs to train the correlation filters without any feature extraction. Henriques et al. [41] designed a CSK tracker using ridge regression and kernel tricks, but only utilized the gray features when training the filter, which limited the tracker's accuracy. Then, Danelljan et al. [25] integrated the color attribute into the CF tracker and improved its performance. In the KCF/DCF tracker [42] designed by Henriques et al., the feature

representation was extended into multichannel HOG and efficiently incorporated those features into the Fourier domain. It also proposed a kernel ridge regression model to accelerate its processing speed.

However, all the trackers listed above showed poor performance when the targets' size scale changed significantly. LCT [43] solved this problem by decomposing the tracking task into translation and scale estimation. Danelljan et al. [44] trained two kinds of filters to tackle the target's fast scale estimation—one for translation and one for scale estimation—and this DSST tracker enhanced the tracking performance significantly and showed a generic method to address the problem of scale estimation in visual tracking, while the tracker in [45] designs a metric learning function to solve the target scale problem. GFS-DCF [46] introduces a channel selection mechanism into CF-based trackers. This tracker is equipped with deep neural network features and the ability of joint feature selection and filter learning. The TRBACF [47] tracker designs a temporal regularization strategy which can efficiently adjust the model to adapt to the change of the tracking scenes, and this makes it more robust to complex environments. The ARCF [48] filter focuses on addressing the boundary effect problem in a correlation filter and adds restrictions to the alteration rate in response maps, so aberrances in detection can be largely suppressed, which makes the tracker more robust and accurate. The TFCR [49] designs a target-focusing loss function to alleviate the influence of background noise on the response map and improves the tracking accuracy.

2.2. CNN-Based Trackers. CNN-based trackers can be categorized into two main types. One uses the CNN as a single component in visual tracking and as a feature extractor to provide powerful features. For example, in HCF [10] and HDT [50], CNN was used to extract features instead of conventional handcrafted features. DeepSRDCF [51] employed the features extracted from shallow layers of CNN in a spatially regularized DCF tracking framework. All the above methods have one characteristic in common, that is, the CNN they used was always pretrained in some other task, such as image classification or target detection. In other words, they did not model the visual tracking as an end-to-end task and did not train the CNN specifically for visual tracking, so CNN's advantage in end-to-end tasks was not realized.

The other method is to model the visual tracking as an end-to-end task and train the CNN especially for tracking. Bertinetto et al. [38] considered visual tracking as a similarity-learning problem and designed a fully convolutional Siamese network (SiamFC) to evaluate the similarity between the target and the candidate search region. To some extent, this framework realized an end-to-end workflow specifically for tracking problems and achieved quite competitive performance. Following SiamFC, CFNet [52] added a correlation filter layer into the SiamFC network to extract features that are consistent with the CF layer. Guo et al. [53] designed the DSiam tracker, which added a component that combined two general transformations to represent target appearance and suppress noise.

With the development of the visual detection task, some researchers have tried to adopt the experience in visual detection to address the visual tracking problem. The SiamRPN [54] tracker introduces a region proposal network (RPN) from visual detection into visual tracking and designs a regression branch for a bounding box on the basis of SiamFC. So, this tracker's ability at target-scale estimation is obviously improved, but as its template is fixed during the tracking process, it is not so robust when the target's appearance changes quickly. The D3S [55] tracker addresses this defect by setting up two models to encode the target. One model is adaptive and discriminative while the other model is invariant to a broad range of transformations. The SiamAttn [56] tracker also focuses on improving the tracker's robustness to large appearance variations. It introduces an attention mechanism into SiamFC to improve the network's feature-learning capability and achieves more stable and accurate tracking. Cascaded RPN (C-RPN) [57] is another tracker that uses RPN to address the visual tracking problem. It focuses on solving the data imbalance during training and designs a hard negative sampling method to train the network. SiamRCNN [58] adapts Faster RCNN [59] on the basis of Siamese architecture and designs a tracker using the Tracking by Re-Detection framework. It uses Faster RCNN to generate region proposals and determines if those proposals are the same as the template target.

What is more, the RCT (real-time complementary tracker) in [60] is produced by combining SiamFC and CF-based trackers together in a series connection. In this tracker, the SiamFC is used to locate the target coarsely, and then, in the second stage, the derived coarse location is refined by CF-based trackers for higher accuracy. Actually, the design of our MFCFSiam tracker is mostly inspired by RCT. But we combine the SiamFC and CF-based trackers in a parallel connection, not a series connection. The CF-based tracker in our framework is used to guide the SiamFC by HOG and CN features. The disadvantage of RCT is that as the trackers are combined in a series connection, so if the SiamFC goes wrong in the first stage, then the CF-based trackers will be sure to lose the target. However, in our MFCFSiam, this disadvantage does not exist. The parallel connection can guarantee that the two trackers work independently, and the evaluation criterion we design can make the two trackers cooperate with each other more effectively. Next, we will introduce our MFCFSiam tracker in detail.

3. Proposed Algorithm

The tracking algorithm we propose in this paper is to design a framework to remedy the defects of SiamFC by combining it with a correlation filter tracker. This correlation filter can use the detailed texture information of the target such as HOG and CN features to guide the SiamFC tracker. We design a criterion to evaluate the validity of the two trackers' tracking results. If the evaluation shows that the correlation filter tracker's result is more reliable, our framework will use this result to adjust the SiamFC tracker. The overview of our algorithm's workflow is shown in Figure 2. Details will be described in the following sections.

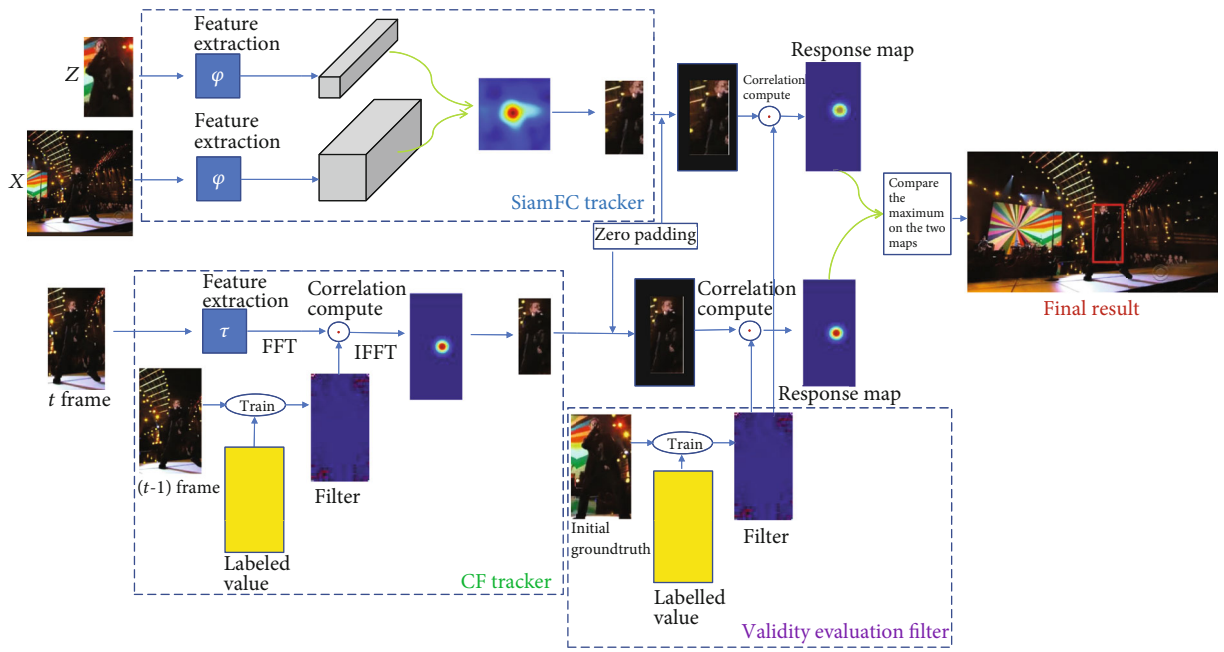


Figure 2: The basic workflow of our proposed tracking framework. The CF tracker based on HOG and CN features is used to guide the SiamFC tracker. The SiamFC tracker and the CF tracker produce their own tracking results. The validity evaluation filter uses the initial groundtruth in the first frame to generate a robust filter, and this filter is used to evaluate the two tracking results' validity. Correlation computation is conducted between this filter and each of the two results, and two response maps are produced. Finally, the result that has the bigger maximum on its response map is considered to be the final result, and this result is also used to update the SiamFC tracker.

3.1. The SiamFC Tracker Using Deep Features. The main architecture of SiamFC is made up of two branches: one branch is used to process the initial groundtruth annotated in the first frame of the image sequence—i.e., the initial template of the target, denoted as exemplar x, is used as the reference to judge whether an image pitch is the target or not—while the other branch is used to process the search region cropped from the frame that is to be searched, denoted as instance z.

The sizes of images input into the exemplar branch and instance branch are set to 127×127 pixels and 255×255 pixels, respectively. Inputs of the two branches will pass through the CNN and meet in the cross-correlation layers. In the cross-correlation layers, the feature maps from the exemplar branch are used as a slide window to compute similarity scores with all the subregions of the feature maps from the instance branch. Each similarity score is achieved by calculating

$$\text{Similarity}(z, x(i)) = \sum_{(m,n)\in S} \varphi(Z)_{(m,n)} \times \varphi(x(i))_{(m,n)}, \quad (1)$$

where $x(i)$ is the ith image pitch of exemplar x with the same size as z, φ represents the process of feature extraction, and $\varphi(z)_{(m,n)}$ is the pixel value vector with location coordinates (m, n) on z's feature maps. S denotes the whole area of each feature map. Thus, this correlation calculation procedure will generate a score map, and each pixel value on the score map records the similarity between an image pitch of instance x and exemplar z. Then, the maximal value of the whole score map is considered to be the target.

Regarding the training of the backbone convolutional neural network, an offline pretraining method with logistic loss function is used. The training data are pairs of images that are input into the instance branch and exemplar branch, respectively. Then, we choose the response score map from the last layer of the CNN and label each pixel with -1 or $+1$ according to the pixel's distance from the map's center; this binary-labeled score map is used as the groundtruth. The loss function is defined as

$$l(y, t) = \log\ (1 + \exp\ (-yt)), \quad (2)$$

where y is the groundtruth and t is the real-valued score map. Then, the final loss of the score map is defined as the mean of each pixel's loss:

$$L(y, t) = \frac{1}{|D|} \sum_{u\in D} l(y[u], t[u]), \quad (3)$$

where D is the whole area of the score map and $u \in D$ represents each position on it. Thus, the parameters of the CNN θ can be obtained by using the Stochastic Gradient Descent (SGD):

$$\arg\ \min L(y, t(z, x, \theta)). \quad (4)$$

3.2. The Guide Correlation Filter Tracker Using HOG and CN Features. As discussed in Section 1, features used in SiamFC

tracker are high-level semantic features from the last layer of CNN. Those features will not be robust enough if the search region contains other same-category distractors. What is more, another factor can make this situation worse: the large search area used in SiamFC. To some extent, a large search area can ensure that the right target is included in it even the target moves very quickly; that is the advantage. However, a large search area also makes it easier to introduce other distractors that may lead the tracker to drift. The location and size of the current frame's search region are determined by the tracking results of the previous frame. Once the tracker drifts to another distractor in one frame, it will be hard for the tracker to get back to the right target in the following frames.

To make a distinction between same-category objects, handcraft features such as HOG and CN are more effective because they always contain detailed texture information on the objects. As a correlation filter is an excellent model that can utilize HOG and CN features effectively in visual tracking, this motivates us to use a correlation tracker to guide the SiamFC tracker.

The features used in the correlation filter tracker in our framework are HOG and CN features. The HOG features are extracted by calculating the gradient information of an image. CN feature is another classic handcrafted feature that describes the color attributes of an image in a new space. Both the HOG and CN features can be calculated very efficiently. For an image X, its HOG and CN features can be concatenated and denoted as a multichannel feature map $x = [x_1, x_2, x_3 \cdots x_d]$, where each x_d in x represents a matrix with the size of M*N. Circular shift sampling is adopted along the M and N dimensions to generate a large number of training samples. The label value of each sample is generated by a Gaussian distribution according to its Euclidean distance to the target's coordinates. The label value map can be denoted as a M*N matrix y. In the training stage, the correlation filter f can be obtained by minimizing the cost function of ridge regression model in

$$\min \left\| y - \sum_{d=1}^{D} x^d * f^d \right\|^2 + \lambda \sum_{d=1}^{D} \left\| f^d \right\|^2, \tag{5}$$

where $*$ denotes the circular convolution and λ is the regularization parameter to control the model overfitting.

Equation (5) can be solved efficiently in each individual channel by FFT in the Fourier domain. The d-th channel of the filter f can be denoted as follows:

$$F^d = \frac{\bar{Y} \odot X^d}{\lambda + \sum_{k=1}^{D} \bar{X}^k \odot X^k}, \tag{6}$$

where $\odot$ represents element-wise multiplication; the capital letters represent the Fourier transformation of corresponding quantities, and the bar represents the complex conjugation.

In the detection stage, an image patch z is cropped from the new frame according to the tracking results of the previous frame and patch z is considered as the search region. Then, the target's location coordinates of the new frame are achieved by using the filter generated in Equation (6) to process patch z using Equation (7):

$$R = \mathscr{F}^{-1} \left(\frac{\sum_{d=1}^{D} A^d \odot Z^d}{B + \lambda} \right), \tag{7}$$

where R denotes the response score map, and the maximum value on R is considered to be the target's location in the new frame. A and B in Equation (7) represent the numerator and denominator in Equation (6), respectively. As the tracking process continues, both A and B are updated iteratively in each frame by the linear interpolation method, as shown in Equation (9):

$$A_t^d = (1 - \eta) A_{t-1}^d + \eta \overline{Y_t} \odot X_t^d, \tag{8}$$

$$B_t = (1 - \eta) B_{t-1} + \eta \sum_{k=1}^{D} \overline{X_t}^k \bigodot X_t^k, \tag{9}$$

where η represents the learning rate. The linear interpolation update strategy can make the tracker more robust to the target's appearance changes.

3.3. The Evaluation Criterion Based on Correlation Filter and Zero Padding. As discussed in Section 1, the high-level semantic information and large search area used in SiamFC can improve the tracking accuracy. On the other hand, they can lead the tracker to drift more easily. Therefore, we introduced a correlation filter tracker in Section 3.2 to remedy the defect. This correlation filter uses HOG and CN features to do visual tracking; those features contain detailed texture information so that the tracker is more robust when meeting other same-category distractors. In our proposed tracking framework, we used the tracking result from this correlation filter tracker to guide the SiamFC tracker. We designed a criterion to evaluate the validity of the tracking results from the SiamFC tracker and correlation filter tracker. If the evaluation shows that the results from the correlation filter tracker are more reliable, our framework will use it to replace the SiamFC's results. Correspondingly, the search region of the next frame in SiamFC tracker is also adjusted on the basis of the correlation filter tracker. In this way, the SiamFC's defects can be remedied.

The initial groundtruth in the first frame is the only template we can use when visual tracking begins. As the tracking goes on, the target's appearance can be influenced by illumination, occlusion, and many other factors, so the initial groundtruth is also the most reliable reference we can use. We chose the initial groundtruth to train a robust regression model, i.e., another correlation filter, to validate the tracking results. When the visual tracking begins, the initial groundtruth of the image sequence is input into the HOG and CN feature extractors. As discussed in Section 3.2, the output can be denoted as a multichannel feature map $t = [t_1, t_2, t_3 \cdots t_d]$, where each t_d in t represents a matrix with the size of M*N. We still adopted circular shift sampling along the M and N dimensions to generate the samples for the

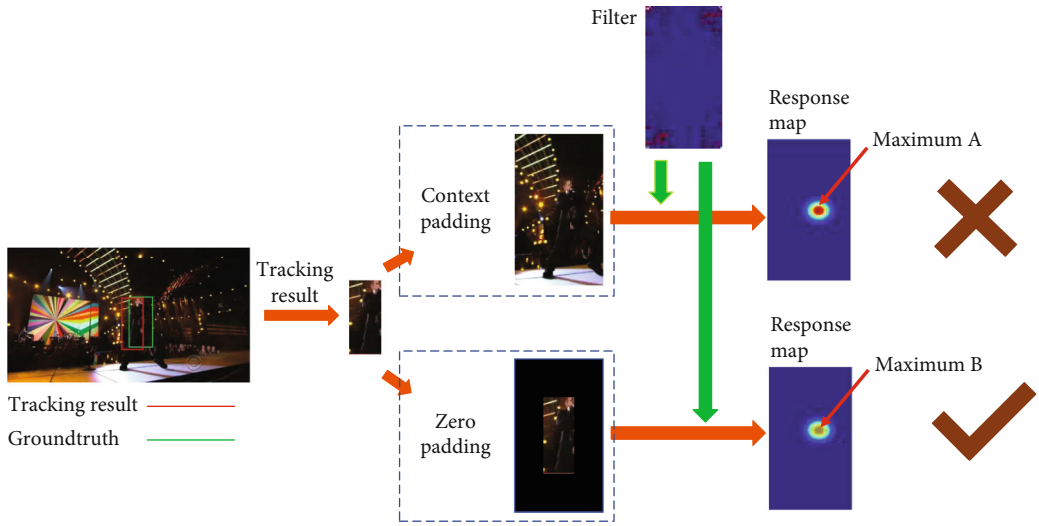


Figure 3: The comparison between the context padding and the zero padding in our proposed tracking framework. The tracking result is obviously not ideal. However, after the context padding, the whole target is still included in the padded bounding box, and it can still generate a significant response in the response map (maximum A). After the zero padding, the target outside the result bounding box is padded as zero. So, the maximum on the response map is compressed (maximum B). On the response map, a redder pixel represents a larger value, so maximum A is bigger than maximum B.

validation regression model. The label value of each sample is produced by a Gaussian distribution according to its Euclidean distance to the initial groundtruth's coordinates. For more details on training the regression model, see Section 3.2.

However, to make the correlation filter more robust and control overfitting, context padding is adopted before inputting the initial groundtruth into the feature extractor. When we use the correlation filter to process the tracking results from SiamFC tracker and CF tracker, the two results should also be padded. The traditional context padding used in CF-based trackers is to find the target's exact location so it can always produce a high response to the filter only if the target is contained in the image patch, whether the target's location is in the center or not. However, our purpose is to evaluate the validation of the two tracking results, so we must make sure that the more reliable tracking result is the one that contains the right target in the center. On the other hand, if the target is contained in the result bounding box, but not in the center, its response to the filter should be compressed.

As shown in Figure 3, the zero-padding method uses 0 to pad around the result bounding box. If the target is not in the center, some parts of the target will be outside the bounding box and the pixels' value in these parts will be set to zero. Then, we can use the correlation filter to process the padded result bounding boxes from the SiamFC and CF trackers. After this procedure, we could get their corresponding response maps. Then, we compared the maximum of each map, and the one that had the larger maximum was considered to be the more reliable tracking result. If this result belongs to the CF tracker, we will use it to adjust the SiamFC tracker.

4. Experiments

We conducted comprehensive experiments on the dataset of Online Tracking Benchmark (OTB) and Temple Color (TC128) to evaluate the effectiveness of our proposed tracking framework. All the experiments were implemented using Google's TensorFlow library. The platform we used to run the experiments is a Dell Alienware DESKTOP-N7K2SPB with a 3.70 Ghz Intel Core i7-8700K CPU and a NVIDIA GeForce GTX 1080Ti GPU. The operating system is 64-bit Windows 10 Professional. Our MFCFSiam tracker can realize real-time tracking with a speed of 16 FPS.

4.1. Benchmark and Evaluation Metric. Both the OTB [13] and TC128 are classic benchmarks designed especially to evaluate the trackers' performance in visual tracking. OTB has three subsets: OTB100 (OTB2015), OTB50, and OTB2013. OTB100 consists of 100 fully labeled video sequences that contain several different tracking scenarios such as scale variation (SV), low resolution (LR), illumination variation (IV), motion blur (MB), out-of-plane rotation (OPR), out of view (OV), background cluttered (BC), deformation (DEF), fast motion (FM), in-plane rotation (IPR), and occlusion (OCC). OTB50 and OTB2013 both consist of 50 video sequences, which are selected from OTB100. TC128 [61] is another famous benchmark used in visual tracking evaluation. It consists of 128 sequences of color images, which contain all kinds of complicated tracking environments. Both OTB and TC128 adopt the one-pass evaluation (OPE) protocol to evaluate the trackers' performance, which means using the tracker to process each image sequence from the beginning to the end only one time and then recording the tracking results to evaluate the tracker's performance.

We followed the metric introduced in OTB and TC128 to evaluate the trackers' performance. This metric contains a success plot and a precision plot, which are based on the Center Location Error (CLE) and Intersection Over Union (IOU), respectively. CLE compares the Euclidean distance

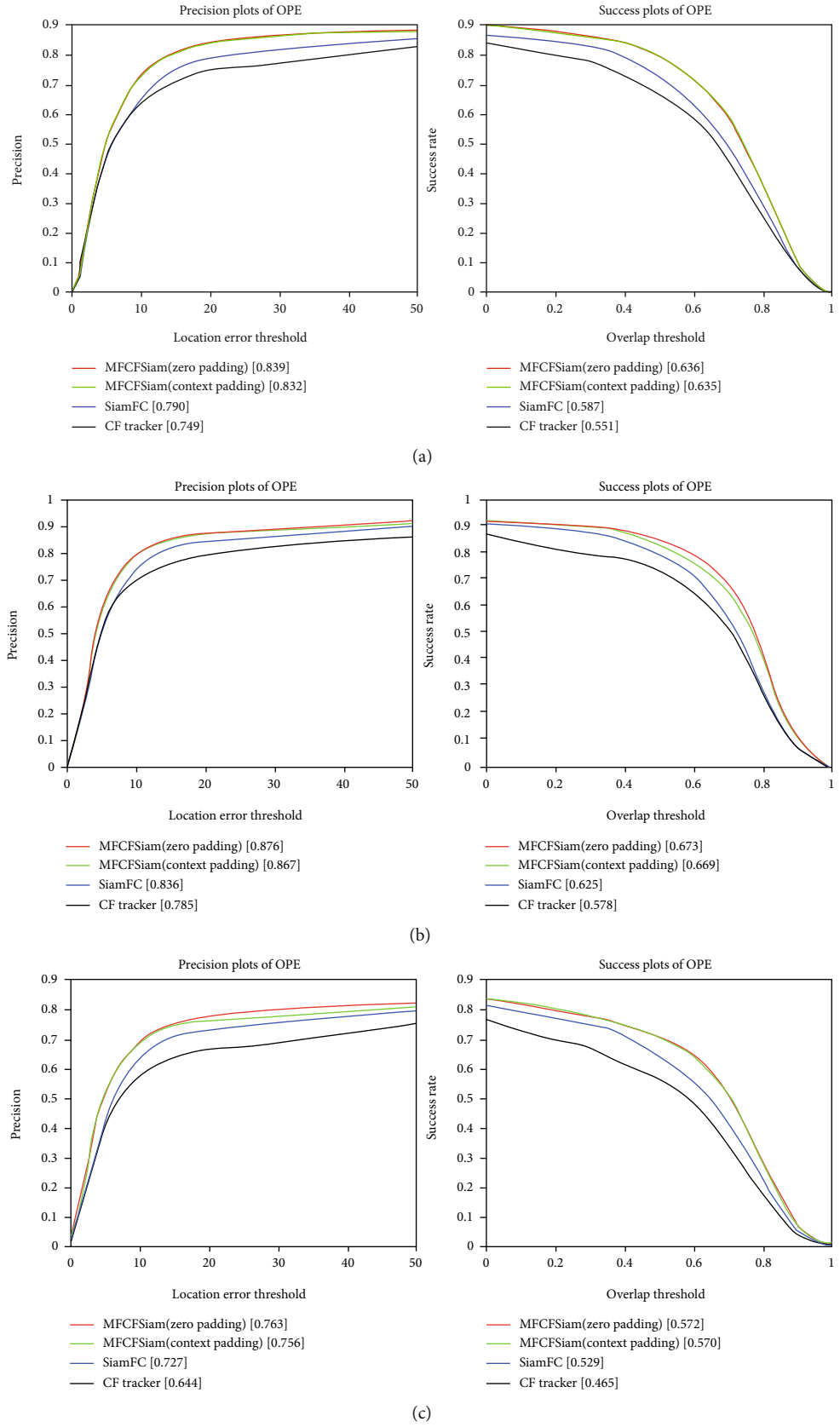


Figure 4: Comparison of the four trackers' performance on OTB dataset. Three plot pairs are results of (a) OTB100, (b) OTB2013, and (c) OTB50. This picture is best viewed on high-resolution displays.

Table 1: The four trackers' average precision values and average AUC values on the OTB dataset.

		OTB100	OTB2013	OTB50
Precision	MFCFSiam (zero padding)	*0.839*	*0.876*	*0.763*
	MFCFSiam (context padding)	**0.832**	**0.867**	**0.756**
	SiamFC	0.790	0.836	0.727
	CF tracker	0.749	0.785	0.644
AUC	MFCFSiam (zero padding)	*0.636*	*0.673*	**0.570**
	MFCFSiam (context padding)	**0.635**	**0.669**	*0.572*
	SiamFC	0.587	0.625	0.529
	CF tracker	0.551	0.578	0.465

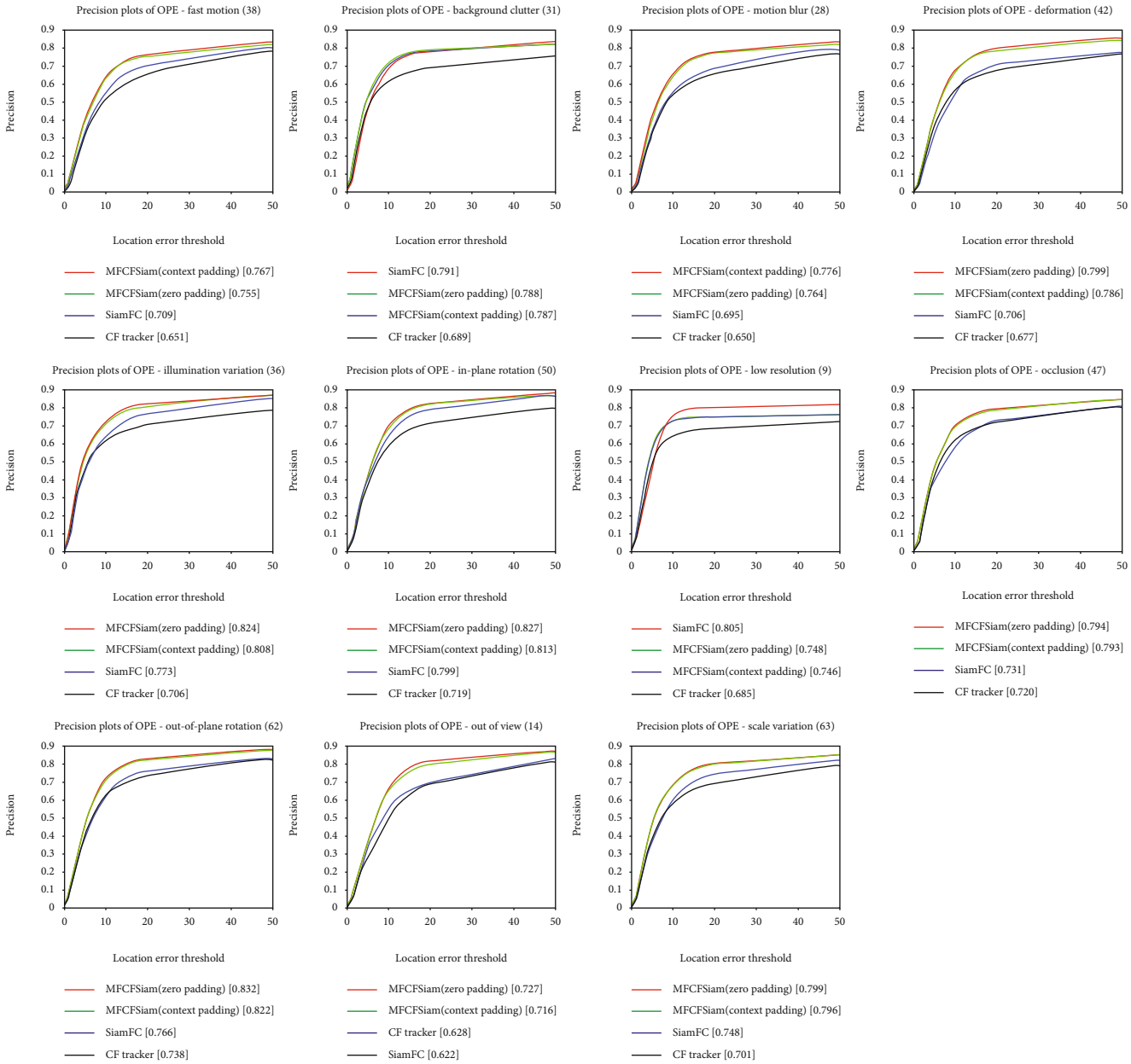


Figure 5: Comparison of MFCFSiam (zero padding) and MFCFSiam (context padding) and two baseline trackers using a precision-plot metric under the 11 tracking scenarios. This picture is best viewed on high-resolution displays.

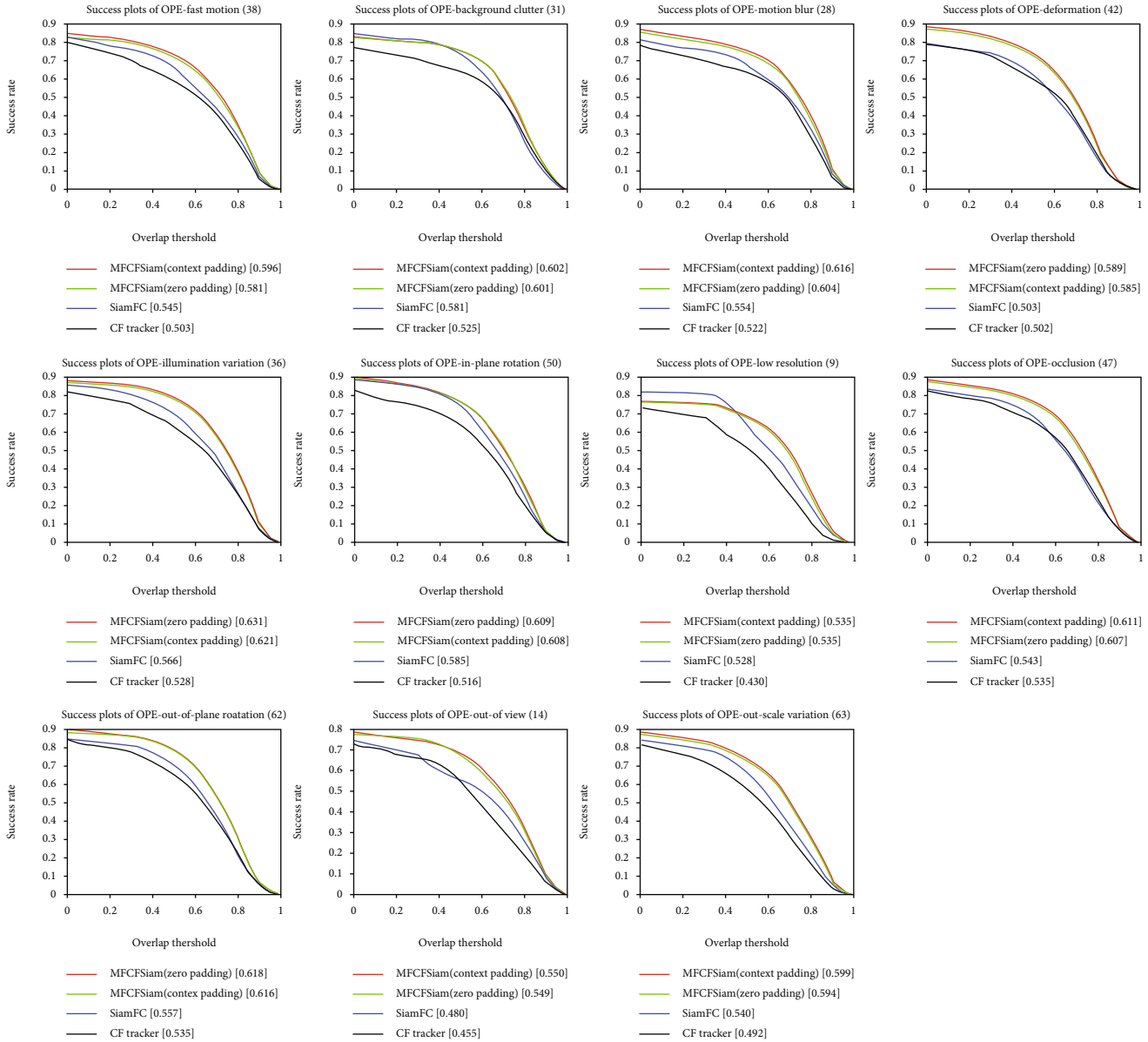


Figure 6: Comparison of MFCFSiam (zero padding) and MFCFSiam (context padding) and two baseline trackers using a success-plot metric under the 11 tracking scenarios. This picture is best viewed on high-resolution displays.

between the center locations of the tracking result provided by the tracker and the corresponding groundtruth of each frame with a given threshold to determine whether the tracking is successful. A smaller Euclidean distance in CLE denotes better tracking. IOU is defined as

$$\mathrm{IOU} = \frac{\mathrm{area}\left(R_T \cap R_G\right)}{\mathrm{area}\left(R_T \cup R_G\right)}, \tag{10}$$

where $\cap$ and $\cup$ are the intersection area and union area between the tracked results (R_T) and groundtruth (R_G), respectively. A parameter called the area under curve (AUC) value is used to represent the tracker's performance in IOU. A bigger AUC value denotes better tracking. So, the precision plot represents the percentage of successfully tracked frames based on the CLE, and the success plot represents the percentage of successfully tracked frames based on the IOU.

4.2. Ablation Experiment. In this section, an ablation experiment is conducted to evaluate the correctness of the tracking strategy proposed in this paper. Four trackers are used in this section: MFCFSiamFC (zero padding) tracker, MFCFSiamFC (context padding) tracker, SiamFC tracker, and CF tracker. The MFCFSiamFC (zero padding) and MFCFSiamFC (context padding) trackers are used to analyze the difference between zero padding and context padding proposed in the validity evaluation criterion in Section 3.3. The SiamFC tracker and CF tracker are used as the baseline trackers. We will run each of these two trackers separately

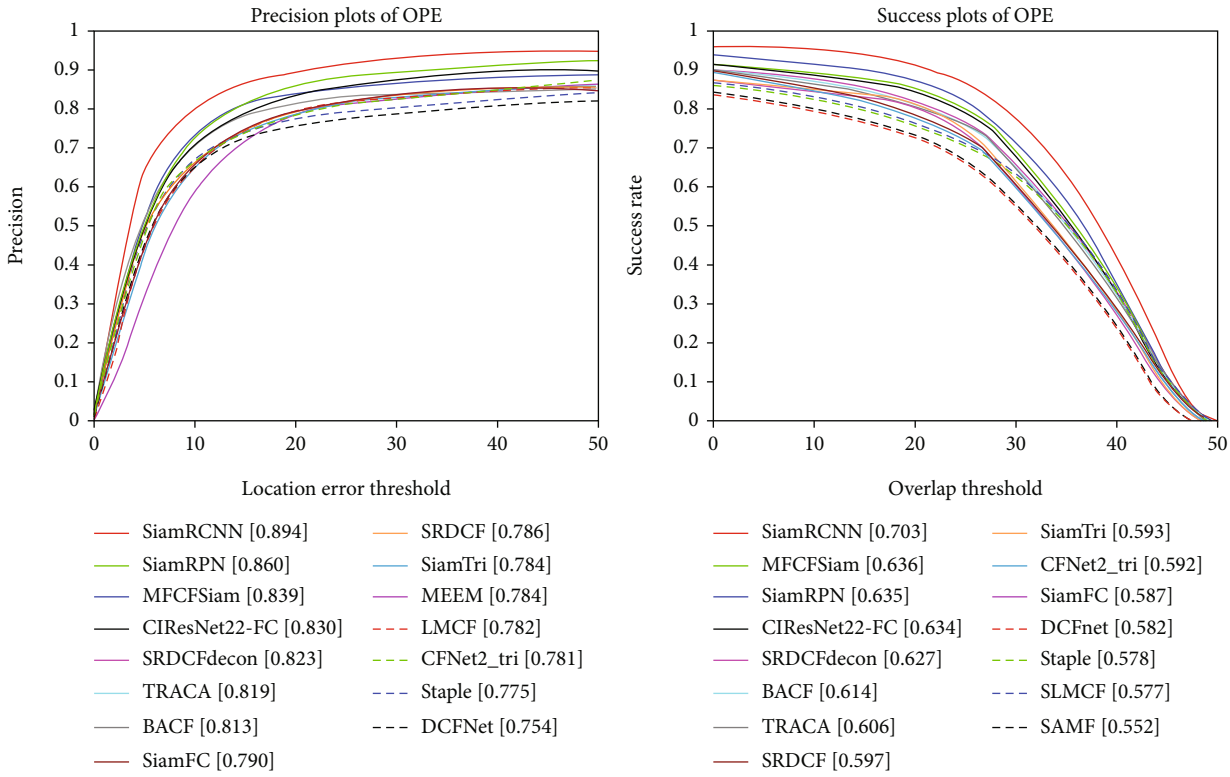


Figure 7: Comparison between our MFCFSiam tracker and 13 several state-of-the-art trackers on OTB100 dataset. This picture is best viewed on high-resolution displays.

on the OTB dataset; then, we can compare the baseline trackers' tracking results with our MFCFSiamFC tracker.

4.2.1. Overall Performance. Figure 4 shows the overall performance of all four trackers on the OTB dataset in terms of the precision based on CLE and success based on IOU. And the four trackers' average precision values and average AUC values on the OTB dataset are summarized in Table 1. The best performance is marked with italic, and the second-best performance is marked with bold. We see that both the MFCFSiamFC (zero padding) tracker and MFCFSiamFC (context padding) tracker outperform the baseline trackers (SiamFC tracker and CF tracker), no matter which dataset is used. This proves the correctness and effectiveness of our tracking theory. The CF tracker alone does not show very good performance; it ranks last in all the six plots in Figure 4 and the SiamFC ranks the third. However, our MFCFSiam tracker, which combines the two trackers into one single framework, has showed obvious better performance in all the six plots. When the CF tracker using HOG and CN features is regard as the guide for the SiamFC tracker, which uses the feature maps from the last layer of the CNN, the advantages of both detailed texture information and high-level semantic information are combined together effectively. That is the reason why our MFCFSiam tracker can make those improvements. What is more, the MFCFSiamFC (zero padding) tracker ranks first in five (OTB100, OTB2013, and precision plot of OTB50) out of the six plots in Figure 4, which proves that the zero-padding method is more robust than the context-padding method in the validity evaluation criterion.

4.2.2. Scenario-Based Performance. A tracker's performance can be influenced by many factors such as deformation, scale variation, and illumination. To evaluate the tracker as a whole, the OTB dataset divides all the video sequences into 11 kinds of tracking scenarios. Each scenario represents one crucial factor that may influence the tracker's performance, i.e., scale variation (SV), low resolution (LR), illumination variation (IV), motion blur (MB), out-of-plane rotation (OPR), out-of-view (OV), background cluttered (BC), deformation (DEF), fast motion (FM), in-plane rotation (IPR), and occlusion (OCC).

We further evaluated and compared the four trackers' performances under the 11 annotated tracking attributes on OTB100 separately. Figures 5 and 6 show the results. We found that in the success plots, our MFCFSiamFC tracker outperformed the two baseline trackers in all 11 different tracking attributes, and in the precision plots, our MFCFSiamFC tracker outperformed the two baseline trackers in nine out of the 11 different tracking attributes (except for BC and LR). The results show that our tracking strategy has made obvious improvements. Because the targets in BC tracking attribute usually have complicated background and the background usually have abundant and complicated texture information, so sometimes our tracker may drift to the background a little. As for the LR tracking attribute, the poor resolution of the targets may lead that

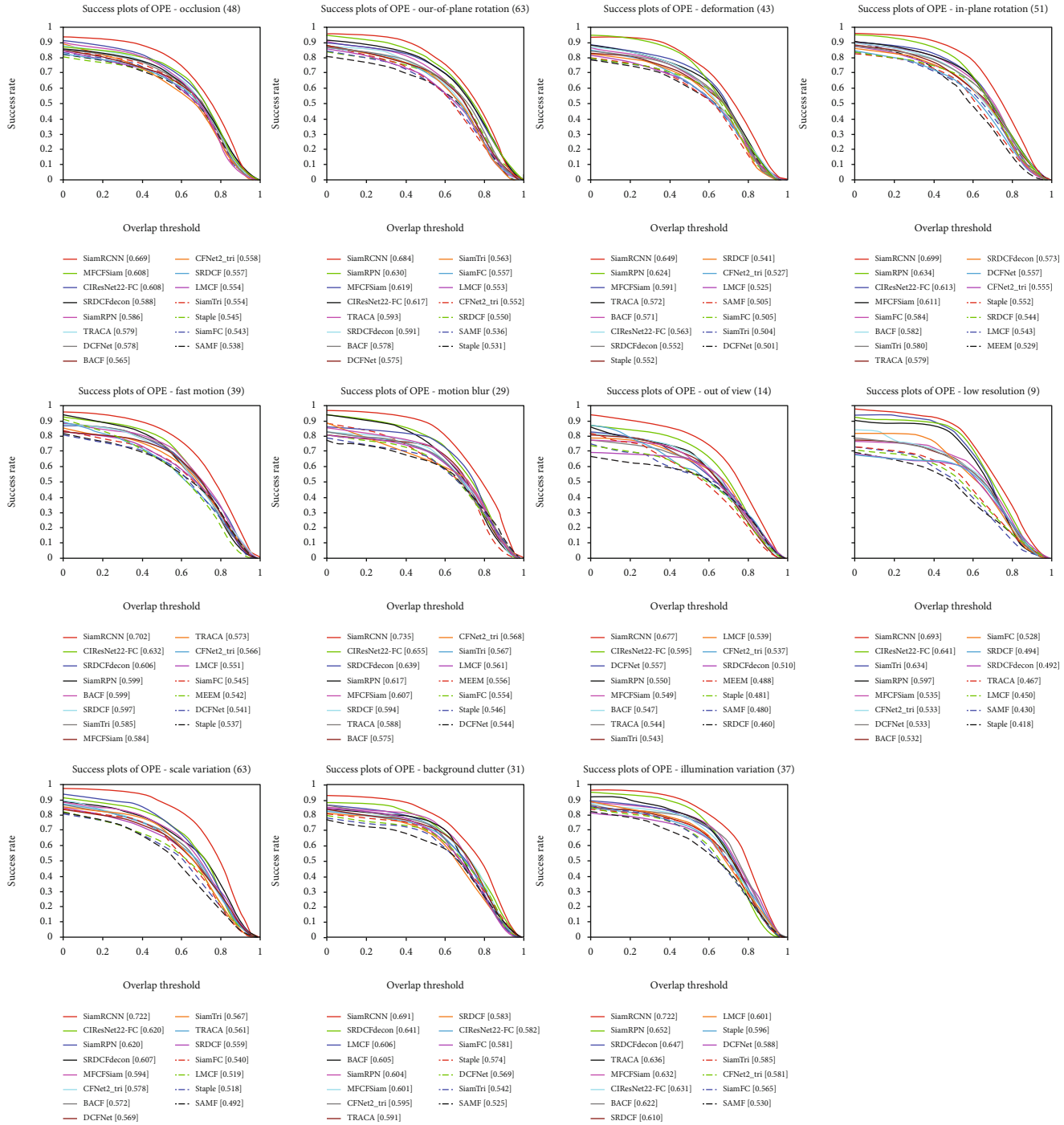


Figure 8: Tracking result of the 15 compared trackers in the success plot of OTB100 in 11 different scenarios. This picture is best viewed on high-resolution displays.

the texture information of the targets and the background are mixed up so that the tracker may drift to the background a little. We think these are the reason that our MFCFSiam tracker's performance is not so good as the SiamFC tracker in the precision plots of BC and LR attributes. Anyway, our tracker has outperformed the baseline trackers in 20 out of the total 22 plots in Figures 5 and 6, This can strongly prove the correctness and effectiveness of our tracking theory. What is more, in the precision plots, the MFCFSiamFC (zero padding) tracker's performance is better than the MFCFSiamFC (context padding) tracker in nine out of the 11 attributes (except FM and MB). In the success plots, the MFCFSiamFC (zero padding) tracker's performance is better than the MFCFSiamFC (context padding) tracker in four out of the 11 attributes (except FM, BC, MB, LR, OCC, OV, and SV). So, in general, the zero padding is more robust than the context padding in the validity evaluation criterion.

Table 2: MFCFSiam's precision ranking in each scenario of all 15 trackers.

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Precision ranking	8	5	5	3	4	4	5	2	3	5	5

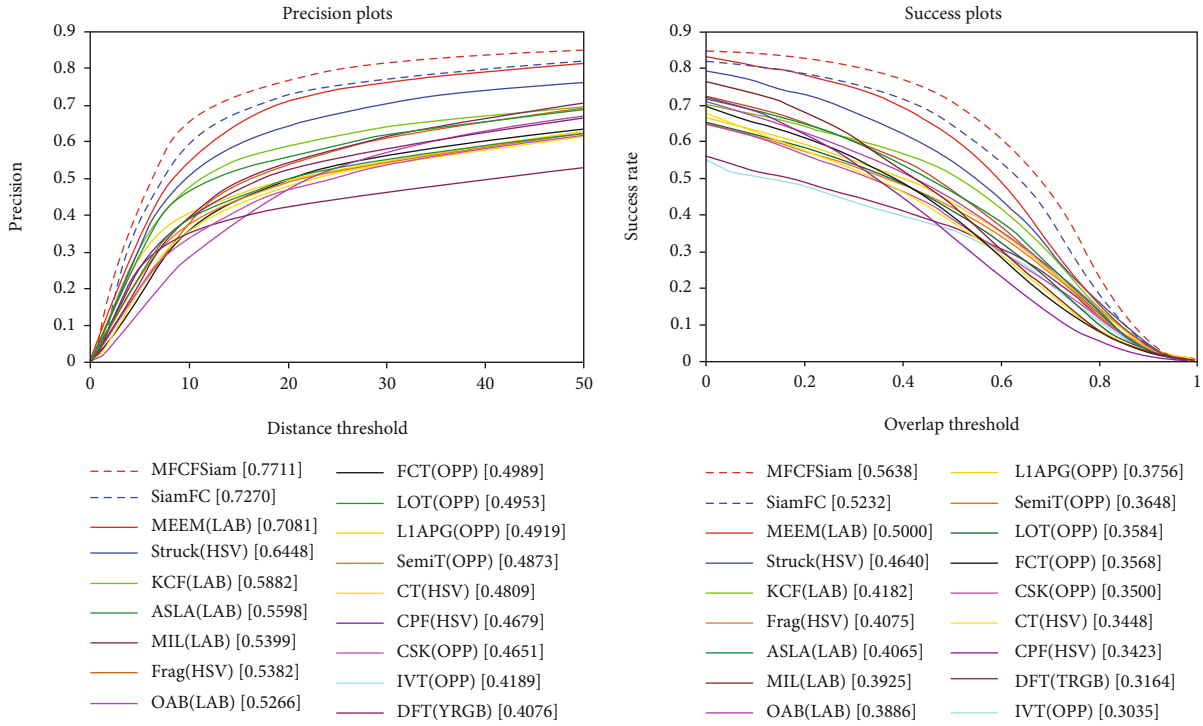


Figure 9: Tracking results of our MFCFSiam tracker and 17 other trackers on the TC128 dataset. This picture is best viewed on high-resolution displays.

4.3. Comparison with State-of-the-Art Trackers on OTB100. To further evaluate the performances of our MFCFSiam tracker, we selected several state-of-the-art trackers that are designed on different tracking strategies. They are all classic and representative in the visual tracking community. Those trackers are as follows: Siam RCNN tracker [58], a fully convolutional Siamese (SiamFC) tracker [38], a Siamese network using triple loss (SiamFC_tri) tracker [62], SiamRPN tracker [54], a context-aware deep feature compression (TRACA) tracker [63], a correlation filter network using triple loss (CFNet2_tri) tracker [62], a cropping-inside residual fully convolutional network (CIResNet-22FC) tracker [64], a staple tracker [65], a spatially regularized discriminative correlation filter (SRDCF) tracker [51], a background-aware correlation filter (BACF) tracker [66], a discriminative correlation filter network (DCFNet) tracker [67], a large-margin correlation filter (LMCF) tracker [68], and scale adaptive kernel correlation filter tracker (SAMF) [69]. The selection of these trackers provided a horizontal comparison to comprehensively evaluate our MFCFSiam tracker. To be concise, we have only listed the results of the precision and success plots on OTB100 because it contains all the sequences that OTB50 and OTB2013 have, so these results were persuasive enough. What is more, the MFCFSiam we adopt here is the one using zero-padding method.

4.3.1. Overall Performances. We compared our MFCFSiam tracker's overall tracking performance on the OTB100 dataset with all the state-of-the-art trackers listed above. Figure 7 shows the results of the precision plot and success plot. Our MFCFSiam tracker achieved the third-best performance (0.839) in the precision plot, inferior only to SiamRCNN (0.894) and SiamRPN (0.860), outperforming all 12 other state-of-the-art trackers. In the success plot, MFCFSiam achieved the second-best performance (0.636), only inferior to the MDNet tracker (0.678) and outperforming all the other 12 state-of-the-art trackers. So, this comparison shows that our MFCFSiam tracker's overall performance on OTB100 is quite competitive and proves the effectiveness of our tracking strategy.

4.3.2. Scenario-Based Performance. We also compared our MFCFSiam tracker's performance with that of the 13 state-of-the-art trackers in 11 different scenarios on OTB100. Figure 8 shows the detailed results of all trackers' performance in the success plot. MFCFSiam's rankings in all 11 scenarios are summarized in Table 2. We can see that our MFCFSiam ranks in the top five in 10 out the 11 scenarios, except for the fast motion (FM) scenario. This proves that our MFCFSiam tracker can handle many complicated tracking environments and shows competitive performance when compared with the state-of-the-art trackers.

Table 3: Precision and area under the curve (AUC) values of the MFCFSiam and eight other trackers on the TC128 dataset (the best and second-best scores are marked with italics and bold, respectively).

	MFCFSiam	MCPF	SRDCF	DeepSRDCF	Staple	BACF	SRDCFdecon	HDT	CNT
Precision	*77.11%*	**76.9%**	69.6%	74.0%	66.8%	66.0%	72.9%	68.6%	44.9%
AUC	*56.38%*	**55.2%**	51.6%	54.1%	50.9%	49.6%	54.3%	48.0%	33.5%

Table 4: Precision and area under the curve (AUC) values of the MFCFSiam and twelve other trackers on the UAV-123 dataset (the best and second-best scores are marked with italics and bold, respectively).

	MFCFSiam	ECO	SRDCF	MEEM	SiamFC	CNT	MUSter	ALSA	DSST	BACF	SAMF	OAB	CFNet
Precision	**71.2%**	*74.1%*	67.6%	62.7%	69.9%	52.4%	72.9%	57.1%	58.6%	65. 4%	59.2%	49.5%	65.1%
AUC	**49.0%**	*52.5%*	46.4%	39.2%	45.7%	36.9%	54.3%	40.7%	35.6%	45.7%	39.6%	33.1%	43.6%

4.4. Comparison with State-of-the-Art Trackers on TC128. Besides from the 100 image sequences of OTB dataset, we also evaluated our MFCFSiam tracker's performance using the TC128 dataset. This dataset has 128 sequences of color images, more than the OTB dataset, and contains some more complicated tracking environments. So, using TC128 helped us to more comprehensively examine MFCFSiam's properties. We also adopted many other classic trackers whose tracking results could be downloaded from the TC128's homepage as comparisons. All those trackers' tracking results of precision plot and success plot are shown in Figure 9. We see that our MFCFSiam tracker (0.7711 and 0.5638) outperformed all the other trackers.

Some other state-of-the-art trackers have also published their average precision values and AUC values for the TC128 dataset. We also summarize these data and compare them with our MFCFSiam tracker. These trackers are as follows: convolutional networks without a training (CNT) tracker [70], a hedged deep tracker (HDT) [50], an adaptive decontamination of spatially regularized discriminative correlation filter (SRDCFdecon) tracker [71], a multitask correlation particle filter (MCPF) [72], a spatially regularized discriminative correlation filter (SRDCF) tracker [51], a background-aware correlation filter (BACF) tracker [66], and convolutional features for correlation filter (DeepSRDCF) tracker [73]. Detailed results are shown in Table 3; the best and second-best performances are marked with italics and bold, respectively. We can see that the MFCFSiam tracker's precision value and AUC value rank first out of the nine trackers.

4.5. Comparison with State-of-the-Art Trackers on UAV-123. UAV-123 [74] is another typical dataset which is widely used for visual tracking. It consists of 123 color image sequences collected by the low-altitude UAV (unmanned aerial vehicle). Compared with the OTB and TC128, UAV-123 contains more long-term image sequences, and the number of frames it contains in total is more than 110 K. The typical characteristic of UAV-123 is that the background in the images is always not so complex, but most sequences contain many changes of view angle, which makes the tracking task quite challenging. So, we believe that the UAV-123 dataset can provide another approach to test the effectiveness of our tracking framework.

Table 5: All the 10 trackers' performance on VOT2018 dataset. The best, second-best, and third-best performances are marked with italics, bold, and bold-italics, respectively.

Tracker	Accuracy ↑	Robustness ↓	EAO ↑
MFCFSiam	**0.589**	0.263	**0.386**
SiamRPN++ [76]	*0.600*	0.234	*0.414*
DeepSTRCF [77]	0.523	0.215	0.345
SiamRPN	***0.586***	0.276	***0.383***
SiamVGG [78]	0.531	0.286	0.348
SA_Siam_R [79]	0.566	0.258	0.337
DSiam [53]	0.512	0.646	0.196
MBSiam	0.529	0.443	0.231
UPDT [80]	0.536	**0.184**	0.378
DRT [81]	0.519	***0.201***	0.356
RCO	0.507	*0.155*	0.376
CPT [82]	0.506	0.239	0.339

The evaluation metric UAV-123 uses are the same as OTB. In this section, we compare our MFCFSiam tracker with 12 classical state-of-the-art trackers, including the efficient convolution operator (ECO) tracker [75], the discriminative correlation filter network (DCFNet) tracker [67], convolutional networks without a training (CNT) tracker [70], and the background-aware correlation filter (BACF) tracker [66]. We summarize all these 13 trackers' performance scores in Table 4 to compare them with our MFCFSiam tracker. The best and second-best scores are marked with italics and bold, respectively. From Table 4, we can find that, among all the 13 compared trackers, our MFCFSiam tracker is only outperformed by the ECO tracker and ranks the second in both plots (71.2% and 49.0%). Compared with the baseline tracker (e.g., SiamFC), our MFCFSiam tracker has made obvious improvements in both the precision plot and the success plot. This can further demonstrate the effectiveness of the tracking framework we proposed.

4.6. Comparison with State-of-the-Art Trackers on VOT2018. VOT (Visual Object Tracking) is another typical dataset designed for visual tracking evaluation. It contains 60 sequences of color images. The property of those images is

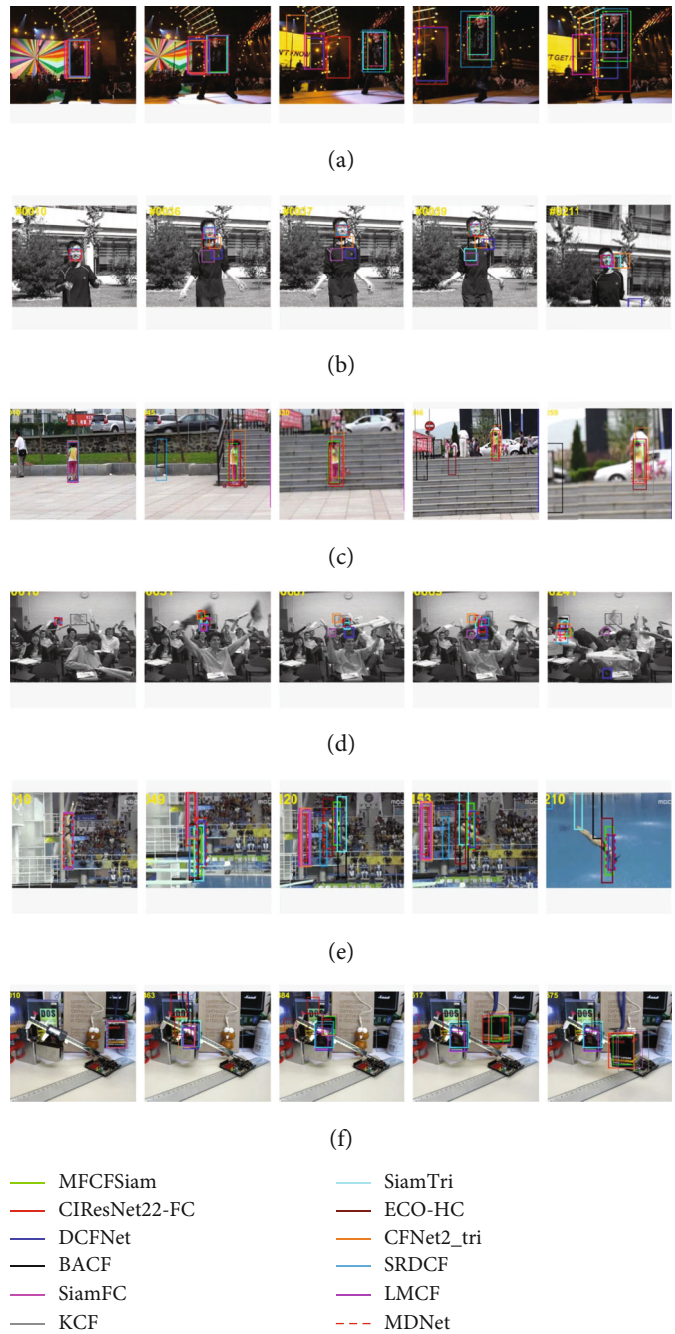


Figure 10: Qualitative tracking results of our MFCFSiam tracker and 11 state-of-the-art trackers on several typical sequences of the OTB. (a–f) The six rows of sequences are (a) singer 2, (b) jumping, (c) girl 2, (d) freeman 4, (e) diving, and (f) box. The color of each tracker is listed at the bottom of the figure.

similar to those of OTB and TC128, but VOT dataset uses an evaluation protocol which is different from the OTB. In the VOT challenge protocol, the tracker will be reinitialized whenever a tracking failure is observed. Three metrics are used to represent the performance of the tracker: accuracy, robustness, and expected average overlap (EAO) score. Accuracy denotes the average overlap between the tracking result and the groundtruth bounding box. Robustness represents how many times tracking failures occur during the tracking process, so a smaller robustness value means a better tracker. EAO represents the average overlap with no reinitialization following a failure. So, using VOT dataset can provide another new approach to show the effectiveness of our tracking framework. In this section, we use the VOT2018 dataset to evaluate the trackers' performance.

We compare our MFCFSiam tracker with eleven other state-of-the-art trackers, and the calculated metrics to represent those trackers' performances are summarized in Table 5. We can see from the table that our MFCFSiam tracker's accuracy (0.589) ranks second, only smaller than the SiamRPN++

(0.600) and outperformed all the other 10 trackers. Our tracker's EAO (0.386) also ranks the second and only inferior to the SiamRPN++ (0.414), outperforming all the other 10 trackers. So, we can see that on the VOT2018 dataset, our MFCFSiam tracker still shows quite competitive performance. This can further demonstrate the effectiveness of the tracking framework we proposed.

4.7. Qualitative Experiments. In this section, we visualize the tracking results of our MFCFSiam tracker and 11 other state-of-the-art trackers on the OTB dataset. Details are shown in Figure 10. The six rows of image sequences are (a–f) singer 2, jumping, girl 2, freeman 4, diving, and box. The color of each tracker's bounding box is listed at the bottom of Figure 10.

The singer 2 sequence is a typical example that contains deformation and background clutter (BC); both the target and the background color are very dark, and the contrast between them is very low, so this makes the trackers tend to drift to the background. Figure 10 shows that our MFCFSiam tracker can constantly capture the right target. The jumping sequence is a typical sequence containing fast motion (FM) and motion blur (MB); the target moves very fast, and his face is blurred when he is jumping, so many trackers drift to the target's body, while our MFCFSiam tracker is among the few trackers that locate the right target. The girl 2 sequence contains many people distractors in the scene, and those distractors usually walk past the right target and block her, so some trackers drift to the background or focus on other distractors because of the occlusion. As shown in Figure 10, the MFCFSiam tracker can always locate the target girl. The freeman 4 sequence is a typical sequence that contains occlusion (OCC); the size of the target is so small that it is very easily blocked. What is more, the occlusion in this sequence is very frequent. Our tracker's performance is good both in precision and scale. The diving sequence is a typical sequence that contains background clutter (BC) and fast motion (FM); the audience in the background can lead the trackers to drift. Before the diver jumped into the river, some trackers had already failed, as shown in the third image; while the diver was entering the river, as shown in the fifth image, only three trackers still held the right target, including the MFCFSiam tracker. The box sequence is another sequence containing typical occlusion (OCC), and the occlusion in this sequence lasts for a few seconds, as shown in the second and third images. When the box appears again later, as shown in the 4th image, many trackers drifted but the MFCFSiam tracker still captured the right target. In conclusion, all six sequences offer evidence of the robustness and effectiveness of our MFCFSiam tracker in challenging tracking scenarios.

5. Conclusions

In this paper, we proposed a novel tracking framework to explore the potential of combining the SiamFC tracker with other CF-based trackers, using the detailed texture features such as the HOG and CN to guide the high-level semantic features in CNN. We also designed an evaluation criterion that uses a correlation filter and the zero-padding method to evaluate the validity of the tracking results. Comparative experiments with other state-of-the-art trackers were conducted on the OTB, TC128, UAV-123, and VOT2018 dataset to verify the effectiveness of our strategy. In the future, our work will mainly focus on keeping optimize the evaluation criterion of tracking results. We believe this can provide a meaningful tool for combining more different trackers.

Data Availability

The data used to support the findings of this study are available from those websites: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html, https://cemse.kaust.edu.sa/ivul/uav123, and https://www.dabi.temple.edu/~hbling/publication/TColor-128.pdf.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

C.L. constructed the tracking framework, performed the experiments, and wrote the original manuscript. Q.X., K.Z, and Z.M. analyzed and interpreted the experiment results and provided suggestions about the experiments and revision of this manuscript.

References

[1] X. Wang, "Intelligent multi-camera video surveillance: a review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.

[2] B. Maurin, O. Masoud, and N. P. Papanikolopoulos, "Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds," *IEEE robotics & automation magazine*, vol. 12, no. 1, pp. 29–36, 2005.

[3] J. Lien, E. M. Olson, P. M. Amihood, and I. Poupyrev, *RF-Based Micro-Motion Tracking for Gesture Tracking and Recognition*, 2019.

[4] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3101–3109, Santiago, Chile, 2015.

[5] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 763–771, 2016.

[6] K. Li, F.-Z. He, and H.-P. Yu, "Robust visual tracking based on convolutional features with illumination and occlusion handing," *Journal of Computer Science and Technology*, vol. 33, no. 1, pp. 223–236, 2018.

[7] H. Alismail, B. Browning, and S. Lucey, "Robust tracking in low light and sudden illumination changes," in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 389–398, Stanford, CA, USA, 2016.

[8] L. Chen, F. Zhou, Y. Shen, X. Tian, H. Ling, and Y. Chen, "Illumination insensitive efficient second-order minimization for planar object tracking," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4429–4436, Singapore, Singapore, 2017.

[9] S. Liu, G. Liu, and H. Zhou, "A robust parallel object tracking method for illumination variations," *Mobile Networks and Applications*, vol. 24, no. 1, pp. 5–17, 2019.

[10] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 3074–3082, Santiago, Chile, 2015.

[11] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[12] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 548–557, Salt Lake City, UT, USA, 2018.

[13] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, Portland, OR, USA, 2013.

[14] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: robust tracking via multiple experts using entropy minimization," in *European conference on computer vision*, pp. 188–203, Springer, 2014.

[15] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang, "Object tracking with multi-view support vector machines," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 265–278, 2015.

[16] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[17] H. Song, Y. Zheng, and K. Zhang, "Robust visual tracking via self-similarity learning," *Electronics Letters*, vol. 53, no. 1, pp. 20–22, 2016.

[18] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1818–1828, 2015.

[19] Y. Xie, W. Zhang, Y. Qu, and Y. Zhang, "Discriminative subspace learning with sparse representation view-based model for robust visual tracking," *Pattern Recognition*, vol. 47, no. 3, pp. 1383–1394, 2014.

[20] Y. Sui, S. Zhang, and L. Zhang, "Robust visual tracking via sparsity-induced subspace learning," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4686–4700, 2015.

[21] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 374–383, 2013.

[22] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 171–190, 2015.

[23] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14277–14301, 2019.

[24] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual tracking via constrained incremental non-negative matrix factorization," *IEEE signal processing letters*, vol. 22, no. 9, pp. 1350–1353, 2015.

[25] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, Columbus, OH, USA, 2014.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[28] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?," in *European conference on computer vision*, pp. 443–457, Springer, 2016.

[29] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 650–657, Washington, DC, USA, 2017.

[30] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2017.

[31] F. Milletari, S.-A. Ahmadi, C. Kroll et al., "Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound," *Computer Vision and Image Understanding*, vol. 164, pp. 92–102, 2017.

[32] Z. Cai and N. Vasconcelos, "Cascade r-cnn: delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, Salt Lake City, UT, USA, 2018.

[33] M. Duan, K. Li, C. Yang, and K. Li, "A hybrid deep learning CNN–ELM for age and gender classification," *Neurocomputing*, vol. 275, pp. 448–461, 2018.

[34] H. Li, Y. Li, and F. Porikli, "Deeptrack: learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2015.

[35] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4293–4302, Las Vegas, NV, USA, 2016.

[36] H. Nam, M. Baek, and B. Han, "Modeling and propagating cnns in a tree structure for visual tracking," 2016, http://arxiv.org/abs/1608.07242.

[37] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," *International conference on machine learning*, pp. 597–606, 2015.

[38] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865, Springer, 2016.

[39] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.

[40] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, 2010.

[41] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*, pp. 702–715, Springer, 2012.

[42] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[43] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5388–5396, Boston, MA, USA, 2015.

[44] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference*, Nottingham, 2014.

[45] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowledge-Based Systems*, no. article 105697, 2020.

[46] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Joint group feature selection and discriminative filter learning for robust visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7950–7960, Seoul, Korea, 2019.

[47] D. Yuan, X. Shu, and Z. He, "TRBACF: learning temporal regularized correlation filters for high performance online visual object tracking," *Journal of Visual Communication and Image Representation*, vol. 72, article 102882, 2020.

[48] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2891–2900, Seoul, Korea, 2019.

[49] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge-Based Systems*, no. article 105526, 2020.

[50] Y. Qi, S. Zhang, L. Qin et al., "Hedged deep tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4303–4311, Las Vegas, NA, USA, 2016.

[51] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 4310–4318, Santiago, Chile, 2015.

[52] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813, Honolulu, HI, USA, 2017.

[53] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1763–1771, Venice Italy, 2017.

[54] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, Salt Lake City, UT, USA, 2018.

[55] A. Lukezic, J. Matas, and M. Kristan, "D3S-A discriminative single shot segmentation tracker," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7133–7142, 2020.

[56] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese attention networks for visual object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6728–6737, 2020.

[57] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7952–7961, Long Beach Canada, 2019.

[58] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam r-cnn: visual tracking by re-detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6578–6588, 2020.

[59] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[60] D. Li, F. Porikli, G. Wen, and Y. Kuai, "When correlation filters meet siamese networks for real-time complementary tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 509–519, 2019.

[61] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[62] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 459–474, Munich, Germany, 2018.

[63] J. Choi, H. Jin Chang, T. Fischer et al., "Context-aware deep feature compression for high-speed visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 479–488, Salt Lake City, UT, USA, 2018.

[64] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4591–4600, Long Beach Canada, 2019.

[65] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1401–1409, Las Vegas, NA, USA, 2016.

[66] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1135–1143, Venice Italy, 2017.

[67] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "Dcfnet: discriminant correlation filters network for visual tracking," 2017, http://arxiv.org/abs/1704.04057.

[68] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4021–4029, 2017.

[69] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *European conference on computer vision*, pp. 254–265, Springer, 2014.

[70] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1779–1792, 2016.

[71] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1430–1438, Las Vegas, NA, USA, 2016.

[72] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4335–4343, Honolulu, HI, USA, 2017.

[73] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 58–66, Santiago, Chile, 2015.

[74] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *European Conference on Computer Vision (ECCV16)*, Amsterdam, The Netherlands, 2016.

[75] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6638–6646, Honolulu, HI, USA, 2017.

[76] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn ++: evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291, Long Beach Canada, 2019.

[77] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4904–4913, Salt Lake City, UT, USA, 2018.

[78] Y. Li and X. Zhang, "SiamVGG: visual tracking using deeper siamese networks," 2019, http://arxiv.org/abs/1902.02804.

[79] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold Siamese network for real-time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4834–4843, Salt Lake City, UT, USA, 2018.

[80] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 483–498, Munich, Germany, 2018.

[81] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 489–497, Salt Lake City, UT, USA, 2018.

[82] M. Che, R. Wang, Y. Lu, Y. Li, H. Zhi, and C. Xiong, "Channel pruning for visual tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6689134, 16 pages
https://doi.org/10.1155/2020/6689134

Research Article

Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient Feature Engineering

Abdelouahid Derhab[1], **Arwa Aldweesh**[2], **Ahmed Z. Emam**[2], **and Farrukh Aslam Khan**[1]

[1]*Center of Excellence in Information Assurance (CoEIA), King Saud University, Saudi Arabia*
[2]*College of Computer and Information Sciences (CCIS), King Saud University, Saudi Arabia*

Correspondence should be addressed to Abdelouahid Derhab; abderhab@ksu.edu.sa

Received 13 October 2020; Revised 23 November 2020; Accepted 4 December 2020; Published 23 December 2020

Academic Editor: Xiaojie Wang

In the era of the Internet of Things (IoT), connected objects produce an enormous amount of data traffic that feed big data analytics, which could be used in discovering unseen patterns and identifying anomalous traffic. In this paper, we identify five key design principles that should be considered when developing a deep learning-based intrusion detection system (IDS) for the IoT. Based on these principles, we design and implement Temporal Convolution Neural Network (TCNN), a deep learning framework for intrusion detection systems in IoT, which combines Convolution Neural Network (CNN) with causal convolution. TCNN is combined with Synthetic Minority Oversampling Technique-Nominal Continuous (SMOTE-NC) to handle unbalanced dataset. It is also combined with efficient feature engineering techniques, which consist of feature space reduction and feature transformation. TCNN is evaluated on Bot-IoT dataset and compared with two common machine learning algorithms, i.e., Logistic Regression (LR) and Random Forest (RF), and two deep learning techniques, i.e., LSTM and CNN. Experimental results show that TCNN achieves a good trade-off between effectiveness and efficiency. It outperforms the state-of-the-art deep learning IDSs that are tested on Bot-IoT dataset and records an accuracy of 99.9986% for multiclass traffic detection, and shows a very close performance to CNN with respect to the training time.

1. Introduction

The Internet of Things (IoT) network is a set of smart devices such as sensors, home appliances, phones, vehicles, and computers that are interconnected through the global Internet. This type of network is increasingly becoming an essential part of our everyday life and is providing a variety of applications such as smart home, smart grid, smart agriculture, smart cities, and intelligent transportation.

Although the IoT can make the human's life more comfortable, this benefit comes at the expense of security [1]. Nowadays, IoT networks are becoming an attractive target for cybercriminals and are exposed to major risks. A report from Unit 42 of Palo Alto Networks revealed that 98% of all IoT device traffic is unencrypted, and 41% of attacks exploit IoT device vulnerabilities [2]. The vulnerable devices could be later used by adversaries to join an IoT botnet and participate in sophisticated and large-scale attacks. For example, the first IoT botnet launched in October 2016, named Mirai [3], was able to compromise vulnerable CCTV cameras that were using default usernames and passwords to launch a DDoS attack on DNS servers. This attack resulted in stopping the Internet accessibility in some parts of the USA. In April 2020, an IoT botnet, named Mozi, was discovered and was found capable of launching various DDoS attacks [4, 5].

To deal with this kind of threat, the intrusion detection systems have been widely used to detect malicious network traffic [6, 7], especially when the preventive techniques fail

at the level of endpoint IoT devices. As cyberattacks targeting IoT are increasingly becoming more sophisticated and stealthy, the IDS should continuously evolve to handle emerging security threats. Due to its heterogeneous nature, IoT network generates high-dimensional, multimodal, and temporal data. By applying big data analytics on such data, it is possible to discover unseen patterns, reveal hidden correlations, and gain new insights [8]. Artificial intelligence is increasingly used in the big data analysis process. In particular, deep learning techniques have proven their success in dealing with heterogeneous data [8–11]. They are also capable of analyzing complex and large-scale data to get insights, spot dependencies within data, and learn from previous attack patterns to recognize new and unseen attack patterns [12–14]. As IoT devices are resource-constrained and have limited capabilities in terms of storage and computation, heavyweight tasks like big data analysis process and building of learning models need to be offloaded to fog and cloud servers [15–21]. Hence, computation offloading [22] can help reduce the execution delay of the task and save energy consumption of battery-powered and mobile IoT devices, but it also poses some security concerns [23].

Many deep learning approaches have been proposed for IDS, and some of them specifically focus on IoT [24–32]. Each approach adopts its own design choices, which might limit its capability in achieving good performance in terms of effectiveness and efficiency.

In this paper, we propose five design principles to be considered when developing an effective and efficient deep learning IDS for IoT, and we use these principles to propose TCNN, a variant of CNN that uses causal convolutions. TCNN is combined with data balancing and efficient feature engineering. More specifically, the main contributions of the paper are the following:

(i) We identify five key design principles for the development of deep learning-based IDS for IoT, including *handling overfitting*, *balancing dataset*, *feature engineering*, *model optimization*, and *testing on IoT dataset*

(ii) Based on the identified key design principles, we compare the state-of-the art methods, identify their gaps, and analyze the main differences with respect to our work.

(iii) We design and implement Temporal Convolution Neural Network (TCNN), a deep learning framework for intrusion detection systems in IoT. TCNN combines Convolution Neural Network (CNN) with causal convolution.

(iv) To handle the issue of imbalanced dataset, we integrate TCNN with Synthetic Minority Oversampling Technique-Nominal Continuous (SMOTE-NC).

(v) We employ efficient feature engineering, which consists of the following:

(1) *Feature space reduction*: it helps in reducing memory consumption.

(2) *Feature transformation*: it is applied on continuous numerical features using *log transformation* and *standard scaler*, which transforms skewed data to Gaussian-like distribution. It is also applied on categorical features using *label-encoding*, which replaces a categorical column with a unique integer value.

(vi) We evaluate the effectiveness and efficiency of the proposed TCNN on Bot-IoT dataset, and compare it with CNN, LSTM, logistic regression, random forest, and other state-of-the-art methods. The results show the superiority of TCNN in scoring an accuracy of 99.9986% for multiclass traffic detection.

The rest of the paper is organized as follows. Section 3 presents the key design principles with respect to deep learning IDS for IoTs. Section 4 overviews related work. Section 4 and Section 5 describe the design and implementation of TCNN, respectively. Section 6 presents the evaluation results and comparison with state-of-the-art methods. Finally, Section 7 concludes the paper and outlines future research directions.

2. Key Design Principles for Deep Learning IDS in IoT

The objective of deep learning-based IDS solutions for IoT is to generate models that perform well in terms of effectiveness and efficiency. However, each model adopts some design choices that might limit its ability in achieving this objective. For example, some deep learning IDSs in IoT do not consider the overfitting problem, or apply their model on an unbalanced dataset, or neglect employing feature engineering, which negatively affects their performance in terms of accuracy, memory consumption, and computational time. Also, some IDSs do not try to optimize their learning model, and some are evaluated on outdated or irrelevant datasets, which do not reflect real-world IoT network traffic.

Motivated by the above observations, the deep learning-based IDS solution for IoT should advocate the following key design principles:

(i) *Handling overfitting*: overfitting happens when the model achieves a good fit on the training data, but it does not generalize well on unseen data. In deep learning, overfitting could be avoided by the following methods:

(1) Applying regularization, which adds a cost to the loss function of the model for large weights.

(2) Using dropout layers, which randomly remove certain features by setting them to 0.

(ii) *Balancing dataset*: data imbalance refers to a disproportion distribution of classes within a dataset. If a model is trained under an imbalanced dataset, it will become biased, i.e., it will favor the majority classes

and fail to detect the minority classes. By balancing the dataset, the effectiveness of the model will be improved.

(iii) *Feature engineering*: it allows reducing the cost of the deep learning workflow in terms of memory consumption and time. It also allows improving the accuracy of the model by discarding irrelevant features and applying feature transformation to improve the accuracy of the learning model.

(iv) *Model optimization*: the objective of model optimization is to minimize a loss function, which computes the difference between the predicted output and the actual output. This is achieved by iteratively adjusting the weights of the model. By applying an optimization algorithm such as SGD and Adam [33], the effectiveness of the model will be improved.

(v) *Testing on IoT dataset*: a deep learning-based IDS for IoT should be tested under an IoT dataset to get results that reflect real-world IoT traffic.

3. Related Work

Deep learning has been applied in many fields of cybersecurity including malware detection [34–39] and intrusion detection system [14, 40–46]. In this section, we give an overview on deep learning-based IDS for IoT networks.

Lopez et al. [26] proposed RNN-CNN, a combination of recurrent neural network (RNN) and convolutional neural network (CNN). To deal with overfitting, they added some layers such as max pooling, batch normalization, and dropout. They only considered a subset of features to improve the effectiveness of the model.

Putchala [28] applied the Gated Recurrent Unit (GRU) algorithm on KDD 99Cup dataset. He also used the random forest classifier as a feature selection technique. The best possible performance results are obtained by minimizing the loss function.

Roy and Cheung [31] presented Bidirectional Long Short-Term Memory Recurrent Neural Network (BLSTM RNN). They applied feature normalization and converted categorical features to numeric values.

Diro et al. [24] applied a deep neural network (DNN) on NSL-KDD dataset. The loss function of DNN is minimized using stochastic gradient descent (SGD). There are fog nodes, which are responsible for training the deep learning model. The local parameters are sent to a fog coordinator node for update. This allows sharing the best parameters and helps avoiding local overfitting.

Roopak et al. [29] applied four different classification deep learning models: MLP, 1d-CNN, LSTM, and CNN +LSTM on CICIDS2017 dataset. They also balanced the dataset by duplicating the records. However, it is not explained which balancing method is used. The overfitting issue is handled by adding some layers to the model such as max pooling and dropout.

In [32], Deep Belief Network (DBN) is used to develop a feed-forward deep neural network (DNN) and is applied on an IoT simulation dataset. DNN is optimized by assigning a cost function to each layer of the model.

Otoum et al. [27] proposed Stacked-Deep Polynomial Network (SDPN) on NSL-KDD dataset. For optimal selection of features, they employed the Spider Monkey Optimization (SMO) algorithm [47]. To avoid overfitting, the L2 regularization technique is integrated with the loss function.

Ferrag and Maglaras [25] applied recurrent neural network (RNN) with the truncated backpropagation through time (BPTT) algorithm on two non-IoT datasets and BoT-IoT dataset. They normalized the features before feeding them to RNN-BPTT.

Roopak et al. [30] proposed a sequential architecture combining CNN and LSTM and applied it on CISIDS2017 dataset. For optimal selection of features, they employed a multiobjective optimization algorithm named nondominated sorting genetic algorithm (NSGA) [48]. To avoid overfitting, they implemented a max-pooling layer between CNN and LSTM layers.

Koroniotis et al. [49] are the first who developed BoT-IoT dataset, and they used it to test RNN and LSTM. For feature selection, they computed the correlation coefficient among the features of the dataset and applied feature normalization to scale the data within the range [0, 1].

Aldhaheri et al. [50] proposed DeepDCA, an IDS that combines Dendritic Cell Algorithm (DCA) and Self Normalizing Neural Network (SNN). They adopted Information Gain as a feature selection technique to decide on the set of features to be fed to BoT-IoT dataset. Although, the authors presented results with balanced dataset but no information about balancing method is provided. As for model optimization, they used a loss function to update the weights of the deep learning layers.

Soe et al. [51] proposed Artificial Neural Network (ANN) to detect DDoS attacks in Bot-IoT dataset. To balance the dataset, they used the SMOTE technique. Also, they applied feature normalization before feeding the input data to ANN.

Ge et al. [52] applied feed-forward neural networks (FNN) on BoT-IoT dataset. The dataset is balanced not through oversampling but in an algorithmic way, i.e., giving class weights to the training data. To optimize the model, they used Adam optimizer and a sparse categorical cross-entropy loss function to update weights. To deal with overfitting, they employed different regularization techniques such as L1, L2, and dropout. They also encoded categorical features as numerical using one-hot encoding.

Muna et al. [53] proposed a combination of deep autoencoder (DAE) and deep feed-forward neural network (DFFNN) to detect malicious activities in industrial IoT. The optimal parameters are obtained by calculating a loss function, which allows updating the weights and minimizes the difference between the actual and the predicted output.

3.1. Key Finding. Table 1 summarizes and compares the IDS solutions with respect to the abovementioned five design principles. We can notice that only 6 out of 14 solutions are tested under IoT dataset [25, 27, 49–51]. The majority of solutions do not consider dataset balancing. Only 4 solutions are designed with data balancing [29, 50–52], two of them do

Table 1: Deep learning-based IDS for IoT.

Ref	DL technique	Overfitting	Unbalanced dataset	Feature engineering	Model optimization	Testing on IoT dataset
[26]	CNN-RNN	Yes	No	FS	No	No: RedIRIS
[28]	GRU	No	No	FS:RF	Yes	No: KDDCup'99
[31]	BLSTM RNN	No	No	FE	No	No: UNSW-NB15
[24]	DNN	Yes	No	FE	Yes	No: NSL-KDD
[29]	MLP, 1d-CNN, LSTM, CNN+LSTM	Yes	Yes	No	No	No: CICIDS2017
[32]	DNN	No	No	No	Yes	Yes: simulation
[27]	SDPN	Yes	No	FS:SMO	Yes	No: NSL-KDD
[25]	RNN-BPTT	No	No	FN	No	Yes: Bot-IoT
[30]	CNN+LSTM	Yes	No	FS: NSGA	No	No: CISIDS2017
[49]	RNN, LSTM	No	No	FS:CC	No	Yes: Bot-IoT
[3]	DeepDCA	No	Yes	FS:IG	Yes	Yes: BoT-IoT
[51]	ANN	No	SMOTE	FN	No	Yes: BoT-IoT
[52]	FNN	Yes	Yes	FE	Yes	Yes: BoT-IoT
[53]	DAE-DFFFN	No	No	FE, FN	Yes	No: NSL-KDD, UNSW-NB15
Our work	TCNN	Yes	SMOTE-NC	FSR FT: LT, SS, FE	Yes	Yes: BoT-IoT

FS: feature selection; RF: random forest; FE: feature encoding; FN: feature normalization; FSR: feature space reduction; IG: information gain; CC: correlation coefficient; FT: feature transformation; LT: log transformation; SS: standard scaler; SMO: Spider Monkey Optimization; NSGA: nondominated sorting genetic algorithm.

not explain how the balancing approach is implemented [29, 50], one solution considers algorithmic-level data balancing [52], and only one solution considers data-level balancing by applying SMOTE algorithm [51]. Handling overfitting is not considered in the design of 7 solutions [25, 28, 31, 32, 49, 50, 53]. On the other hand, model optimization is only considered by 7 solutions [24, 27, 28, 32, 50, 52, 53]. Most of the solutions employ feature engineering in their design, except for two solutions [29, 32].

3.2. Comparison with Related Work. To the best of our knowledge, our wok and [52] are the only ones that consider all the five design principles. Differently from [52], which adopts algorithmic-level data balancing, our work applies the SMOTE-NC algorithm on Bot-IoT dataset, which can handle continuous and categorical features. We use overfitting and optimization techniques in achieving effective IDS. We also use feature space reduction and feature transformation in achieving efficient and lightweight IDS in terms of memory usage and training time.

4. Proposed Framework

4.1. Basic Principles. Deep learning is a concatenation of different layers. The first layer is called the input layer, and the last layer is called the output layer. In addition, hidden layers are inserted between the input and the output layers. Each layer is composed of a set of units, called neurons. The size of the input layer depends on the dimension of the input data, whereas the output layer is composed of C units, which corresponds to the C classes of a classification task.

Convolutional neural network (CNN), as shown in Figure 1, is a deep neural network that is composed of multiple layers. The three main types of layers are the following:

(i) *Convolutional layer*: it applies a set of filters, also known as convolutional kernels, on the input data. Each filter slides over the input data to produce a feature map. By stacking all the produced feature maps together, we get the final output of the convolution layer

(ii) *Pooling layer*: it operates over the feature maps to perform subsampling, which reduces the dimensionality of the feature maps. Average pooling and max pooling are the most common pooling methods

(iii) *Fully connected layer*: It takes the output of the previous layers, and turns them into a single vector that can be an input for the next layer

The TCNN deep learning architecture [54] is a combination of CNN architecture and causal padding, which results in causal convolutions. Figure 2 shows 1D causal convolution with a kernel size of 3, which is applied on time-series input data $(x_0, x_1, \cdots, x_T)$. By causal convolutions, we mean that an output at time t is convolved only with elements from time t and earlier in the previous layer. Therefore, it does not violate the temporal order of the data, and there is no leakage of information from future to past. Zero padding of length $(\text{kernel size} - 1)$ is added to the layers to have the same length as the input causal convolution layer.

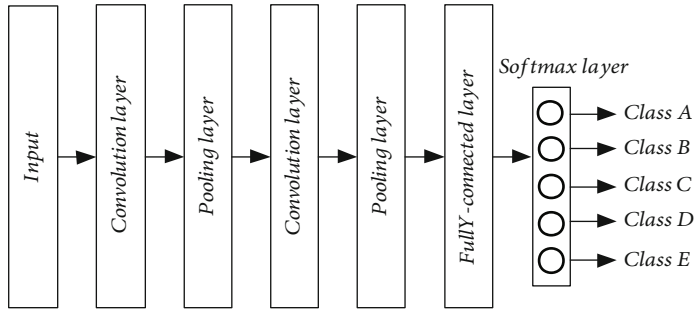


Figure 1: CNN architecture.

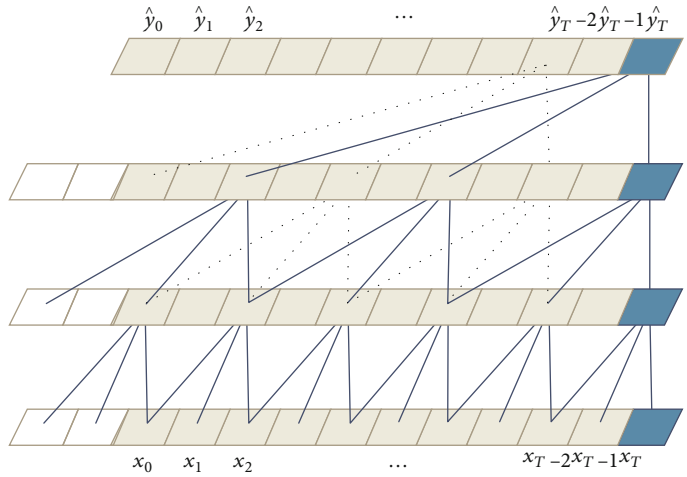


Figure 2: 1D causal convolution [54].

4.2. Overall Architecture. Figure 3 shows the overall architecture of the proposed TCNN framework, and its implementation is detailed in Section 5. The proposed architecture is composed of the following phases:

(i) *Dataset balancing*: as mentioned above, an imbalanced dataset can produce misleading results. To handle this problem, we use in this phase the SMOTE-NC method, which creates synthetic samples of minority classes and is capable of handling mixed dataset of categorical and continuous features.

(ii) *First feature engineering (feature space reduction)*: in this phase, we clean the dataset, i.e., reduce the feature space by removing unnecessary features, and converting the memory-consumption features into lower-size datatype.

(iii) *Dataset splitting*: in this phase, the dataset is split into: training, validation, and testing subsets in order to counter overfitting.

(iv) *Second feature engineering (feature transformation)*: in this phase, we apply feature transformation on the training subset. Log transformation and standard scaler are applied on the continuous numerical features. In addition, label encoding is applied to categorical features, which simply replaces each categorical column with a specific number. This transformation process is later applied on the validation and the testing subsets.

(v) *Training and optimization*: in this phase, the TCNN model is built, as described in Section 4.3. It is trained using the training subset, and its parameters are optimized using Adam optimizer and the validation subset.

(vi) *Classification:* the generated TCNN model is applied on the testing subset to attribute each testing record to its actual class: normal or a specific category of attack.

4.3. Training and Optimization of TCNN Framework. The training and optimization phase of the proposed TCNN is composed of two 1D causal convolution layers, two dense layers, and a softmax layer, which applies softmax functions for multiclass classification task. To overcome overfitting, we use global maximum pooling, batch normalization, and dropout layers. We choose Adam optimizer to update weights and optimize cross-entropy loss function. Adam optimizer combines the advantages of two stochastic gradient descent algorithms, namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

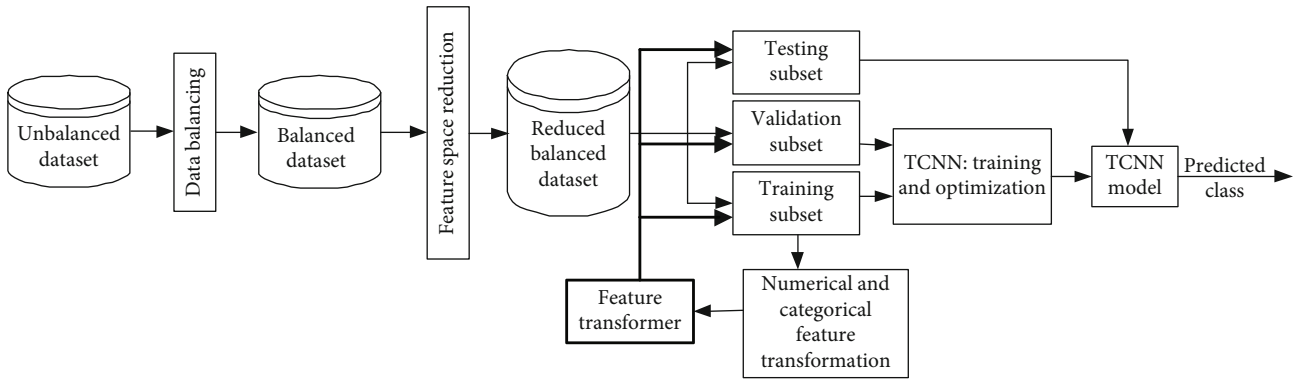


Figure 3: TCNN framework.

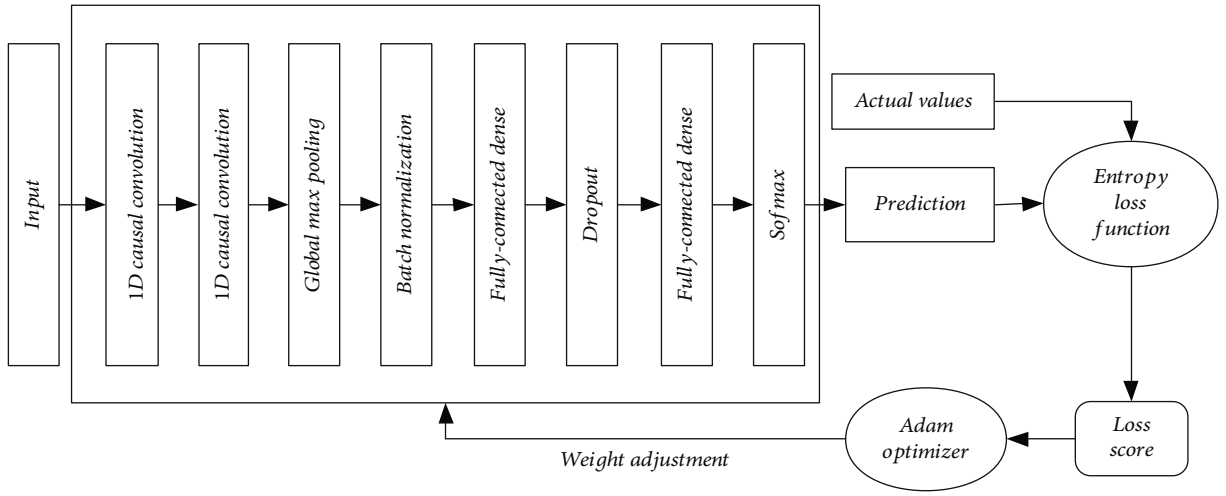


Figure 4: Training and optimization of the proposed TCNN framework.

Specifically, the training and optimization phase of the proposed TCNN architecture, as shown in Figure 4, is composed of the following layers:

(i) *First 1D causal convolution layer*: it convolves across the input vectors with 64 filters and filter size of 3.

(ii) *Second 1D causal convolution layer*: it uses 128 filters and a filter size of 3. This second layer before pooling allows the model to learn more complex features.

(iii) *1D global maximum pooling layer*: it replaces data, which is covered by the filter, with its maximum value. It prevents overfitting of the learned features by taking the maximum value.

(iv) *Batch normalization layer*: it normalizes the data coming from the previous layer before going to the next layer.

(v) *Fully connected dense layer*: it employs 128 hidden units and a dropout ratio of 30%.

(vi) *Fully connected dense layer with softmax activation function*: it produces five units that correspond to the five categories of traffic for multiclass classification.

5. Implementation

To implement the detection learning models, we use Intel Quad-core i7-8550U processor with 8 GB RAM and 256 GB Hard drive. As for software, we use Python 3.6 programming language, and TensorFlow to build deep learning models. Moreover, different libraries are used including Scikit-learn, Keras API, Panda, and Inmblearn. We implement the framework in Figure 3 on Bot-IoT Dataset [49].

5.1. Bot-IoT Dataset. We use Bot-IoT [49], an IoT dataset that was released in 2018 by the Cyber Center in the University of New South Wales. By virtualizing the setup of various smart home appliances including weather stations, smart fridges, motion-activated lights, remotely activated garage doors, and smart thermostats, legitimate and malicious traffic is generated. The dataset consists of more than 73,000,000 records, which are represented by 42 features, as shown in Table 2. Each record is labeled either as normal or attack. In addition, the attack dataset is divided into four categories: DoS, DDoS, reconnaissance, and information theft, and each category is further divided into subcategories, as shown in Table 3.

Table 2: Features of Bot-IoT dataset.

Feature	Description	Data type	Feature	Description	Data type
pkSeqID	Row identifier	Integer	Dpkts	Destination-to-source packet count	Integer
stime	Record start time	Float	Sbytes	Source-to-destination byte count	Integer
flgs	Flow state flags seen in transactions	Category	Dbytes	Destination-to-source byte count	Integer
proto	Textual representation of transaction protocols presents in network flow	Category	Rate	Total packets per second in transaction	Float
Saddr	Source IP address	Category	Srate	Source-to-destination packets per second	Float
Sport	Source port number	Category	Drate	Destination-to-source packets per second	Float
Daddr	Destination IP address	Category	TnBPSrcIP	Total number of bytes per source IP	Integer
Dport	Destination port number	Category	TnBPDstIP	Total number of bytes per destination IP.	Integer
Pkts	Total count of packets in transaction	Integer	TnP_PSrcIP	Total number of packets per source IP.	Integer
Bytes	Total number of bytes in transaction	Integer	TnP_PDstIP	Total number of packets per destination IP.	Integer
State	Transaction state	Category	TnP_PerProto	Total number of packets per protocol.	Integer
Ltime	Record last time	Float	TnP_Per_Dport	Total number of packets per dport	Integer
Seq	Argus sequence number	Integer	AR_P_Proto_P_SrcIP	Average rate per protocol per source IP. (calculated by pkts/dur)	Float
Dur	Record total duration	Float	AR_P_Proto_P_DstIP	Average rate per protocol per destination IP.	Float
Mean	Average duration of aggregated records	Float	N_IN_Conn_P_SrcIP	Number of inbound connections per source IP.	Integer
Stddev	Standard deviation of aggregated records	Float	N_IN_Conn_P_DstIP	Number of inbound connections per destination IP.	Integer
Sum	Total duration of aggregated records	Float	AR_P_Proto_P_Sport	Average rate per protocol per sport	Float
Min	Minimum duration of aggregated records	Float	AR_P_Proto_P_Dport	Average rate per protocol per dport	Float
Max	Maximum duration of aggregated records	Float	Pkts_P_State_P_Protocol_P_SrcIP	Number of packets grouped by state of flows and protocols per source IP.	Integer
Spkts	Source-to-destination packet count	Integer	Pkts_P_State_P_Protocol_P_DestIP	Number of packets grouped by state of flows and protocols per destination IP	Integer

Table 3: Bot-IoT dataset statistics.

Normal/attack	Category	Subcategory	Number of records
Attack	Reconnaissance (2.48%)	Service scanning	1,463,364
		OS fingerprinting	358,275
	DoS (52.25%)	TCP	19,547,603
		UDP	18,965,106
		HTTP	19,771
	DDoS (44.98%)	TCP	12,315,997
		UDP	20,659,491
		HTTP	29,706
	Information theft (0.22%)	Keylogging	1,469
		Data exfiltration	118
Normal (0.13%)			9,543
	Total		73,370,443

In this work, we use a subset of Bot-IoT dataset, consisting of approximately 3,700,000 records, which is the same as the one used in [49].

5.2. Dataset Balancing. In the dataset, there are 9,543 normal and 73,360,900 attack samples. The subset of the dataset is composed of 477 normal samples and 3,668,045 attack samples. We can notice that more than 97% of the samples belong to DoS and DDoS categories, as shown in Table 3. In this way, the learning model will predict the majority classes and fail to spot the minority classes, which means the model is biased.

To deal with this problem, different resampling methods have been proposed [55] like (1) random oversampling, which randomly replicates the exact samples of the minority classes, and (2) oversampling by creating synthetic samples of minority classes using techniques such as synthetic minority oversampling technique (SMOTE), synthetic minority oversampling technique for nominal and continuous (SMOTE-NC), and adaptive synthetic (ADASYN). In this work, we use the SMOTE-NC technique as it is capable of handling mixed dataset of categorical and continuous features [56]. The minority classes such as normal and theft are increased to 100,0000 samples in the training subset, as shown in Table 4.

5.3. Feature Space Reduction. One of the main objectives of this work is to develop a lightweight IDS for IoT environment. Therefore, it is important to improve the efficiency of the detection models by reducing the feature space and noise in the dataset, as well as reducing the memory usage and computation complexity. By using the full set of features, 2.9 GB of memory is used. Feature space reduction decreases the processing complexity and speeds up the training and detection processes. The following steps are applied to the dataset, which successfully decrease the memory consumption to 668 MB, i.e., 77% reduction.

Table 4: Training dataset: left: original dataset, and right: oversampling dataset.

DDoS	1541299	DDoS	1541299
DoS	1320208	DoS	1320208
Reconnaissance	72865	Normal	100000
Normal	382	Theft	100000
Theft	63	Reconnaissance	72865

Table 5: Data type of features.

Data type	Int64	Float64	Object
# features	22	15	9

(i) *Conversion of object data type into categorical data type*: Table 5 shows the data types and the number of features encoded for each type. As shown in the table, there are 9 memory-consuming features that are encoded as objects, which are "flgs," "proto," "saddr," "sport," "daddr," "dport," "state," "category," and "subcategory." As category datatype is more efficient, object features are converted into category datatype [57].

(ii) *Conversion of Int64 data type into Int32 data type*: by default, the 22 integer features in the dataset, as shown in Table 2, are stored as Int64 (8-bytes) type. After checking these features, we find out that they do not exceed the capacity of Int32 (4-bytes) type. Therefore, all the values of Int64 type are encoded into Int32 type, which incurs half of the memory consumption that is incurred by the Int64 type.

(iii) *Removing unnecessary features*: in the dataset, we exclude some useless features such as the following:

(1) "pkSeqID": it has the same role as the automatically generated index.

(2) "stime" and "ltime": they are captured in the "*dur*" feature, which computes the duration between "*stime*" and "*ltime*".

5.4. Feature Transformation. We describe how the numerical features and categorical features are transformed. After the dataset is split into training, validation, and testing subsets, the transformation is only applied on the training subset. Then, the same transformation is reapplied on the validation and the testing subsets.

(1) *Numerical feature transformation*: the dataset contains 31 numerical features, including both discrete and continuous values. There are two discrete features, i.e., "*spkts*" and "*dpkts*," and are represented by a finite number of values. So, they do not require any feature engineering.

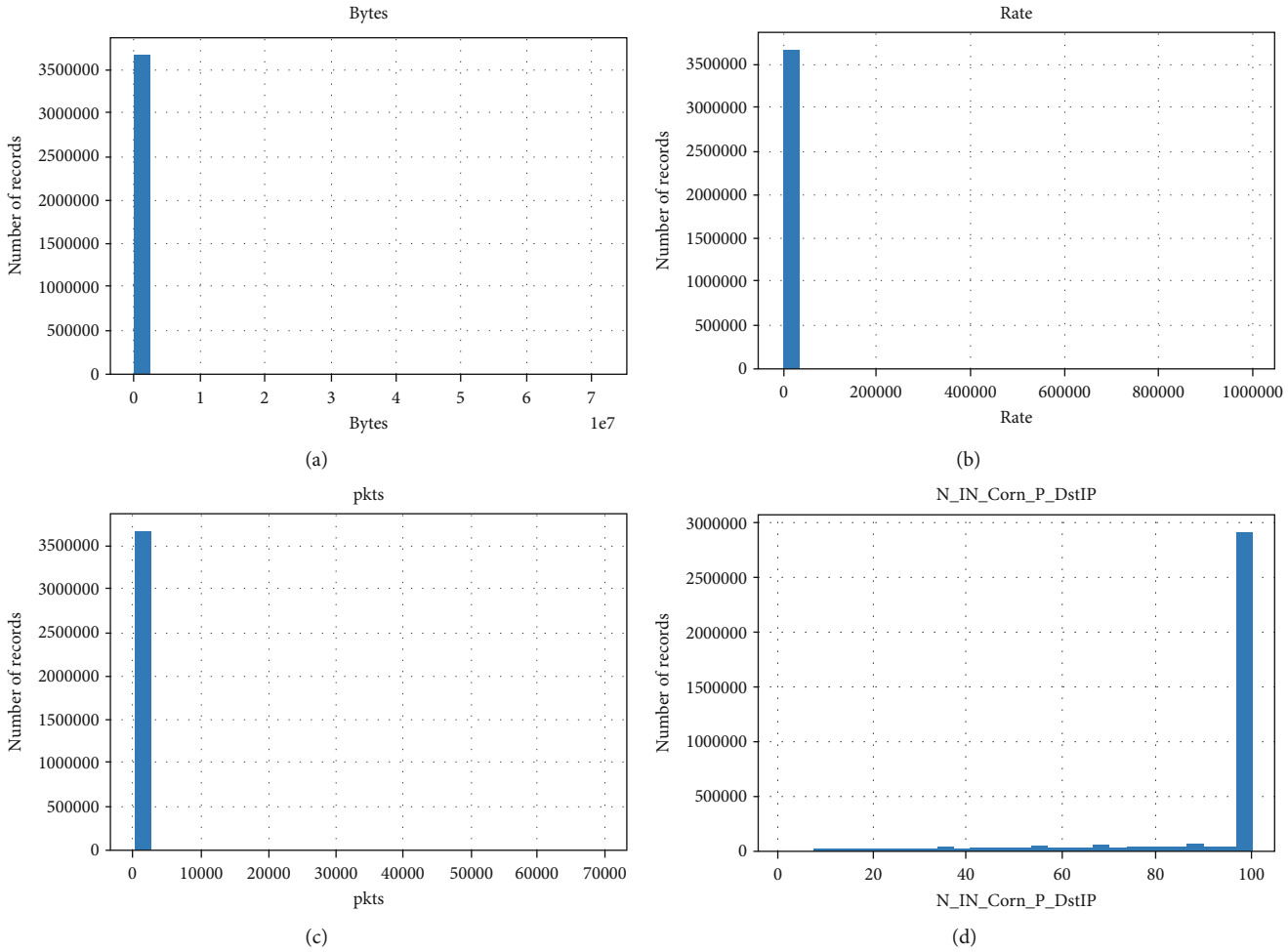


Figure 5: Histogram of some continuous feature before transformation.

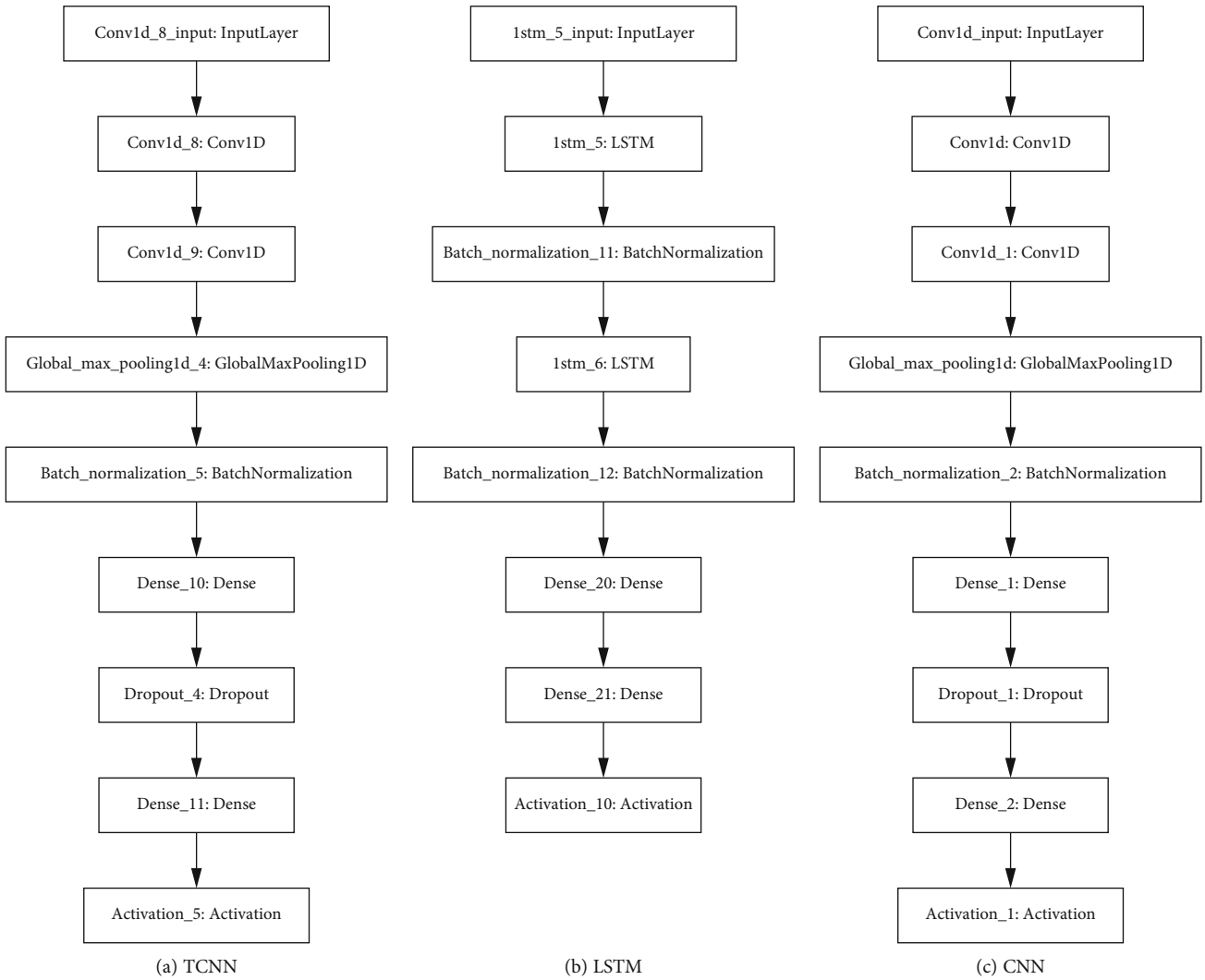
There are 29 continuous features in the dataset, which are "pkts," "bytes," seq, dur,, mean, stddev, sum, min, max, spkts, dpkts, sbytes, dbytes, rate, srate, drate, TnBPSrcIP, TnBPDstIP, TnP_PSrcIP, TnP_PDstIP, TnP_PerProto, TnP_Per_Dport, AR_P_Proto_P_SrcIP, AR_P_Proto_P_DstIP, N_IN_Conn_P_DstIP, N_IN_Conn_P_SrcIP, AR_P_Proto_P_Sport, AR_P_Proto_P_Dport, Pkts_P_State_P_Protocol_P_DestIP, and Pkts_P_State_P_Protocol_P_SrcIP. Figure 5 shows the histograms of 4 features. As shown in the figure, the continuous features are not normally distributed, which usually affects the performance of linear models. To this end, log transformation and standard scaler are applied to the continuous features to be Gaussian-like distribution as follows:

(i) *Log transformation*: the new value of the feature $x' = \log_{10} x$, where x is the original value of the feature.

(ii) *Standard scaler*: it computes the mean μ and standard deviation σ on a training set. Then, the features are normalized to Gaussian distribution. For each x', we compute the normalized value $x'' = x' - \mu/\sigma$

5.5. Dataset Splitting. Conventional splitting and cross-validation are the main approaches used to split datasets. Cross-validation is mainly used in legacy machine learning to overcome the overfitting problem. When a large dataset is used with deep learning, cross-validation increases the training cost. In this work, the dataset is split using the conventional three-way split into: training, validation, and testing subsets. In addition, regularization is applied to deal with the overfitting if it appears [58]. Also, a stratified split is used to ensure that there is a portion of each class in each split [59].

5.6. Deep Learning Models. All deep learning models are built using Keras API on top of TensorFlow. Different Keras packages are used, including preprocessing, models, layers, optimizers, and callback. The same activation functions are used in all models. To model nonlinear relationships between input and output in each layer, relu activation function is used. The output layer activation function is softmax; a generalized logistic regression activation function is used. The number of output units in softmax is equivalent to the number of attack categories in addition to the normal class [60]. The deep learning architectures of TCNN, LSTM, and CNN are shown in Figure 6, and their hyperparameters are shown in Table 6.



(a) TCNN (b) LSTM (c) CNN

Figure 6: Deep learning models.

To deal with overfitting, some techniques such as Global maximum pooling, Batch normalization, and dropout are used. To adjust the weights, Adam optimizer is selected since it outperforms the other optimizers, such as SGD and AdaGrad.

6. Evaluation

We evaluate the performance of TCNN and compare it with two legacy machine learning algorithms, i.e., logistic regression (LR) and random forest (RF), and two deep learning models, i.e., LSTM, and CNN.

6.1. Performance Metrics. The multiclass detection models are evaluated with respect to the following metrics:

(i) *Effectiveness metrics*: we measure how the detection model is effective in distinguishing between the different classes of network traffic. To this end, we use the following metrics:

(i) Accuracy = TP + TN/TP + TN + FP + FN

(ii) Precision = TP/TP + FP

(iii) Recall = TP/TP + FN

(iv) $F1 - \text{score} = 2(\text{Precision} \times \text{Recall})/\text{Precision} + \text{Recall}$

where TP, TN, FP, and FN denote the true positives, true negatives, false positives, and false negatives, respectively.

(ii) *Log loss (cross-entropy loss)*: it measures the performance of the classification model whose output is a probability value. A perfect model would have a log loss of 0, and it increases as the predicted probability diverges from the actual label. Formally,

$$\text{logloss} = -\frac{1}{C}\sum_{i=1}^{C}(y_i \log p_i + (1 - y_i) \log (1 - p_i)) \quad (1)$$

where C is the number of classes.

Table 6: Hyperparameters of deep learning models.

Hyperparameters		Value	Activation function
TCNN	First 1D causal convolution layer	#filters = 64, filter size = 3	ReLU
	Second 1D causal convolution layer	#filters = 128, filter size = 3	ReLU
	Fully connected dense layer	#neurons = 128, dropout = 0. 3	ReLU
	Fully connected dense layer	#neurons = 5	Softmax
CNN	First 1D convolution layer	#filters = 64, filter size = 3	ReLU
	Second 1D convolution layer	#filters = 128, filter size = 3	ReLU
	Fully connected dense layer	#neurons = 128, dropout = 0. 3	ReLU
	Fully connected dense layer	#neurons = 5	Softmax
LSTM	First LSTM layer	#neurons = 20, recurrent dropout = 0.2	ReLU
	Second LSTM layer	#neurons = 20, recurrent dropout = 0.2	ReLU
	Fully connected dense layer	#neurons = 128	ReLU
	Fully connected dense layer	#neurons = 5	Softmax
Optimizer		Adam with learning rate = 0.001	/
Batch size		1024	/
Epochs		15	/

Table 7: Performance of machine learning models.

Detection model	Oversampling	Phase	Log loss	Accuracy	Precision	Recall	F1-score	Training time (s)
LR	None	Training	0.055841	97.0861%	59.8890%	81.2382%	52.1265%	511
		Testing	0.057109	97.0598%	59.9419%	82.6940%	52.1741%	
	SMOTE-NC	Training	0.075336	99.2955%	75.2781%	99.2496%	79.6344%	709
		Testing	0.077694	99.2858%	74.5496%	98.6987%	78.9640%	
RF	None	Training	0.200992	97.4837%	80.4852%	98.8858%	86.8911%	191
		Testing	0.20116	97.4586%	77.8298%	98.8643%	84.4592%	
	SMOTE-NC	Training	0.195178	96.6396%	79.8083%	98.5854%	86.3145%	197
		Testing	0.195124	96.6341%	75.8464%	98.5543%	82.6850%	

(iii) *Training time*: it measures the required time to build the classification model

6.2. Evaluation of Legacy Learning Models. Logistic regression and random forest are evaluated under original and rebalanced datasets, and their results are shown in Table 7. Training and testing scores are almost similar for all the experiments, which confirm the absence of overfitting. As for logistic regression, SMOTE-NC oversampling leads to an improvement in precision, recall, and F1-score, which means that there is improvement in detecting minority classes. On the other hand, the oversampling does not improve the effectiveness of random forest.

6.3. Evaluation of Deep Learning Models. We conduct a series of experiments with different hyperparameter values (e.g., learning rate, batch size, number of layers, and number of units in each layer) in order to get the best performance. Different learning rates of the optimizer are tested. The best performance is achieved when the learning rate is 0.001. Also, different number of epochs 10, 15, 20, 50, and 100 and different batch sizes 100, 256, 512, and 1024 are tested. We can notice that increasing the number of epochs will slow down the learning process. Similarly, using a smaller batch size does not improve the performance. The number of epochs and the batch size for TCNN are set to 15 and 1024, respectively.

Figure 7 shows the accuracy and log loss of TCNN for multiclass classification during the training and validation phases. TCNN reaches high performance in the first epochs, which emphasizes that 15 epochs would be enough. Additionally, the training and validation results show the absence of overfitting. The log loss results of LSTM and CNN are shown in Figure 8. We can observe that TCNN outperforms LSTM and CNN in terms of log loss.

Tables 8–10 show the performance of TCNN, LSTM, and CNN, respectively. We can observe that deep learning models perform better than LR and RF, as some accuracy results exceed 99.99%. The accuracy results are very close but TCNN slightly outperforms LSTM and CNN in terms of effectiveness metrics. We can also observe that the deep learning models show good results even without applying dataset balancing. By applying, SMOTE-NC oversampling, we record an insignificant and very slight decrease in the effectiveness of TCNN and LSTM. On the other hand, the effectiveness of CNN slightly increases after applying

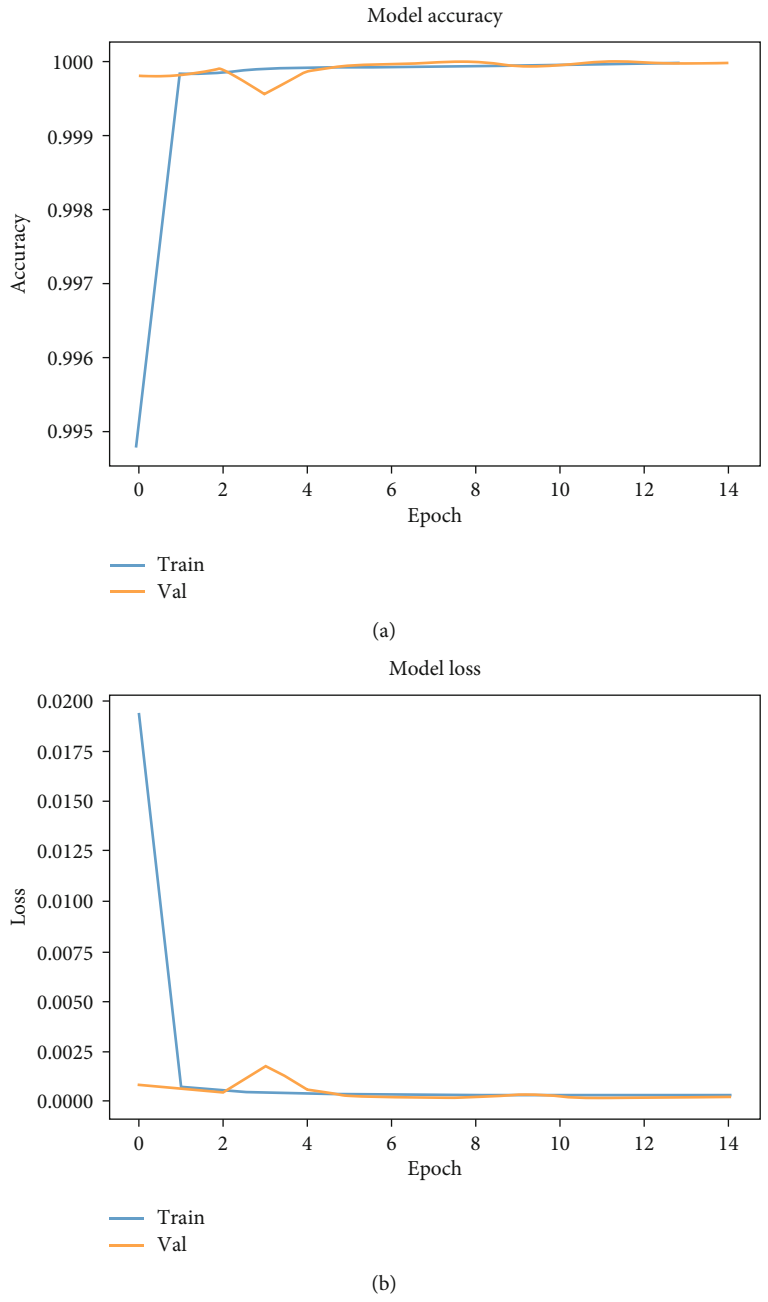


Figure 7: Accuracy and log loss of TCNN vs. number of epochs.

SMOTE-NC oversampling. CNN also incurs lower training time compared to TCNN and LSTM. TCNN offers a good trade-off between effectiveness and efficiency, as it is the closest competitor to CNN with respect to training time, and it records the best accuracy result.

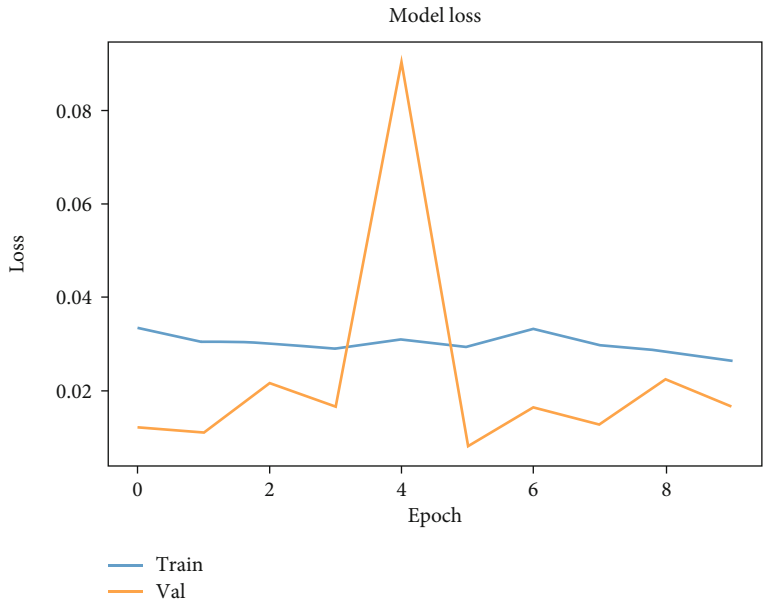
6.4. Comparison with Related Work Tested under Bot-IoT Dataset. In Table 11, we compare the performance of our work with other state-of-the-art methods that are tested under Bot-IoT dataset. The comparison is conducted with respect to accuracy, precision, recall, F1-score, training time, and classification task. According to the table, we can identify the following classification tasks:

(i) *Binary classification task*: it aims to distinguish between normal and attack records.

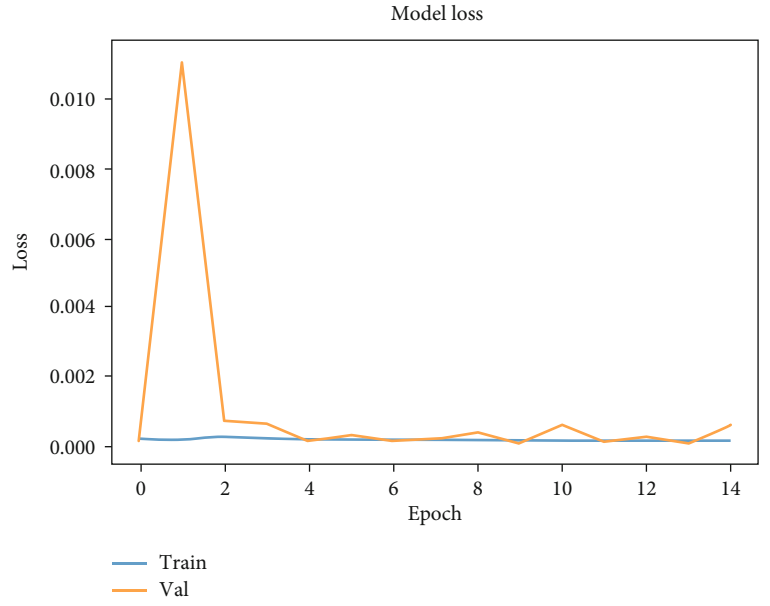
(ii) *Normal/one-attack classification task*: it aims to distinguish between normal records and one type of attacks.

(iii) *Multiclass classification*: it aims to attribute a record to its correct class among the five classes, i.e., one normal class and four attack classes.

It is known that multiclass classification is the most challenging task, whereas the normal/one-attack classification is the easiest one as the dataset only contains one type of attack, which means less diversity within the dataset, and easy



(a) LSTM



(b) CNN

Figure 8: Log loss of LSTM and TCN vs. number of epochs.

Table 8: Performance of TCNN model.

Oversampling	Log loss	Accuracy	Precision	Recall	F1-score	Training time (s)
None	0.000072	99.9986%	99.9974%	97.4975%	98.6641%	424
SMOTE-NC	0.000101	99.9978%	97.1379%	94.9972%	95.9961%	447

Table 9: Performance of LSTM model.

Oversampling	Log loss	Accuracy	Precision	Recall	F1-score	Training time (s)
None	0.002027	99.9654%	99.9443%	84.5703%	89.3016%	762
SMOTE-NC	0.002131	99.9643%	84.0246%	99.5169%	88.5303%	746

Table 10: Performance of CNN model.

Oversampling	Log loss	Accuracy	Precision	Recall	F1-score	Training time (s)
None	0.000094	99.9973%	95.1360%	97.0783%	96.0500%	419
SMOTE-NC	0.000118	99.9984%	99.9952%	94.9975%	97.1392%	490

Table 11: Comparison with related work tested on Bot-IoT dataset.

Ref	Model	Task	Accuracy	Precision	Recall	F1-score	Training time (s)
[49]	RNN	Binary	99.7404%	99.9904%	99.7499%	—	8035
	LSTM	Binary	99.7419%	99.9910%	99.7508%	—	10482.19
[61]	Ensemble learning	Binary	99.97%	—	—	—	—
[25]	RNN with BPTT	Multiclass	99.912%	—	—	—	2012
[50]	DeepDCA	Multiclass	98.73%	99.17%	98.36%	98.77%	—
[51]	ANN	Normal/DDoS	100%	100%	100%	100%	—
[52]	FNN	Multiclass	99.02%	—	—	—	—
Our	TCNN	Multiclass	99.9986%	99.9974%	97.4975%	98.6641%	424
	LSTM		99.9654%	99.9443%	84.5703%	89.3016%	762
Work	CNN		99.9973%	95.1360%	97.0783%	96.0500%	419
	LR		99.2858%	74.5496%	98.6987%	78.9640%	709
	RF		97.4586%	77.8298%	98.8643%	84.4592%	191

learning for the detection model. From Table 11, we can observe that [51] achieves 100% effectiveness. However, this result can be explained by the fact that [51] aims to distinguish between normal traffic and only one type of attack, i.e., DDoS. The three deep learning models, TCNN, LSTM, and CNN, outperform the rest of related work, although they are evaluated under multiclass classification task. We can also observe that TCNN, LSTM, and CNN incur the best results in terms of training time. This is due to the adopted feature engineering that reduces the computation complexity and due to the use of simple deep learning architectures with larger batch size and less number of layers.

7. Conclusion and Future Work

In this paper, we have identified five design principles for the development of an effective and efficient deep learning-based intrusion detection system for the Internet of Things (IoT). By adopting these principles, we have designed and implemented Temporal Convolution Neural Network (TCNN), which combines Convolution Neural Network (CNN) and causal convolution. TCNN is integrated with SMOTE-NC data balancing and efficient feature engineering, which consists of feature space reduction and feature transformation.

TCNN has been evaluated on Bot-IoT dataset and compared with logistic regression, random forest, LSTM, and CNN. Evaluation results show that TCNN achieves a good trade-off between effectiveness and efficiency. It outperforms the state-of-the-art deep learning IDS methods, which were tested under Bot-IoT dataset, by recording an accuracy of 99.9986% for multiclass traffic detection. Also, it shows a very close performance to CNN with respect to training time. As part of future work, it would be interesting to consider another design principle, i.e., testing the resiliency of IDS against adversarial attacks, which can confuse the deep learning model to produce wrong predictions.

Data Availability

We used the Bot-IoT, which is a publicly accessed dataset (https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php), for the evaluation of the proposed IDS.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group No (RG-1439-021).

References

[1] H. Belkhiri, A. Messai, M. Belaoued, and F. Haider, "Security in the internet of things: recent challenges and solutions," in *International Conference on Electrical Engineering and Control Applications*, pp. 1133–1145, Constantine, Algeria, 2019.

[2] Palo alto networks, "2020 unit 42 iot threat report," 2020, https://unit42.paloaltonetworks.com/iot-threat-report-2020/.

[3] M. Antonakakis, T. April, M. Bailey et al., "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security17)*, pp. 1093–1110, Vancouver, BC, Canada, 2017.

[4] S. Fadilpasic, "Researchers discover iot botnet capable of launching various ddos attacks," April 2020, https://www.itproportal.com/news/researchers-discover-iot-botnet-capable-of-launching-various-ddos-attacks/.

[5] J. Vijayan, "New malware family assembles iot botnet," April 2020, https://www.darkreading.com/iot/new-malware-family-assembles-iot-botnet–/d/d-id/1337578.

[6] A. Derhab, M. Guerroumi, A. Gumaei et al., "Blockchain and random subspace learning-based ids for sdn-enabled industrial iot security," *Sensors*, vol. 19, no. 14, p. 3119, 2019.

[7] M. Imran, M. H. Durad, F. A. Khan, and A. Derhab, "Toward an optimal solution against denial of service attacks in software defined networks," *Future Generation Computer Systems*, vol. 92, pp. 444–453, 2019.

[8] B. Du, H. Peng, S. Wang et al., "Deep irregular convolutional residual lstm for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 972–985, 2020.

[9] E. Bou-Harb, M. Debbabi, and C. Assi, "Big data behavioral analytics meet graph theory: on effective botnet takedowns," *IEEE Network*, vol. 31, no. 1, pp. 18–26, 2017.

[10] E. M. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "Scalable and robust unsupervised android malware fingerprinting using community-based network partitioning," *Computers & Security*, vol. 96, article 101932, 2020.

[11] M. Marjani, F. Nasaruddin, A. Gani et al., "Big IOT data analytics: architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[12] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, article 105124, 2020.

[13] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, article 102419, 2020.

[14] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, 2019.

[15] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5g health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, pp. 1–16, 2020.

[16] Z. Ning, P. Dong, X. Wang et al., "Partial computation offloading and adaptive task scheduling for 5g-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, 2020.

[17] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[18] Z. Ning, K. Zhang, X. Wang et al., "Joint computing and caching in 5g-envisioned internet of vehicles: a deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[19] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2018.

[20] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2020.

[21] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, 2020.

[22] A. Derhab, M. Belaoued, M. Guerroumi, and F. A. Khan, "Two-factor mutual authentication offloading for mobile cloud computing," *IEEE Access*, vol. 8, pp. 28956–28969, 2020.

[23] A. Boulemtafes, A. Derhab, and Y. Challal, "A review of privacy-preserving techniques for deep learning," *Neurocomputing*, vol. 384, pp. 21–45, 2020.

[24] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[25] M. A. Ferrag and L. Maglaras, "DeepCoin: a novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1285–1297, 2020.

[26] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[27] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: a deep learning–based intrusion detection framework for securing IoT," *Transactions on Emerging Telecommunications Technologies*, no. - article e3803, 2019.

[28] M. Kumar, *Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)*, Wright State University, 2017.

[29] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, pp. 452–457, Las Vegas, NV, USA, 2019.

[30] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against ddos attacks in iot networks," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 562–567, Las Vegas, NV, USA, 2020.

[31] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 1–6, Sydney, NSW, Australia, 2018.

[32] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the internet of things," *Sensors*, vol. 19, no. 9, article 1977, 2019.

[33] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, https://arxiv.org/abs/1609.04747.

[34] S. Hou, A. Saas, L. Chen, and Y. Ye, "Deep4maldroid: a deep learning framework for android malware detection based on linux kernel system call graphs," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*, pp. 104–111, Omaha, NE, USA, 2016.

[35] E. M. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "Maldozer: automatic framework for android malware detection using deep learning," *Digital Investigation*, vol. 24, pp. S48–S59, 2018.

[36] T. G. Kim, B. J. Kang, M. Rho, S. Sezer, and E. G. Im, "A multimodal deep learning method for android malware detection using various features," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 773–788, 2019.

[37] S. Ni, Q. Qian, and R. Zhang, "Malware identification using visualization images and deep learning," *Computers & Security*, vol. 77, pp. 871–885, 2018.

[38] G. Sun and Q. Qian, "Deep learning and visualization for identifying malware families," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2018.

[39] Z. Wang, J. Cai, S. Cheng, and W. Li, "Droiddeeplearner: identifying android malware using deep learning," in *2016 IEEE 37th Sarnoff Symposium*, pp. 160–165, Newark, NJ, USA, 2016.

[40] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computers & Security*, vol. 92, article 101752, 2020.

[41] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: a two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[42] Z. Li, Q. Zheng, P. Shen, and L. Jiang, "Intrusion detection using temporal convolutional networks," in *International Conference on Neural Information Processing*, pp. 168–178, Sydney, NSW, Australia, 2019.

[43] W.-H. Lin, P. Wang, B.-H. Wu, M.-S. Jhou, K.-M. Chao, and C.-C. Lo, "Behaviorial-based network flow analyses for anomaly detection in sequential data using temporal convolutional networks," in *Advances in E-Business Engineering for Ubiquitous Computing. ICEBE 2019*, pp. 173–183, Springer, 2019.

[44] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "Tr-ids: anomaly-based intrusion detection through text-convolutional neural network and random forest," *Security and Communication Networks*, vol. 2018, 9 pages, 2018.

[45] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[46] Q. Zhou, J. Wu, and L. Duan, "Recommendation attack detection based on deep learning," *Journal of Information Security and Applications*, vol. 52, article 102493, 2020.

[47] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic Computing*, vol. 6, no. 1, pp. 31–47, 2014.

[48] M. Kumar and C. Guria, "The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA- II) and its jumping gene adaptations for multi-objective optimization," *Information Sciences*, vol. 382-383, pp. 15–37, 2017.

[49] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[50] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, "Deepdca: novel network-based detection of iot attacks using artificial immune system," *Applied Sciences*, vol. 10, no. 6, p. 1909, 2020.

[51] Y. N. Soe, P. I. Santosa, and R. Hartanto, "Ddos attack detection based on simple ann with smote for iot environment," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pp. 1–5, Semarang, Indonesia, 2019.

[52] M. Ge, F. Xiping, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for iot networks," in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 256–25609, Kyoto, Japan, 2019.

[53] A. L.-H. Muna, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information security and applications*, vol. 41, pp. 1–11, 2018.

[54] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, https://arxiv.org/abs/1803.01271.

[55] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: a review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[56] 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.

[57] W. McKinney, *Pydata development team pandas: powerful python data analysis toolkit*, 2019.

[58] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.

[59] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.

[60] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pp. 167–172, Amman, Jordan, 2017.

[61] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," *Electronics*, vol. 8, no. 11, p. 1210, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 6661022, 11 pages
https://doi.org/10.1155/2020/6661022

Research Article

An Energy-Efficient Silicon Photonic-Assisted Deep Learning Accelerator for Big Data

Mengkun Li[1] **and Yongjian Wang**[2]

[1]*School of Management, Capital Normal University, Beijing 100089, China*
[2]*National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China*

Correspondence should be addressed to Mengkun Li; limengkun@cnu.edu.cn and Yongjian Wang; wyj@cert.org.cn

Received 15 November 2020; Revised 7 December 2020; Accepted 10 December 2020; Published 16 December 2020

Academic Editor: Xiaojie Wang

Deep learning has become the most mainstream technology in artificial intelligence (AI) because it can be comparable to human performance in complex tasks. However, in the era of big data, the ever-increasing data volume and model scale makes deep learning require mighty computing power and acceptable energy costs. For electrical chips, including most deep learning accelerators, transistor performance limitations make it challenging to meet computing's energy efficiency requirements. Silicon photonic devices are expected to replace transistors and become the mainstream components in computing architecture due to their advantages, such as low energy consumption, large bandwidth, and high speed. Therefore, we propose a silicon photonic-assisted deep learning accelerator for big data. The accelerator uses microring resonators (MRs) to form a photonic multiplication array. It combines photonic-specific wavelength division multiplexing (WDM) technology to achieve multiple parallel calculations of input feature maps and convolution kernels at the speed of light, providing the promise of energy efficiency and calculation speed improvement. The proposed accelerator achieves at least a 75x improvement in computational efficiency compared to the traditional electrical design.

1. Introduction

In a modern society driven by big data, artificial intelligence (AI) has brought great convenience to human life. As an indispensable part of solving complex problems in the field of AI, deep learning has been used in many applications, e.g., image and speech recognition, machine translation, self-driving, Internet of Things (IoTs), 5th generation (5G) mobile networks, and edge computing [1–13]. Deep learning can use effective learning and training methods to discover the inherent rules in the data model, thus helping machines to perform advanced reasoning tasks like human beings. In deep learning, convolutional neural networks (CNNs) are considered the most representative framework due to its advantages: the simple structure, few parameters, noticeable extraction features, and high recognition rate [14, 15]. Due to the enormous amount of data, the efficient inference of CNNs has high computing requirements. Therefore, the development of the hardware inference accelerator, which can provide strong computing power, is the key to meet the needs of CNNs.

At present, hardware accelerators that perform CNN operation mainly include GPUs, ASICs [16], FPGAs [17], TPU [18], and the emerging near data processing accelerator ISAAC [19]. However, current accelerators rely on a large degree of data movement. The energy consumption of electrical wire-based data movement is even greater than the energy consumed by the computing itself. Due to the widening gap between abundant data and limited power budget, these electric-based accelerators' energy crisis is still unpredictable. Limited by the transmittance rate of the electrical line, the calculation speed and throughput of these accelerators may not be able to keep up with the increase in power, resulting in limited throughput per second per watt.

Recently, silicon photonic technology has emerged as a promising solution to address the issues above [20–25]. Firstly, a certain transistor-based circuit's power consumption has a positive correlation with f^3 (f is the clock frequency). The

photonic circuit only consumes the power proportional to f, so that the photonic circuit can provide ultralow energy consumption [26]. Secondly, light has a very low transmission delay on a chip, typically 0.14 ps for 10 microns, which is 1–2 orders of magnitude faster than the transistor-based circuit [27]. Finally, the photonic circuit is insulated and has strong antielectromagnetic interference performance.

Furthermore, benefitting from the peaceful development of photonic integration technology and manufacturing platform, various mature active and passive building blocks have been demonstrated experimentally, such as modulators, photodetectors, splitters, wavelength multiplexers, and filters [28–31]. Based on these photonic devices, photonic computing elements such as photonic adders, differentiators, integrators, and multipliers can be realized [32–35]. Once the photonic devices can be successfully applied to the CNN accelerator's design, it is expected to improve energy efficiency in deep learning significantly. In addition, by utilizing optical multichannel multiplexing technologies, such as wavelength division multiplexing (WDM) [36–38], we can easily use the speed of light to achieve massively parallel computing to improve the inference speed of CNNs significantly.

Thus, we propose a silicon photonic-assisted CNN accelerator for deep learning. We first use the mature microring resonators (MRs) as the basic unit to design a photonic matrix-vector multiplier (PMVM) to perform the most complex convolution operation on CNNs. Then, we introduce an analytical model to identify the number of MRs used, power consumption, area, and execution time in each layer of the CNNs. At last, we introduce our PMVM-based photonic-assisted CNN accelerator architecture and its workflow. The simulation results show that our accelerator can increase the CNN's inference speed by at least 75 times under the same energy consumption than the current electricity-based accelerators.

The rest of the paper is organized as follows. Section 2 briefly discusses the related works. Section 3 discusses the proposed PMVM and accelerator architectures, followed by Section 4 presenting the performance evaluation of the silicon photonic-assisted accelerator. Section 5 concludes this paper.

2. Related Work

In this section, we first describe CNNs' structure and computing process in deep learning. Then, we introduce photonic devices that might be used. These related works can be used as the guide for our research on the photonic-assisted accelerator design.

2.1. Convolutional Neural Network (CNN) Basics. CNN is comprised of stacking multiple computation layers for feature extraction and classification. Compared to the fully neural networks with simple training but limited scalability, CNN has very deep convolutional (CONV), pooling (POOL), and full connection (FC) layers. Therefore, it can achieve high accuracy [14]. In each CONV layer, the input maps are transformed into highly abstract representation feature maps and convolution with the kernel to generate output feature maps. After nonlinearity and pooling, the output features can be used as the input for the next layer. After multi-CONV and POOL layers, the features are sent to the FC layers and finally output the classification results. The CONV layers take more than 90% of the calculation time [39]. Therefore, the design of an optimization accelerator for CONV layers can significantly improve the entire CNN's performance. Figure 1 shows a CONV layer. It has M 3D convolutional kernels with size $S \times R \times C$ and N input maps with size $W \times H \times C$. M kernels perform M times 3D convolution on the input maps with a sliding stride of S and generate an $E \times F \times M$ output map. In each output map, the value of the element (m, f, e) can be computed as

$$O(m, f, e) = \sigma\left(\sum_{c=0}^{C-1}\sum_{i=0}^{S-1}\sum_{j=0}^{R-1} K[m][c][i][j] \times I[c][f * S + i][e * R + j]\right), \tag{1}$$

where I, K, and O are the input, kernel, and output matrices, respectively. $\sigma(\cdot)$ is an activation function, such as ReLU and sigmoid. The pseudocode to perform this normal convolution operation is shown in Figure 1. Note that in each layer, all kernels share the same input data. Therefore, if the accelerator can support multiple kernels that simultaneously convolve with the same input data, the number of access buffers is reduced. The cycle time can also be reduced, thereby increasing the throughput. As shown in the pseudocode, assuming the input map can be reused by G_m kernels simultaneously, the total convolution cycles can be saved by G_m time. The size of G_m is determined by the accelerator. Therefore, designing the corresponding accelerator architecture to maximize this data reuse capability is the paper's primary motivation.

2.2. Silicon Photonic Devices. Microelectronic devices are the basis of the current CNN accelerator. But with the reduction of feature size, the ability of electronic information processing has approached its limit. Silicon photonic devices offer an exact route to solve the electrical processing bottleneck due to its low loss, high speed, low energy consumption, and compatibility with CMOS platforms. Among the various silicon photonic devices, MRs are considered the most critical devices in photonic computing due to their excellent wavelength selection characteristics, small size, high modulation rate, low energy consumption, and high-quality factors [40, 41]. Figure 2 shows two commonly used MR structures: all-pass MR (Figure 2(a)) and 1×2 cross-MR (Figure 2(e)). All-pass MRs include one straight waveguide and one MR, assuming that the resonant wavelength of the MR is λ_{mr} and the input signal wavelength is λ_{in}. When $\lambda_{in} = \lambda_{mr}$, the input signal will be wholly coupled into the MR, so that the signal power output from the through port is zero (transmittance rate is 0). When $\lambda_{in} \neq \lambda_{mr}$, the coupling ability between the input waveguide and the MR will become weak, and when it is weak enough, the signal will output from the through port (transmittance rate is 1). When the MR's resonance wavelength is between λ_1 and λ_2, the transmittance rate of the MR will be between 0 and 1.

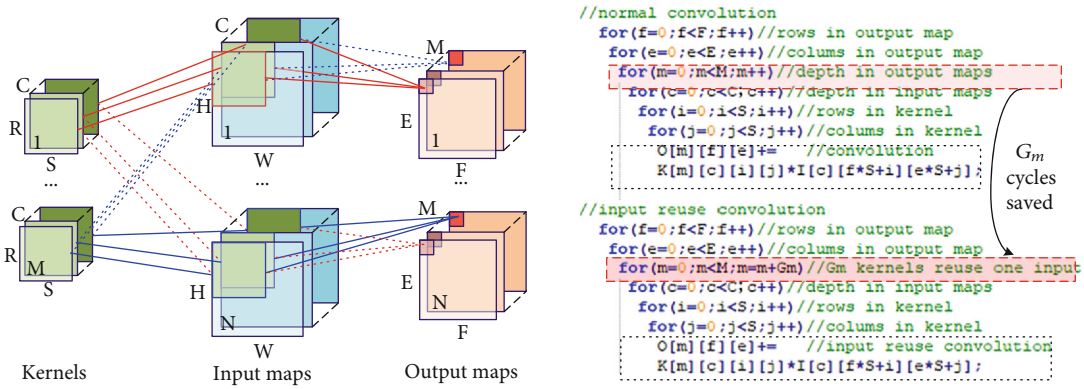


Figure 1: The logical graph and pseudocode for standard convolution and input map reuse convolution of a CONV layer.

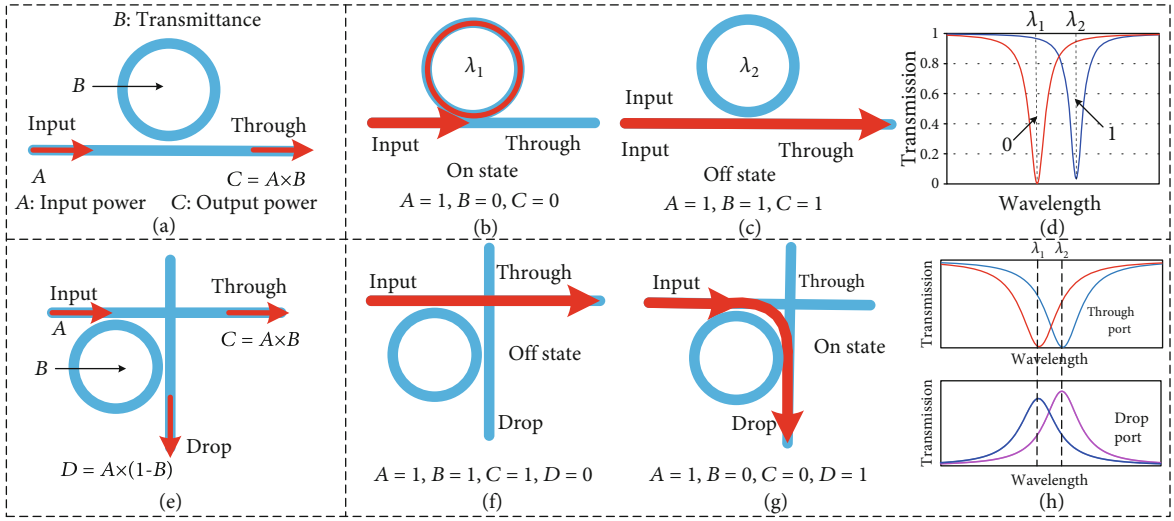


Figure 2: (a) All-pass MR photonic computing unit. (b, c) Computing process with on-state MR and off-state MR. (d) The transmission spectrum lines of the through port for all-pass MR with different wavelengths. (e) 1×2 cross-MR photonic computing unit. (f, g) Computing process with off-state MR and on-state MR. (h) The transmission spectrum lines of through and drop ports for 1×2 cross-MR with different wavelengths.

Therefore, we can use the resonance effect of MR to adjust the output power to realize the photonic multiplication calculation. For instance, as shown in Figure 2(a), assuming that the input optical signal power is A, the transmittance of the MR is B ($0 \le B \le 1$). When the input optical signal passes through the MR, part of the light $(1 - B)$ will be coupled to the MR, and the output optical power of the through port is $C = A \times B$. Usually, by adding a bias voltage to the MR, the transmittance rate of MR (B) can be changed under the thermooptic or electrooptic effect. According to [34], each MR can store more than 16 levels of transmittance rate (i.e., 4 bits). Therefore, for a 16-bit floating-point calculation [19], only 4 MRs are needed. Figure 2(e) shows the structure of 1×2 cross-MR, which has the same working principle as the all-pass MR. The output powers of the through and drop can be controlled by controlling the MR's resonant wavelength, as shown in Figures 2(f)–2(h). Since the multiplication operation of the above two structures can be realized in the optical domain, they have a high processing speed, making them ideal choices for photonic multiplication units.

3. Silicon Photonic-Assisted CNN Accelerator Architecture Design

In order to use silicon photonic technology to improve the calculation rate in deep learning, we first propose a PMVM based on photonic devices in this section. Then, we create a photonic-assisted CNN accelerator architecture based on PMVM.

3.1. Silicon Photonic Matrix-Vector Multiplier. Matrix-vector multiplication is the most important operation in CNN. Therefore, in this section, we will use the essential photonic devices to construct a PMVM and map the input feature map and kernel weight data to the PMVM to complete the parallel multiplication operation.

Figure 3 shows the PMVM architecture. It relies on an all-pass MR-based input matrix and 1×2 cross-MR-based kernel matrix. Current CNNs have tens of kernels in each layer to convolve the same set of input data. Therefore, in PMVM, we multiplex the input data to be convolved with multiple kernels simultaneously, reducing the waste of time

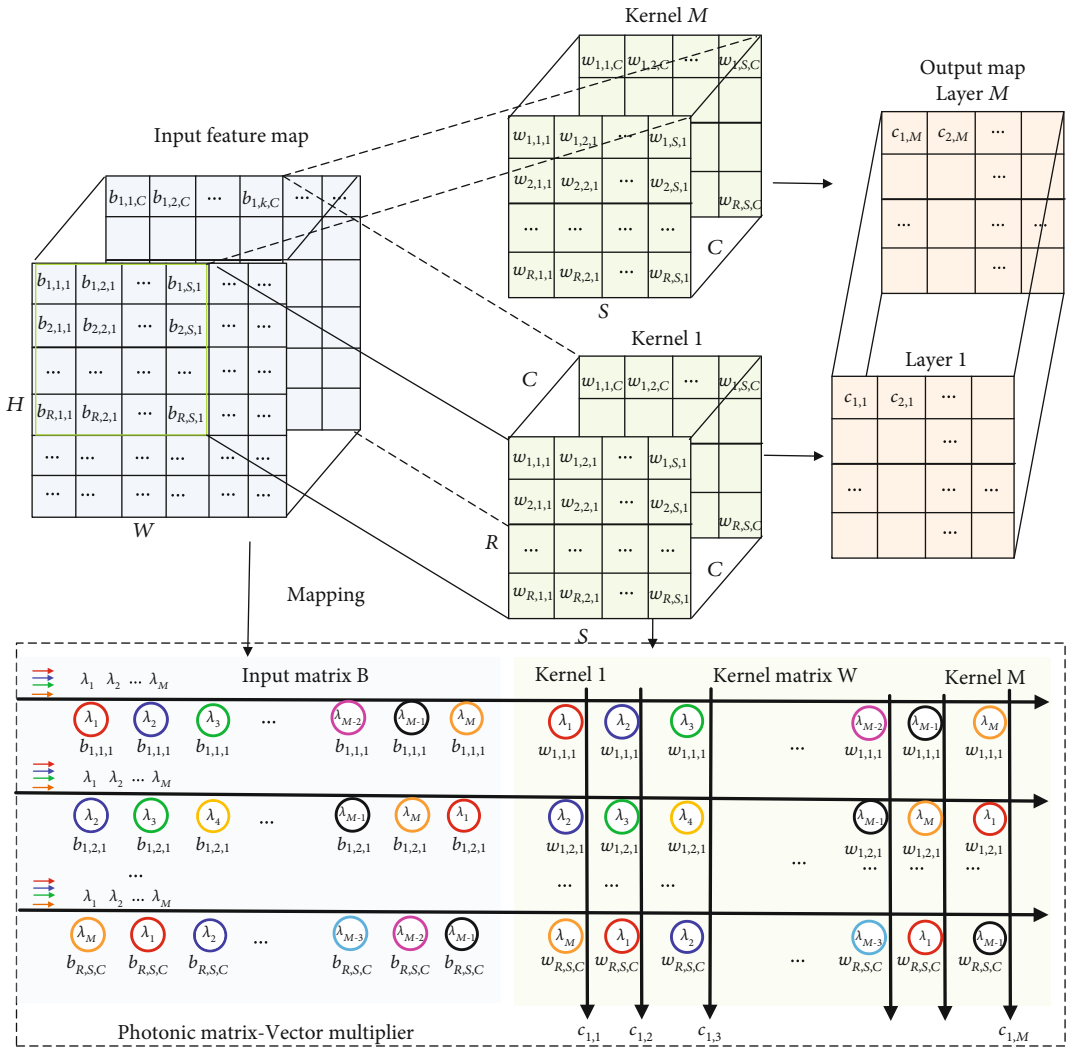


Figure 3: Photonic matrix-vector multiplier.

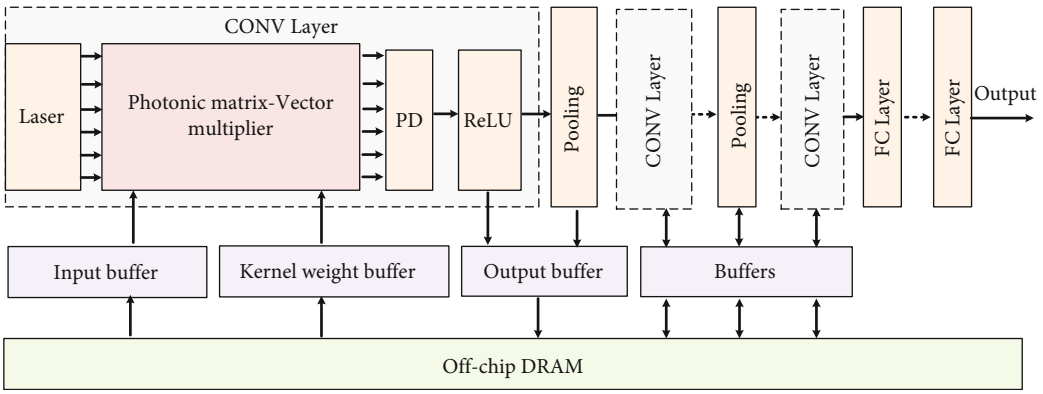
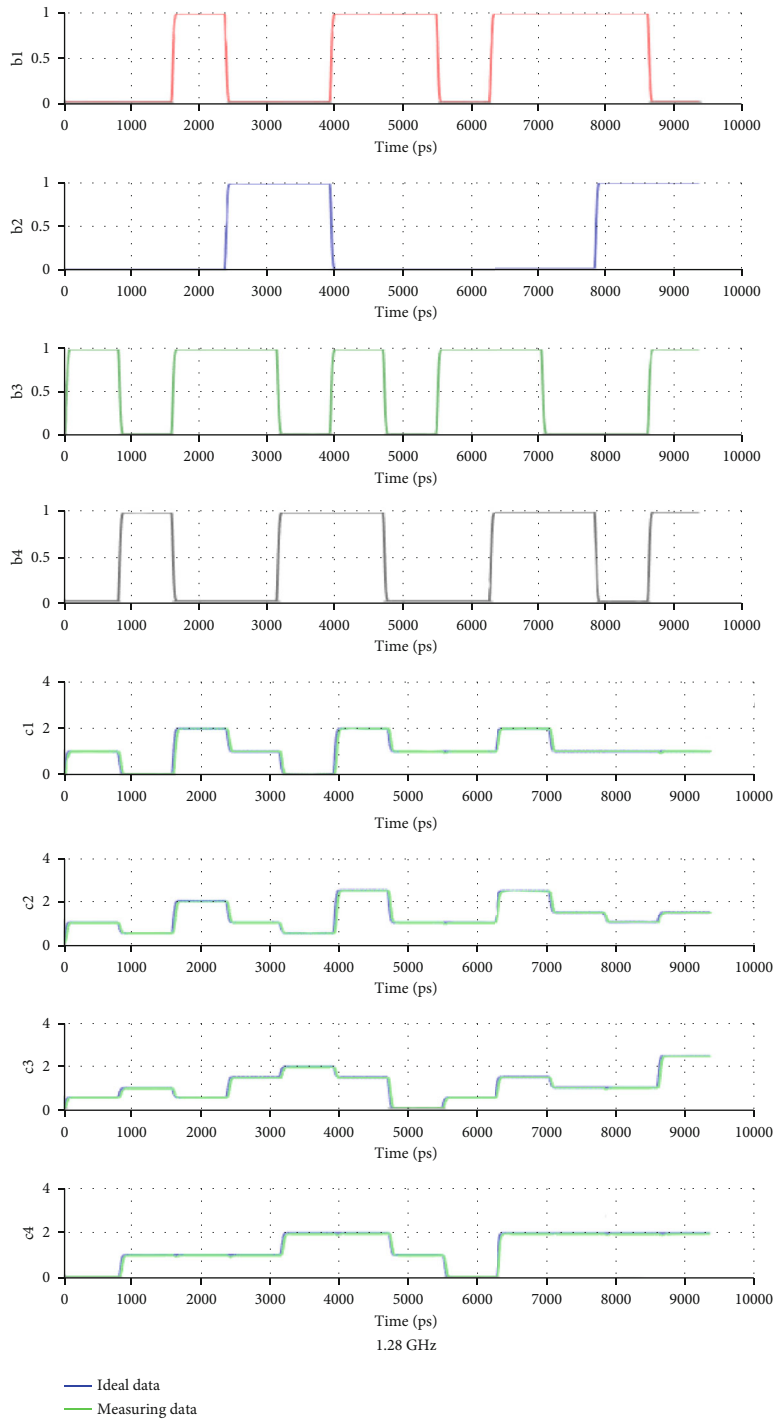


Figure 4: Photonic-assisted CNN inference accelerator architecture.

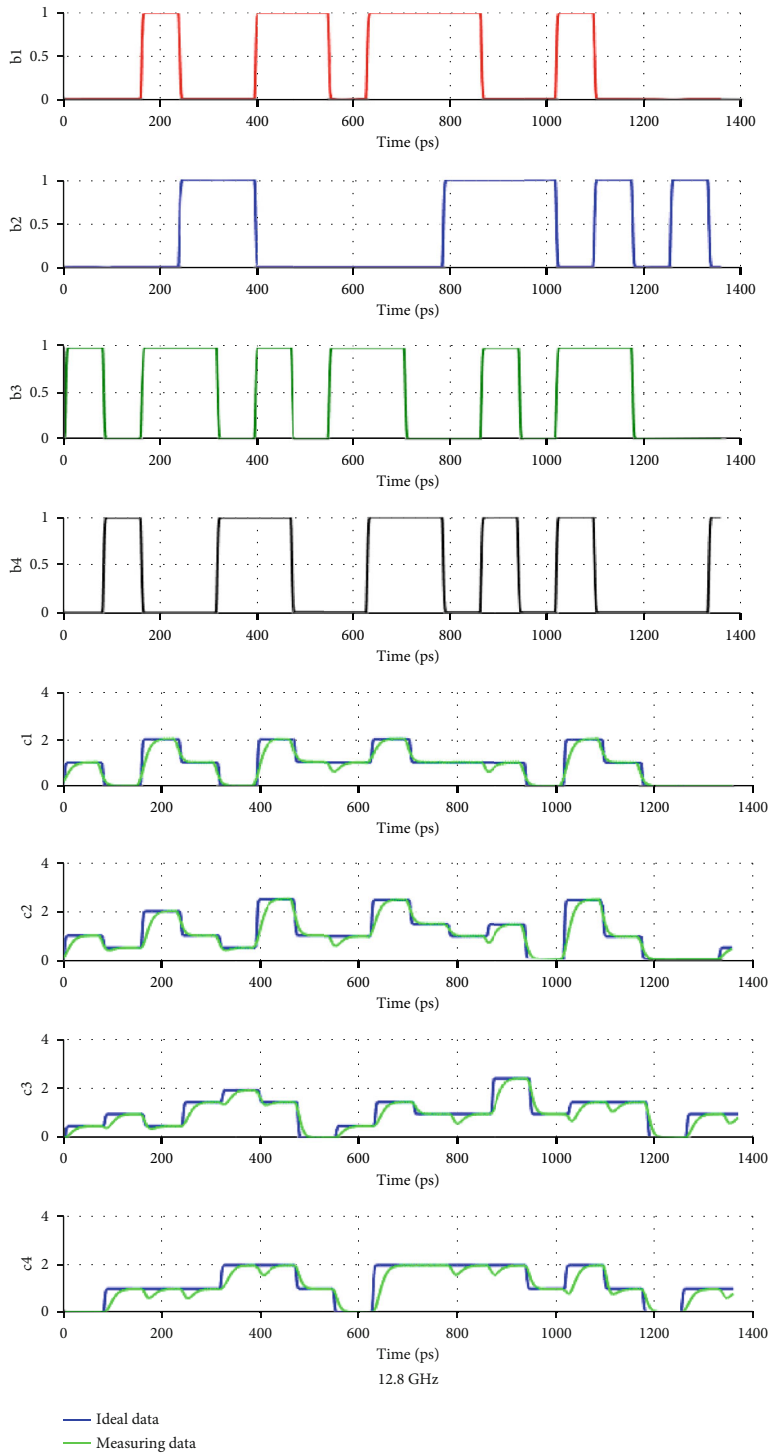
and energy consumption caused by repeated reading of the input data. For convenience, if we assume that the size of each kernel is $R \times S \times C$, the number of the kernels is M. The weight matrix W in PMVM can be composed of an $(R \times S \times C) \times M$ MR-based crossbar array. The MR in the array has different resonance wavelengths to ensure parallel computing. The MR would be on resonance when the wavelength of the light fits a whole number of times inside the optical length of the MRs:

$$\lambda_{\text{res}} = \frac{n_{\text{eff}} L}{m}, \quad L = 2\pi R, m = 1, 2, 3 \cdots . \tag{2}$$



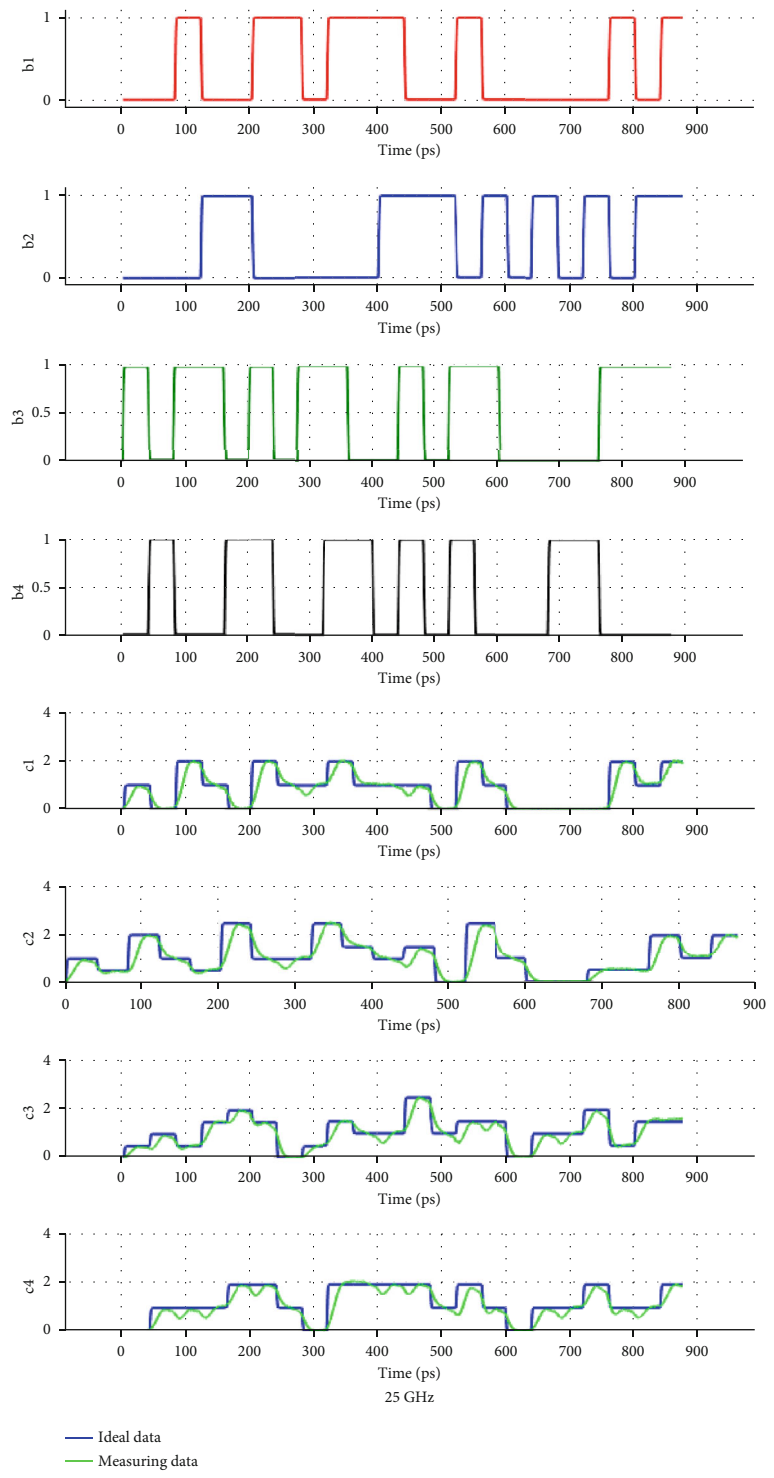
(a)

Figure 5: Continued.



(b)

Figure 5: Continued.



(c)

Figure 5: The simulation waveforms for 4 × 4 PMVM with (a) 1.28 GHz; (b) 12.8 GHz; (c) 25 GHz.

Here, λ_{res} is the resonant wavelength, n_{eff} is the effective refractive index, and R is the radius of the MRs, respectively. Therefore, in this paper, we use MRs with different radii to realize the control of different resonance wavelengths.

As shown in Figure 3, the weight value of the coordinate (i, j, n) in the m-th kernel can be represented by the drop port transmittance rate of the m-column and $((n-1) \times S \times R + (i-1) \times S + j)$ -row MR in the crossbar array, where $0 < i < S, 0 < j < R, \quad 0 < n < C,$ and $0 < m < M$. According to CNN's characteristics, the state of all MRs in the kernel matrix remains unchanged during the inference process. In PMVM, the feature data of the input feature maps are

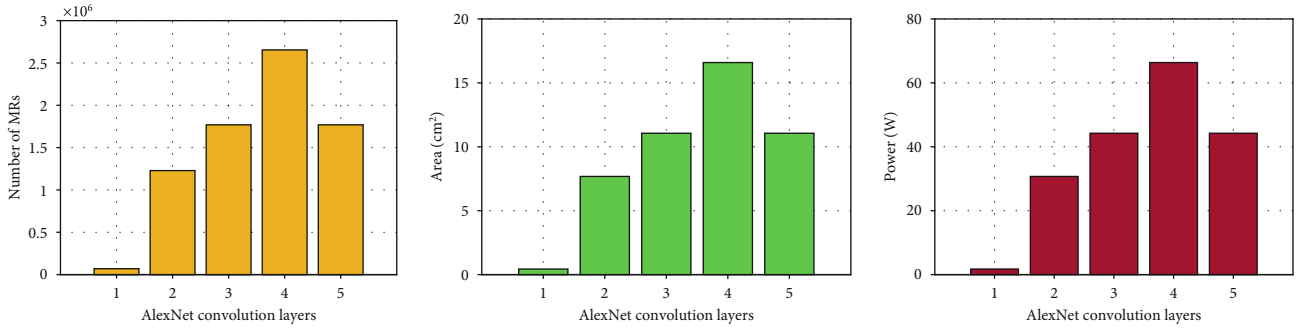


Figure 6: Total number of MRs required, area occupied, and power consumption for different convolutional layers of AlexNet.

mapped to the input matrix in turn. The input matrix comprises all-pass MR, and the size is the same as the kernel matrix. The values of the MR in the input matrix are updated with the sliding window. As shown in Figure 3, assuming the stride of the sliding window is 1, the value of MR with wavelength $\lambda_{1,1}$ is $b_{1,1,1}$ at time t_1, and it will be updated to $b_{1,2,1}$ at time t_2. In this PMVM, the multi-wavelength optical signals emitted by the lasers are injected from the input port of the input matrix and output from the kernel matrix after photonic multiply-accumulate (MAC) operation. The output power is the sum of all wavelength signals. As shown in Figure 3, the calculation process of the PMVM at time t_1 is

$$\begin{aligned}&[b_{1,1,1}, b_{1,2,1}, \cdots, b_{R,S,C}]\\ &\times \begin{bmatrix} w_{1,1,1(\text{kernel }1)} & w_{1,1,1(\text{kernel }2)} & \cdots & w_{1,1,1(\text{kernel }M)} \\ w_{1,2,1(\text{kernel }1)} & w_{1,2,1(\text{kernel }2)} & \cdots & w_{1,2,1(\text{kernel }M)} \\ \cdots & \cdots & \cdots & \cdots \\ w_{R,S,C(\text{kernel }1)} & w_{R,S,C(\text{kernel }2)} & \cdots & w_{R,S,C(\text{kernel }M)} \end{bmatrix}\\ &= [c_{1,1}, c_{1,2}, \cdots, c_{1,M}].\end{aligned} \tag{3}$$

Therefore, the PMVM enables all MAC operations to finish with high parallelism. According to [39], the number of multiplexed wavelengths can reach 128. Thus, the computation speed of the PMVM will be $128 \times 128 \times 10 \times 10^{10} = 1.6384 \times 10^{15}$ MAC/s when all MRs work at 10 Gb/s modulation speed.

3.2. Silicon Photonic-Assisted Accelerator Architecture Design. Based on the PMVM, we propose a photonic-assisted CNN accelerator architecture, as shown in Figure 4. The accelerator consists of multilayer CONV layers, pooling layers, and FC layers, and all layers are processed sequentially. According to different CNN models, the distribution between layers can be adjusted. The proposed PMVM is deployed in the CONV layers. The input matrix and kernel matrix values are read from the off-chip DRAM (the off-chip DRAM data will be sent to the on-chip buffer first). Once the CNN model is sufficiently trained, the weight values of kernels in each layer are determined and programmed into PMVMs by configuring each MR's transmittance rate in the kernel matrix. During the whole process, only the value of the input matrix will be updated. After highly parallel MAC operations, the output optical signals are converted into the electrical signals by photodetectors (PDs) and then activated and pooled. This process can be done very fast because all the photonic-assisted devices' operating frequency can reach tens of GHz, e.g., lasers, MR, and PD. The calculation results are stored back to the off-chip DRAM for reading and calculation of the next layer. After multiple layers of convolution, pooling, and full interconnection operations, the accelerator will output the final inference results.

Table 1: Execution time for convolution layers of AlexNet ($P = 0$, $S = 1$).

CONV layers	Input patch size	Kernel size	Execution time (μs)
1	55×55	11×11	337.561
2	27×27	5×5	19.881
3	13×13	3×3	1.0368
4	13×13	3×3	1.0368
5	13×13	3×3	1.0368

4. Simulation Evaluations

In this section, we used a widely adopted deep learning accelerator simulator, FODLAM [42], to evaluate the performance of our accelerator. FODLAM does total up the latency and energy for each layer, including the storage and read/write costs of the intermediate layers. The simulation of the photonic part of our accelerator structure is performed using a professional optical simulation platform, i.e., Lumerical Solutions [43]. The configuration parameters of other accelerators are obtained from the prior art as referenced.

4.1. Photonic Matrix Multiplication Function Verification. The photonic vector multiplication results of $B \times W$ with different working frequencies are exhibited in Figure 5. Assuming the matrix size is 4×4, we perform the simulation using four CW lasers with different working wavelengths. The input matrix ($B = [b1\,;\, b2\,;\, b3\,;\, b4]$) is modulated by four 2^7-1 pseudorandom binary sequence (PRBS) from the pattern generators. The values in the kernel matrix W are randomly generated once programmed into the corresponding MR

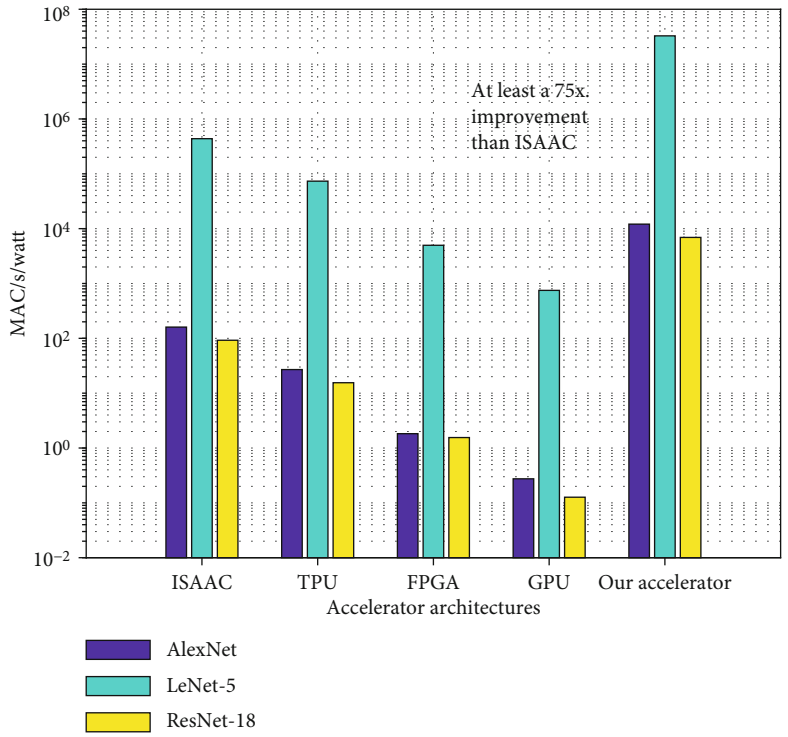


Figure 7: The inference performance of different accelerators under different CNN models.

units with $W = [1, 0, 0.5, 1\,;\, 0, 1, 1, 1\,;\, 1, 0.5, 0, 1\,;\, 0, 1, 1, 0]$, which is fixed throughout the simulation. The simulation output $C = [c1, c2, c3, c4]$ results from the multiply-accumulate of W and B.

It can be seen from Figure 5 that when PMVM works at 1.28 GHz, the simulation results are almost the same as the ideal results. Although a particular error will occur as the operating frequency increases, the designed PMVM can also maintain good calculation accuracy under the operating frequency of 25 GHz.

4.2. Area and Power Consumption Evaluation Models. The area of PMVM is affected by MRs. According to [44], the area of each MR unit is $25\,\mu\text{m} \times 25\,\mu\text{m}$ with 0.025 mW energy consumption. The size of the kernel determines the number of MRs used in PMVM. For example, the first CONV layer of the AlexNet architecture contains 96 kernels, and the size of each kernel is $11 \times 11 \times 3$. Assuming that a set of input data completes all convolution operations of this layer within one cycle, theoretically, the PMVM of this layer needs 69,696 MRs. The area and power of PMVMs in this layer are 43.56 mm^2 and 1.74 W, respectively. Due to the current technological limitations, it is difficult to integrate so many MRs on a single chip. Therefore, multiple interconnected chips are usually used to complete the above functions [19, 39]. Figure 6 shows the number of MRs, occupied area, and power consumption in each convolutional layer of AlexNet. It can be seen that the fourth layer of AlexNet has the largest consumption because this layer has the largest convolution kernel.

4.3. Execution Time Evaluation Models. As mentioned in the previous section, our PMVM can compute convolutions of multiple kernels in parallel for a single input data within one cycle. In AlexNet, the length and width of the input patches are the same. Assuming the size of input patches is $W \times W$, the kernel size is $K \times K$, the padding size is P, and the stride is S. Thus, the number of convolution calculations for each input patch is

$$N_{\text{Calculation}} = \left(\left[\frac{W - K + 2P}{S} \right] + 1 \right)^2. \tag{4}$$

Thus, the computation time of each input patch is

$$T = \frac{N_{\text{Calculation}}}{f_{\text{PMVM}}}, \tag{5}$$

where f_{PMVM} is the operating frequency of the PMVM.

Assuming $P = 0$ and $S = 1$, the execution time results for each layer of AlexNet as shown in Table 1 when the working frequency of the PMVM is 25 GHz.

4.4. Inference Performance. To fully evaluate our accelerator's inference performance, the energy-efficient performance is considered in our simulation, i.e., MAC/s/watt. We compared our accelerator with GPU, FPGA, TPU, and ReRAM-based CNN accelerator ISAAC. The CNN architecture are AlexNet, LeNet-5, and ResNet-18, and the database are ImageNet (AlexNet and ResNet-18) and MNIST (LeNet-5). In the simulation, we use the parameters of the electrical devices listed in Ref. [19]. The simulation results of MAC/s/watt are shown in Figure 7. Compared to other electricity-based accelerators, our accelerator can increase energy efficiency

by at least 75 times because it can use silicon photonics' advantages to increase computing speed while reducing energy consumption.

5. Conclusions

This paper proposed a silicon photonic-assisted CNN accelerator to maximize the inference performance in deep learning. It achieved a high inference throughput by exploiting the high modulation rate MRs and WDM technology. The proposed accelerator achieves at least 75x improvement in computational efficiency compared to the state-of-the-art designs. The photoelectric hybrid CNN accelerator needs to match the operating frequency of the electronic device, which affects the performance of the photonic device. In the future, we will explore the all-optical accelerators to maximize acceleration performance.

Data Availability

Data are available on request. The data are available by contacting Mengkun Li (limengkun@cnu.edu.cn).

Conflicts of Interest

The authors declare that there is no conflict of interest.

Acknowledgments

This research was funded by the Major Technology Project of China National Machinery Industry Corporation (SINOMACH): "Research and Application of Key Technologies for Industrial Environment Monitoring, Early Warning and Intelligent Vibration Control (SINOMAST-ZDZX-2017-05)," and partially supported by the Scientific Research Foundation of the Beijing Municipal Education Commission (KM201810028021).

References

[1] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, 2019.

[2] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: a survey," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2020.

[3] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2021.

[4] J. Chen and X. Ran, "Deep learning with edge computing: a review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[5] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[6] Z. Q. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[7] Z. Ning, R. Y. K. Kwok, K. Zhang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[8] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[9] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[10] S. Huang, C. Yang, S. Yin, Z. Zhang, and Y. Chu, "Latency-aware task peer offloading on overloaded server in multi-access edge computing system interconnected by metro optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5949–5961, 2020.

[11] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications, To Appear*, pp. 1–6, 2020.

[12] Z. Ning, P. Dong, X. Wang et al., "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[13] W. Wang, H. Huang, L. Zhang, and C. Su, "Secure and efficient mutual authentication protocol for smart grid under blockchain," *Peer-to-Peer Networking and Applications*, 2020.

[14] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017.

[15] 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.

[16] A. Graves, G. Wayne, M. Reynolds et al., "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[17] C. Farabet, C. Poulet, J. Han, and Y. LeCun, "CNP: an FPGA-based processor for convolutional networks," in *IEEE International Conference on Field Programmable Logic and Applications*, pp. 32–37, Prague, Czech Republic, 2019.

[18] N. P. Jouppi, C. Young, N. Patil et al., "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pp. 1–12, Toronto, ON, Canada, 2017.

[19] A. Shafiee, A. Nag, N. Muralimanohar et al., "ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[20] L. Guo, Z. Ning, W. Hou, B. Hu, and P. Guo, "Quick answer for big data in sharing economy: innovative computer architecture design facilitating optimal service-demand matching," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1494–1506, 2018.

[21] P. Guo, W. Hou, L. Guo, Q. Yang, Y. Ge, and H. Liang, "Low insertion loss and non-blocking microring-based optical router for 3d optical network-on-chip," *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–10, 2018.

[22] J. Feldmann, N. Youngblood, C. Wright, H. Bhaskaran, and W. H. P. Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature*, vol. 569, no. 7755, pp. 208–214, 2019.

[23] P. Guo, W. Hou, L. Guo et al., "Fault-tolerant routing mechanism in 3d optical network-on-chip based on node reuse," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 547–564, 2020.

[24] Y. Shen, N. C. Harris, S. Skirlo et al., "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[25] L. Chen, K. Preston, S. Manipatruni, and M. Lipson, "Integrated GHz silicon photonic interconnect with micrometer-scale modulators and detectors," *Optics Express*, vol. 17, no. 17, pp. 15248–15256, 2009.

[26] Z. Ying, C. Feng, Z. Zhao et al., "Electronic-photonic arithmetic logic unit for high-speed computing," *Nature Communications*, vol. 11, no. 1, article 2154, 2020.

[27] Z. Ying, Z. Wang, Z. Zhao et al., "Silicon microdisk-based full adders for optical computing," *Optics Letters*, vol. 43, no. 5, pp. 983–986, 2018.

[28] T. Baba, S. Akiyama, M. Imai et al., "50-Gb/s ring-resonator-based silicon modulator," *Optics Express*, vol. 21, no. 10, pp. 11869–11876, 2013.

[29] J. Michel, J. Liu, and L. C. Kimerling, "High-performance Ge-on-Si photodetectors," *Nature Photonics*, vol. 4, no. 8, pp. 527–534, 2010.

[30] Y. Urino, Y. Noguchi, M. Noguchi et al., "Demonstration of 12.5-Gbps optical interconnects integrated with lasers, optical splitters, optical modulators and photodetectors on a single silicon substrate," *Optics Express*, vol. 20, no. 26, pp. B256–B263, 2012.

[31] H. Jia, L. Zhang, J. Ding, L. Zheng, C. Yuan, and L. Yang, "Microring modulator matrix integrated with mode multiplexer and de-multiplexer for on-chip optical interconnect," *Optics Express*, vol. 25, no. 1, pp. 422–430, 2017.

[32] Z. Ying, S. Dhar, Z. Zhao et al., "Electro-optic ripple-carry adder in integrated silicon photonics for optical computing," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 24, no. 6, pp. 1–10, 2018.

[33] J. Dong, A. Zheng, D. Gao et al., "High-order photonic differentiator employing on-chip cascaded microring resonators," *Optics Letters*, vol. 38, no. 5, pp. 628–630, 2013.

[34] M. Ferrera, Y. Park, L. Razzari et al., "On-chip CMOS-compatible all-optical integrator," *Nature Communications*, vol. 1, no. 1, article 29, 2010.

[35] L. Yang, R. Ji, L. Zhang, J. Ding, and Q. Xu, "On-chip CMOS-compatible optical signal processor," *Optics Express*, vol. 20, no. 12, pp. 13560–13565, 2012.

[36] F. Liu, H. Zhang, Y. Chen, Z. Huang, and H. Gu, "WRH-ONoC: a wavelength-reused hierarchical architecture for optical network on chips," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1912–1920, Kowloon, Hong Kong, April 2015.

[37] P. Guo, W. Hou, L. Guo, Z. Cao, and Z. Ning, "Potential threats and possible countermeasures for photonic network-on-chip," *IEEE Communications Magazine*, vol. 58, no. 9, pp. 48–53, 2020.

[38] P. Guo, W. Hou, L. Guo, Z. Ning, M. S. Obaidat, and W. Liu, "WDM-MDM silicon-based optical switching for data center networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[39] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang, "Holylight: a nanophotonic accelerator for deep learning in data centers," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1483–1488, Florence, Italy, March 2019.

[40] W. Bogaerts, P. de Heyn, T. van Vaerenbergh et al., "Silicon microring resonators," *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.

[41] P. Guo, W. Hou, and L. Guo, "Designs of low insertion loss optical router and reliable routing for 3D optical network-on-chip," *Science China Information Sciences*, vol. 59, no. 10, article 102302, 2016.

[42] A. Sampson and M. Buckler, "FODLAM, a first-order deep learning accelerator model," https://github.com/cucapra/fodlam.

[43] https://www.lumerical.com/cn/.

[44] A. N. Tait, T. F. de Lima, E. Zhou et al., "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific Reports*, vol. 7, no. 1, article 7430, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8818616, 9 pages
https://doi.org/10.1155/2020/8818616

Research Article

Image Annotation via Reconstitution Graph Learning Model

Shi Chen,[1] **Meng Wang**,[2] **and Xuan Chen**[3]

[1]*Emporia State University, Kansas, USA*
[2]*Shandong Port Group Co., Ltd., Shandong, China*
[3]*Dalian University of Technology, Dalian, China*

Correspondence should be addressed to Meng Wang; dlutwangmeng@163.com

Received 20 September 2020; Revised 13 November 2020; Accepted 28 November 2020; Published 14 December 2020

Academic Editor: Xiaojie Wang

With great developments of computing technologies and data mining methods, image annotation has attracted much attraction in smart agriculture. However, the semantic gap between labels and images poses great challenges on image annotation in agriculture, due to the label imbalance and difficulties in understanding obscure relationships of images and labels. In this paper, an image annotation method based on graph learning is proposed to accurately annotate images. Specifically, inspired by nearest neighbors, the semantic neighbor graph is introduced to generate preannotation, balancing unbalanced labels. Then, the correlations between labels and images are modeled by the random dot product graph, to deeply mine semantics. Finally, we perform experiments on two image sets. The experimental results show that our method is much better than the previous method, which verifies the effectiveness of our model and the proposed method.

1. Introduction

With great developments of computing technologies and data mining methods, smart agriculture has attracted much attraction since it can greatly increase crop yields by effectively recommending methods to control pests [1, 2]. For example, the internet of vehicles with task scheduling [3, 4] can help formers to harvest crops automatically, and content-based crop image retrieval can help producers to keep track of plant growth in real time, which contributes to developing disease control and production plans. Meanwhile, with the technological advancement, the form of crop monitoring is also undergoing tremendous changes, posing great challenges to the current machine learning-based methods [5–9], due to the collected data that are of high volume, high velocity, high value, and high variety [10, 11]. Thus, to mine patterns of data in smart agriculture requires novel methods.

Image annotation, as a typical method for images analysis in agricultural big data, predicts labels for a given image, which can well match the image content [12]. In recent years, a large number of researchers have done extensive research on image annotation [13, 14]. For example, to reduce the semantic gap between visual features and text features, some researchers have proposed the generative model, which models image annotation as a joint likelihood distribution between images and labels. Nevertheless, the generative method only uses the image-label correlations, ignoring the relation over images. To use the relation over images, the discriminant model is proposed, focusing on finding the difference between images. Typically, this method trains a classifier to predict image labels, but the balance of sample labels has a large impact on the model performance. At the same time, some researchers proposed a graph model that utilizes all the data to build the intrinsic structure of unlabeled images and annotated images. Also, the nearest neighbor model is used to construct the label propagation graph, based on the theory that similar images share common labels [15, 16]. However, this method pays too much attention to the correlation between images, ignoring the image differences.

To solve those problems, a nearest neighbor graph model is proposed in this paper, which combining superiorities of graph and K nearest theories. Specifically, the semantic neighbors of test image under each label are firstly searched to the semantic neighbor graph. Then, a preannotation score

is obtained by graph learning of the semantic neighbor graph, considering relationships between images. The preannotation of the semantic neighbor graph can effectively solve the label imbalance problem, increasing the annotation probability of the rare labels and suppressing the high-frequency labels.

Next, the relationships between labels are used to improve the accuracy of the image annotation. The previous work was simply to calculate the cooccurrence probability between labels without considering the imbalance of cooccurrence between labels. For example, "Sea" and "Ship" are likely to appear in the same picture, and the two labels are strongly related. However, the possibility of "sea" in "ship" images may be greater than that of "ship" in "sea" images. This is because the "sea" is associated with more things, such as "fish" and "coral." To solve this imbalance of labels, the random dot product graph is used to mine the deep associations of labels. After that, visual differences that lead to lower similarity between similar images are used to further improve the performance of the proposed method. Finally, the naive Bayes nearest neighbor (NBNN) classifier is used to establish a joint likelihood between images and labels because of its simplicity and efficiency. Finally, the proposed method is conducted on Corel 5K and IAPR TC12. And results show that the proposed method has obvious improvement in terms of label recall. The main contributions of this paper are as follows:

(i) To effectively solves the label imbalance problem, the semantic neighbor graph learning is proposed to generate preannotation based on the nearest neighbor where all the labels are included in the initial label candidate

(ii) To mine the deep associations of labels, the random dot product graph is proposed, balancing the distributions of cooccurrence of paired labels

The remaining content structure of the thesis is as follows: in Section 2, we introduce the related work of image annotation. Then, in Section 3, we present our image annotation framework and concrete implementation of the framework. The datasets, experimental, settings, and results are illustrated in Section 4. The paper is concluded in Section 5.

2. Related Work

Image annotation has been a research hotspot which attracts increasing attention. Many fields are related to it, and they can benefit from the progress of each other. For example, the internet of vehicles [17, 18] can provide a lot of images to be annotated, and the better annotated images can be used to train the distinguishing model for better driving vehicles. Thus, a large number of researchers have introduced many kinds of methods to image annotation in recent years. They can be divided into four classes: the generating model, discriminating model, graph learning model, and nearest neighbor model.

2.1. Generating Model. To solve the problem in image annotation, some scholars proposed the mixture model, which is one of the generating model. For example, Jeon et al. proposed a cross-media relevance model (CMRM) [19]. In this method, image is segmented into several blobs, which can be clustered. Then, they calculate the probability between words and images by establishing maximum likelihood estimation. However, this method is affected by clustering of the image feature. Therefore, a continuous relevance model (CRM) [20] was proposed by Lavrenko et al., which used a continuous image feature. The method calculates the probability of labeling the word using polynomial distribution. But this method needs to store a large kernel matrix, resulting in a computational burden.

In order to solve the hybrid model's "visual ambiguity" problem, that visual similarities do not mean semantic similarities, researchers proposed the topic model. The topic model can be thought as a hybrid model with a particular topic used to portray the relationship between the image and the label. For example, Barnard et al. proposed a method with modeling multimodal cooccurrence [21]. This method imports several topic variables and attempts to find the relation between labels and visual features through probability. But this method is affected by model initialization. Blei et al. presented the LDA method [22], which used the Dirichlet distribution in the stage of choosing topics and words. However, the topic model is complex and has too many parameters. Thus, it is not suitable for large-scale datasets.

2.2. Discriminating Model. To solve the problem of the generating model, some researchers proposed the discriminating model. The discriminating model uses multilabel classification to solve the problem of image annotation. This method trains a classifier for each label, then determines which label the image belongs to by the classifier. For instance, Carneiro et al. proposed SML [23], which established a relationship between semantic labels and semantic classes. This method does not need to segment the image in advance, but it requires a high balance of classes and does not consider the relationship between labels. Sun et al. [24] used sparse factor representation to come up with sparse structure based on label dependency for weakening the negative effect caused by the unbalance of labels. But this method does not consider the potential relationship between images with labels and the lack of high-quality image dataset.

2.3. Graph-Based Learning Model. To address the issue of insufficient labeled images, some investigators put forward the graph learning model. The graph learning model is a semisupervised learning model, which uses labeled and unlabeled images to create the graph, then uses the Laplacian matrix for transferring labels. Liu et al. proposed the nearest spanning chain (NSC) [25]. In this method, they use a graph algorithm to transfer labels, but they do not take into account the relationship between images and labels. So Su and Xue proposed GLKNN [26]. In the stage of initializing graph weights, the cooccurrence relationship between labels is considered. However, they discount that the cooccurrence relationship is unbalance. This graph model only considers

Framework of the proposed method.
Input: images
Output: predicted labels
1: Find the best nearest neighbor images by improving the nearest neighbors.
2: Construct a similar matrix W through $W_{ij} = \exp[-\mathrm{DIS}(x_i, x_j)/\sigma^2]$.
3: Mine the deep relationship between images, using random dot product graph (RDPG) for refactoring, $P_X(G) = \prod_{i \neq j}(x_i \cdot x_j)^{a_{ij}} (1 - x_i \cdot x_j)^{1-a_{ij}}$.
4: Iterate to convergence through $R^*(t+1) = \alpha I \cdot R(t) + (1-\alpha)Y$.
5: Build a semantic matrix through $P(v_m \mid v_i) = \mathrm{sum}(m, i)/\mathrm{sum}(i)$
6: Consider the effect of the association between labels on the results of the annotations, $R'(t+1) = \alpha I \cdot R * (t) + (1-\alpha)P$.
7: Consider the relationship between images and labels, $\mathrm{Dis}(M, i) = \log(1/n)\sum_{K \in N(M.i)} K(d^M, d^K)$.
8: Return the final score of the label, $\mathrm{Score}(M, i) = \sigma R_{M,i} + \omega R^*_{M,i} + \xi \mathrm{Dis}(M, i)$.

Algorithm 1.

visual features and has no regard for problem of "visual ambiguity." Meanwhile, in the condition of a big image dataset, this model has high time complexity and poor annotation performance.

2.4. Nearest Neighbor Model. Because the nearest neighbor model performs better under big data conditions, this method has attracted more and more researchers. This model transfers the image annotation problem into the image retrieval problem. First, this method needs to search images which are highly similar to unlabeled images, then labels unlabeled images by means of label transmission. For example, Guillaumin et al. proposed a method based on weighted KNN called TagProp [27]. In this case, the labeled probability of lower frequency labels is increased and the labeled probability of higher frequency labels is suppressed. And Verma and Jawahar put forward 2PKNN [28], in which image distance metric learning was used. They adjust the weights of different visual features in order to make the relationship between visual features more consistent with the relationship between image semantics. In CCAKNN [29], the aim is to get the image subset of each semantic label. They map two features to the same subspace and model the visual features by using the Bayesian probability model. However, the nearest neighbor model only uses the similarity between images and ignores the difference between the image samples.

3. Our Approach

A new image annotation framework is proposed on the basis of graph learning, which is composed of three steps. First, we propose the nearest neighbor graph based on the principle that similar images share labels, to obtain preannotation results. Next, the association between labels is used to improve the accuracy of image annotation by the random dot product graph, which deeply mines the internal association of labels to increase probabilities of labeling weak labels. Finally, the naive Bayes nearest neighbor classifier is used to calculate the distance between images and labels. The main process of the proposed method is shown in Algorithm 1:

3.1. RPDG-Based Image Graph for Image Annotation. Let $X = \{x_1, x_2, \cdots, x_n\}$ be a collection of n images, $V = \{v_1, \cdots, v_l\}$ be a set of labels, and the training set be denoted by $T = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, which is composed of each marked image x_i and corresponding label set y_i which is presented as a binary vector. For example, if the nth image is labeled by mth label, $y_n(m) = 1$; otherwise, $y_n(m) = 0$. To solve the problem of label imbalance, the nearest neighbor graph is constructed based on the neighbor image sets.

For a given image M to be labeled, its neighbor image set $\mathrm{Nei}(M)$ constitutes a set $S(M)$. We select a set of k nearest neighbor images to form a set T for each label based on the visual distance of images. The main idea of this method is that similar images have a high probability of passing labels. The traditional approach finds the semantic nearest neighbor by using the weighted multiple vision distance, without considering the probability that two images are neighbors to each other is different. As a result, the nearest neighbors of unlabeled images would have some noise images, which brings noise labeled and decreases the image annotation accuracy. Due to the complex distribution of visual features in images, some images in the image dataset have a higher probability of being selected as neighbor images, some images are less likely to be selected, and others may not even be selected. But in practice, the nearest neighbor relationship between images is not symmetric. For example, the image M is a nearest neighbor image of the image N, but the image N is not a nearest neighbor image of the image M, which degrades the accuracy of the conventional methods in selecting nearest neighbor images. Therefore, we propose a novel way to select the nearest neighbor images.

We propose a novel method based on the common neighbor images. We use this improved method to select the nearest neighbors of the test image, reducing the noise labels. Our method first sorts images of each label according to the visual distance and selects the first $2k$ images. Then, our method selects the nearest neighbors for each of these 2 k images. Sorting according to the number of their common neighbor images, the top K images are selected as the neighbor images of image M. The nearest neighbor images selected in this way are more consistent with the image similarity

under actual conditions. And the number of images which are related to the test image in semantic is also increased. As a result, the possibility of introducing a noise image is reduced and the accuracy of the annotation improves.

We assume a simple graph $G = (V, E)$, where V is the vertex set representing images in $S(M)$. The edge set is denoted as E representing a relationship between two images. The weight W of the edge is the similarity of two images.

The principle of the graph-based learning method is semisupervised learning. This method uses the image features and annotation information of the training data. Then, it iterates the similarity matrix of the training data and passes the appropriate semantic label from the labeled images to the unlabeled images based on this similarity, which is a preliminary result of the first step.

The detail of this method is as follows:

(Step 1) Construct a similar matrix $W^{k\times k}$ of $S(M)$ set as

$$W_{ij} = \exp\left[-\frac{\mathrm{DIS}(x_i, x_j)}{\sigma^2}\right], \tag{1}$$

where DIS() is a measure of distance. And W_{ii}=0, because there is no self-loop in the graph

(Step 2) Symmetrically normalize W by

$$I = D^{1/2} W D^{1/2}, \tag{2}$$

where D is a diagonal matrix and $D_{ii} = \sum_{j=1}^{l} W_{ij}$

(Step 3) Iterate according to the Eq. (3) until convergence

$$R(t+1) = \alpha I \cdot R(t) + (1-\alpha)Y, \quad R(0) = Y, \tag{3}$$

where t is the number of iteration until convergence and α is the propagation parameter

(Step 4) Label the unlabeled images according to the convergence matrix R^*

Through the above steps, we finally get the tag score and ranking. On the basis of the above discussion, there are two key parts of the graph-based learning method: a similarity graph (I) and an initial state representation (L). I describes the similarity between the test image and its nearest neighbor images, which provides a basis for the label transmission.

Thus, the construction of a similarity graph (I) is very important. In constructing a graph in the traditional graph-based image annotation methods, the weight of the edge between the vertices (images) directly uses the visual distance. However, because of the existence of the "visual ambiguity," this method may ignore the hidden relationships of the images. So different from the previous work, we use the random dot product graph to discover hidden relationships.

The random dot product graph is a point-edge random graph model. For each node v_i, $i = 1, \cdots, n$ in the node set V, a d-dimensional vector x_i is randomly and uniformly selected from the d-dimensional unit space as the assignment of v_i. The probability of the edge between each pair of nodes v_i, v_j is

$$p_{ij} = f(x_i \cdot x_j). \tag{4}$$

This probability is used for generating a random dot product graph as the assignment $X = [x_1, x_2, \cdots, x_n]_{d\times n}$.

The two main properties of random dot product graph are as follows:

Property 1. Clustering: the edges of random dot product graph appear with incompletely equal probability, with obvious clustering characteristics.

Property 2. Transitivity: if two nodes have strong connections with the third node at the same time, then the two nodes should also have a great correlation directly. Conversely, if two nodes have no other associated third node, then the probability that the two nodes are related should be small.

Each edge in the random graph appears randomly and independently. According to the Bernoulli distribution, the random dot product graph $G_x(V, E)$ generates the edge set E according to the probability p_{ij} to obtain an observation graph. If the observation graph $G = (V, E)$ is an undirected weighted graph and its adjacency matrix is $A = (a_{ij})_{n\times n}$, $a_{ij} \in [0, 1]$, then

$$P_X(G) = \prod_{i\neq j} (x_i \cdot x_j)^{a_{ij}} (1 - x_i \cdot x_j)^{1-a_{ij}}. \tag{5}$$

Its log likelihood function is

$$L_X(G) = \sum_{i\neq j} a_{ij} \ln (x_i \cdot x_j) + (1 - a_{ij}) \ln (1 - x_i \cdot x_j). \tag{6}$$

In the observation, the probability of the edge reflects the correlation between the nodes. It can be seen from Equation (6) that when $L_X(G)$ is maximum, the probability of the edge corresponds to the weight as much as possible. According to the principle of duality, we have

$$\max L_X(G) = \min F_Z(X), \tag{7}$$

where $F_Z(X) = \sum_{i\neq j} (x_i \cdot x_j - a_{ij})^2$.

Therefore, the objective function is expressed as

$$\min F_Z(X) = \min \sum_{i\neq j} (x_i \cdot x_j - a_{ij})^2 \tag{8}$$

where $X = [x_1, x_2, \cdots, x_n]$ is a random assignment of n nodes, the probability of the edge is the inner product form, and

Random dot product method for simple graphs.
Input: the weight matrix W of the image data graph.
Output: the weight matrix of random point product.
1: Take an all-zero matrix D.
2: Find spectral decomposition of $W + \text{diag}\ (D)$.
3: U is a matrix of d largest eigenvectors, $U \in \Re^{n\times d}$. $\tilde{\Lambda}$ is a $d \times d$ diagonal matrix composed of d largest eigenvalues, where each negative eigenvalue is changed to 0.
4: $X = \sqrt{U}\tilde{\Lambda}$, $D = \text{diag}\ (X'X)$
5: Return 2 until D converges.
6: Calculate $L_X(G)$, return 1 until converges. T is the edge probability matrix after random reconstruction, where $T = XX'$.

Algorithm 2.

the right side of Equation (8) is the Frobenius norm of the matrix, so it can be written as $A \approx X^T X$.

Based on the above principles, we have the following algorithm.

Based on the above method, for given nodes i and j, the $W_{ij}{}'$ weighted distance is expressed as

$$W_{ij}{}' = W_{ij} + \omega T_{ij}. \tag{9}$$

The random dot product graph improves the weight of the similar matrix. With the improvement of the nearest neighbor graph, we pay more attention to the internal relations between images. By this method, the weak label problem can be effectively solved.

3.2. Word-Based Graph Learning. The frequencies of the labels in the image dataset are different. The low-frequency labels are easily ignored during the annotating process, which leads to the accuracy decrease of the annotation. In the previous work, people usually used the semantic symbiosis between labels to solve this problem. However, there is a cooccurrence imbalance between the labels, which makes it impossible to significantly improve the label effect of the low-frequency label. By the transitive nature of the random dot product graph, we reconstruct the association graph of the label words and find the inherent hidden relationship between the labels. The random dot product graph can obtain the relationship between any annotated words. The probability of common semantic relations is large, and the probability of uncommon semantic relations is small, which is consistent with the real semantic relationship.

In the label set $V = \{v_1, \cdots, v_l\}$, we record the probability of label v_i to label v_m denoted by $P(v_m \mid v_i)$,

$$P(v_m \mid v_i) = \begin{cases} \dfrac{\text{sum}(m,i)}{\text{sum}(i)}, & m \neq i, \\ 1, & m = i, \end{cases} \tag{10}$$

where $\text{sum}(m, i)$ represents the number of cooccurrences between labels v_i and v_m. In this paper, we abbreviate $P(v_m \mid v_i)$ to P_{im}. Because of semantic cooccurrence imbalance, P_{im} is not equal to P_{mi}.

We first get the transfer matrix between the labels according to Equation (10). P is reconstructed by random dot product to obtain P'. Bringing transfer matrix and the matrix $R*$ obtained on the basis of graph learning into Equation (3), we iterate to get result R'.

3.3. Image to Word Relation. This relationship can be regarded as the possibility of having an image to produce a label. In most cases, the relationship can be estimated on a training set by some hypothetical distribution. In many methods, the image is clustered and divided into several "blob," with each "blob" corresponding to a label word. However, in the process of clustering, problems will be caused due to that the underlying features are similar, but the actual contents are different, which makes the blob itself wrong. In this paper, the method used to calculate the image to word distance is the naive Bayes nearest neighbor (NBNN) classifier [30] for image classification. This method is simple and has good performance. At the same time, it calculates the association between the whole image and the annotated label, avoiding the wrong correspondence between the "blob" and the annotated label.

The features of the image are recorded as f, and $N(M, i)$ represents a collection of $\text{Nei}(M)$ annotated as label vi. The definition of image to word distance is

$$\text{Dis}(M, i) = \log \frac{1}{n} \sum_{k\in N(M.i)} K\left(f^M, f^k\right), \tag{11}$$

where n is the figure for images in $\text{Nei}(I, k)$. $K()$ is the Gaussian kernel function:

$$K\left(f^M, f^k\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|f^M - f^k\right\|^2\right). \tag{12}$$

3.4. Combination of Three Scores. Finally, we combine the two scores based on the graph learning with the score of the image-to-label distance to get the final score, which is the basis for the final labels.

$$\text{Score}(M, i) = \sigma R_{M,i} + \omega R^*_{M,i} + \xi \text{Dis}(M, i), \tag{13}$$

where $R'_{M,i}$ is a score based on the association between images and $R^*{}_{M,i}$ is the probability that the image M is labeled with the label v_i based on an association between labels. In addition, $\sigma + \omega + \xi = 1$.

4. Experiment

In this section, we introduce two datasets used in the experiment and the extraction of features of two datasets. Also, the evaluation indicators of the image annotation methods are given.

4.1. Datasets. During the experiment, we used two datasets. Table 1 shows the statistics of these datasets.

Corel 5K [31]. This dataset contains 4,500 training images and 499 test images. It is divided into 50 themes, each with 100 images except the last. The dataset contains 260 labels. Each image is manually labeled with 1-5 different labels, and the average is 3.4.

IAPR TC12 [32]. This dataset contains 19,627 images, where 17,665 are training images and 1962 are test images. This dataset contains a total of 291 tags, and each image in the dataset is averaged as 5.836 tags.

4.2. Feature. The first step in our approach is to extract features, which is a very important part. Feature extraction has a profound impact on the performance of image annotation systems. Recently, CNN has been widely applied to feature extraction of images. Compared with using 15 handcrafted features, it is not necessary to use metric learning to determine the optimal weight of each feature, so it is easier to determine the parameters. We use CNN to extract individual features instead of handcrafted features, which can effectively reduce the number of features and improve system accuracy.

4.3. Evaluation Metrics. In our experiments, we use the same evaluation method as [33] to effectively evaluate and compare our method with the previous methods. In our approach, we give each image five labels. Then, we calculate the labeling precision and recall for each label in each image of the test set. Suppose that a label v_k marking n_1 images in the ground truth, and the number of images marked as v_k during the test is n_2, in which the correct number of marks is recorded as n_3. The method of calculating the precision of the label v_k is $p = n_3/n_2$ and recall of the label v_k is $r = n_3/n_1$. These values are obtained by calculating each label, and then, the mean value is calculated to get the average precision P and the average recall R. Define that F1 is the score for combining P and R, $\mathrm{F1} = 2PR/(P+R)$. And define that $N+$ represents the number of tags that have been correctly tagged at least once, which indicates the ability of our proposed method to solve class imbalance and weak label problems.

5. Result

In this subsection, we describe the performance of the proposed method compared with the previously proposed methods. Table 2 gives the experimental results on the datasets Corel 5K and IAPR CT12. This table shows that our method outperforms the previous work. Among the Corel 5K, our accuracy is the second highest, and our tag recall number is the highest. Detailed results and analysis of the experiment will be presented in the following sections.

It is worth noting that we have selected several methods based on nearest neighbors as comparison methods. As shown in Table 2, our method performs better than JEC in all aspects. Compared with 2PKNN, our recall value and $N+$ value is also much higher on the Corel 5K dataset. And our RDPGKNN is superior to TagProp. The comparison with these methods shows that the graph learning method also has unique advantages in the field of image annotation and proves the validity and rationality of the label using the graph learning method for propagation.

Table 1: Details of the training set of the two datasets. The number of images and labels are given in the format mean/maximum.

Dataset	Corel 5K	IAPR TC12
# of img.	4999	19627
Vocab. size	260	291
Training img.	4500	17665
Testing img.	499	1962
Labels per img.	3.4/5	5.836/10.04
Img. per label	5.7/23	347.7/4999

We also compare RDPGKNN with graph-based learning algorithms, and the results show that our approach is generally better than previous work. Since most of the graph learning algorithms are applied to small vocabularies, there are few research methods on image annotation based on graph learning in Corel 5K and other datasets, so we mainly choose TGLM and GLKNN. In comparison with TGLM, the experimental results show that our method is obviously superior in Corel 5K. This shows the advantage of the nearest neighbor method, which effectively solves the label imbalance problem, so that each annotation word has the opportunity of being selected. At the same time, compared with GLKNN, our $N+$ has a significant improvement, because we consider label cooccurrence asymmetry. Using graph-based learning to calculate the label transition probability can maximize the selected probability of low-frequency tags, provide more appropriate weights for the transfer between tags, and improve the performance of the image tagging system.

On the IAPR CT12, our algorithm also has excellent performance. Compared with the previous work, the RDPGKNN method recalls the most labels. On this basis, our recall rate is second only to that of the CAAKNN method, and the recall rate is greatly improved on the premise that the accuracy does not drop too much. Compared to the GLKNN based on the graph, the recall rate of our method has also increased by 2%. This also confirms the need to consider the problem of cooccurrence imbalance between images. Figure 1 shows some examples of the annotation of our method on two datasets. Among them, we use the black mark to indicate the labels annotated in ground truth and annotated with RDPGKNN, and the red mark does not appear in the ground truth. It should be noted that some images in the dataset have fewer than five labels annotated in ground truth, but our method must label five labels.

After comparing with all methods, we find that our method effectively increased the value of $N+$. This shows that compared with the traditional methods, our method has strong performance in recall, and the other performance

Table 2: The performance of our proposed method is compared with the previous work on Corel 5K and IAPR TC12 datasets in detail. *P*: average precision; *R*: average recall; *F*: the combination of *P* and *R*; *N* + : number of labels with nonzero recall value.

Method	Corel 5K				IAPR TC12			
	P (%)	*R* (%)	*F* (%)	*N* +	*P* (%)	*R* (%)	*F* (%)	*N* +
CRM [20]	16	19	17	107				
MBRM [34]	24	24	24	122	24	23		223
SML [23]	23	29	26	137				
JEC [35]	27	32	29	139	29	28		250
TGLM [25]	25	29	27	131				
TagProp σSD [27]	28	35	31	145	41	30		259
TagProp ML [27]	31	37	34	146	48	25		227
TagProp σML [27]	33	42	37	160	46	35		266
KSVM-VT [33]	32	42	36	179	47	26		268
FastTag [36]	32	43	37	166	47	26		280
GLKNN [26]	36	47	41	184	41	36		282
2PKNN [28]	39	40	39	177	49	32		274
LDMKL [37]	29	44	35	179				
IDFRW [38]	38	49	43	185	49	31	38	275
CCAKNN [29]	41	43	42	185	41	40	41	278
RDPGKNN (this work)	40	45	40	195	40	38	38	283

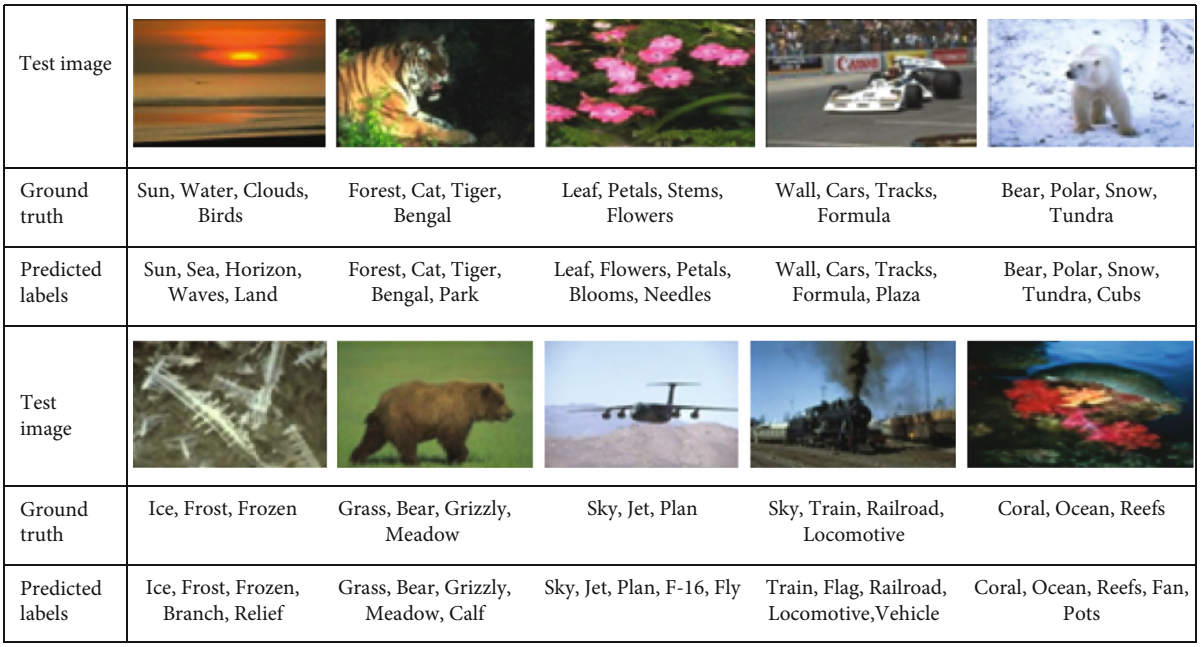


Figure 1: The result of annotating some images in the Corel 5K dataset of our method. The figure shows the test image, the ground truth labels, and the predicted labels, where red indicates that the label is not present in the ground truth labels.

is almost unchanged. Also, the problem that some labels cannot be selected due to the unbalanced label co-occurrence phenomenon is solved.

6. Conclusion

In this paper, a reconstitution graph learning model is proposed to for image annotation in smart agriculture. To solve the weak label problem, a nearest neighbor graph learning model is proposed to get the prelabels. Meanwhile, for the cooccurrence imbalance between labels, the random dot product graph is used to explore the intrinsic links between labels. Many experiments on the Corel 5K and IAPR TC12 are conducted, and the result shows that the recall of our method is much larger than that of the previous graph-based learning methods. At the same time, our accuracy and recall rate are basically the same as the latest methods. In the future, we will force on the computational complexity

of the proposed method and the depth correlation between labels and images in the annotation process.

Data Availability

The datasets used in this paper are public datasets which can be accessed by the following websites: Corel 5K: https://rdrr.io/cran/mldr.datasets/man/corel5k.html and IAPR TC12: http://www-i6.informatik.rwth-aachen.de/imageclef/resources/iaprtc12.tgz;

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by "the Fundamental Research Funds for the Central Universities", No. DUT20LAB136.

References

[1] I. Mat, M. R. M. Kassim, A. N. Harun, and I. M. Yusoff, "Smart agriculture using internet of things," in *Proceedings of the 2018 IEEE Conference on Open Systems (ICOS)*, Langkawi, Malaysia, 2018.

[2] F. Bu and X. Wang, "A smart agriculture IoT system based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 99, pp. 500–507, 2019.

[3] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[4] Z. Ning, P. Dong, X. Wang et al., "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[5] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An improved stacked auto-encoder for network traffic flow classification," *IEEE Network*, vol. 32, no. 6, pp. 22–27, 2018.

[6] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.

[7] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.

[8] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2021.

[9] J. Gao, P. Li, and Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in internet of things," *Future Generation Computer Systems*, vol. 99, pp. 508–516, 2019.

[10] B. Rani, M. Kumari, K. Sobha, P. Kumari, J. Majhi, and S. Chakraborty, "Application of Big Data in Smart Agriculture," *SSRN Electronic Journal*, 2020.

[11] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, 2020.

[12] Q. Cheng, Q. Zhang, P. Fu, C. Tu, and S. Li, "A survey and analysis on automatic image annotation," *Pattern Recognition*, vol. 79, pp. 242–259, 2018.

[13] Y. Sun and K. A. Loparo, "Context aware image annotation in active learning," 2020, http://arxiv.org/abs/2002.02775.

[14] Y. Niu, Z. Lu, J. R. Wen, T. Xiang, and S. F. Chang, "Multi-modal multi-scale deep learning for large-scale image annotation," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1720–1731, 2019.

[15] B. N. Tandel and U. Desai, "Various face annotation techniques: survey," in *Intelligent Communication Technologies and Virtual Mobile Networks*, pp. 94–102, Francis Xavier Engineering College, Tamil Nadu, Tirunelveli, India, 2019.

[16] M. Sangeetha, K. Anandakumar, and A. Bharathi, "Automatic image annotation and retrieval: a survey," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 4, pp. 1143–1147, 2016.

[17] Z. Ning, K. Zhang, X. Wang, L. Guo, and R. Y. K. Kwok, "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[18] Z. Ning, R. Y. K. Kwok, K. Zhang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[19] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03*, pp. 119–126, Toronto, Canada, 2013, ACM.

[20] V. Lavrenko, R. Manmatha, and J. Jeon, "A model for learning the semantics of pictures," *Advances in Neural Information Processing Systems*, vol. 16, pp. 553–560, 2003.

[21] K. Barnard, P. Duygulu, D. Forsyth, N. D. Freitas, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[23] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, 2007.

[24] F. Sun, J. Tang, H. Li, G. J. Qi, and T. S. Huang, "Multi-label image categorization with sparse factor representation," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1028–1037, 2014.

[25] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma, "Image annotation via graph learning," *Pattern Recognition*, vol. 42, no. 2, pp. 218–228, 2009.

[26] F. Su and L. Xue, "Graph learning on K nearest neighbours for automatic image annotation," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 403–410, Shanghai, China, 2015.

[27] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tag Prop: discriminative metric learning in nearest neighbor models for image auto-annotation," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 309–316, Kyoto, Japan, Oct. 2009.

[28] Y. Verma and C. V. Jawahar, "Image annotation using metric learning in semantic neighbourhoods," in *Proceedings of the*

12th European conference on Computer Vision, pp. 836–849, Springer, Berlin, Heidelberg, 2012.

[29] X. Wang, H. Ge, and L. Sun, "Image automatic annotation algorithm based on canonical correlation analytical subspace and k-nearest neighbor," *Journal of Ludong University(Natural Science Edition)*, vol. 32, no. 2, pp. 97–104, 2018.

[30] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, June 2008.

[31] https://rdrr.io/cran/mldr.datasets/man/corel5k.html.

[32] http://www-i6.informatik.rwth-aachen.de/imageclef/resources/iaprtc12.tgz.

[33] Y. Verma and C. V. Jawahar, "Exploring SVM for image annotation in presence of confusing labels," in *British Machine Vision Conference 2013*, Bristol, UK, 2013.

[34] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli relevance models for image and video annotation," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, vol. 2, pp. 1002–1009, Washington, DC, USA, 2004.

[35] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *Proceedings of 10th European Conference on Computer Vision*, Marseille, France, 2008.

[36] M. Chen, A. Zheng, and K. Weinberger, "Fast image tagging," in *Proceedings of the 30th International Conference on Machine Learning, PMLR*, pp. 1274–1282, Atlanta, GA, USA, 2013.

[37] M. Jiu and H. Sahbi, "Nonlinear deep kernel learning for image annotation," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1820–1832, 2017.

[38] Z. Ning, G. Zhou, Z. Chen, and Q. Li, "Integration of image feature and word relevance: toward automatic image annotation in cyber-physical-social systems," *IEEE Access*, vol. 6, pp. 44190–44198, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8816681, 9 pages
https://doi.org/10.1155/2020/8816681

Research Article

DeepCF: A Deep Feature Learning-Based Car-Following Model Using Online Ride-Hailing Trajectory Data

Yizhen Xie,[1] Qichao Ni,[2] Osama Alfarraj,[3] Haoran Gao,[2] Guojiang Shen,[4] Xiangjie Kong,[4] and Amr Tolba[3,5]

[1]*International School, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*School of Software, Dalian University of Technology, Dalian 116620, China*
[3]*Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia*
[4]*College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China*
[5]*Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-Kom 32511, Egypt*

Correspondence should be addressed to Osama Alfarraj; oalfarraj@ksu.edu.sa

Received 14 August 2020; Revised 26 September 2020; Accepted 20 November 2020; Published 7 December 2020

Academic Editor: Nathalie Mitton

The car-following model describes the microscopic behavior of the vehicle. However, the existing car-following models set the drivers' reaction time to a fixed value without considering its dynamics. In order to improve the accuracy of car-following model, this paper proposes Deep Feature Learning-based Car-Following Model (DeepCF), a car-following model based on fatigue driving and Generative Adversarial Networks (GAN). The model is composed of the drivers' reaction time model and the car-following decision algorithm. First, we regard driving fatigue as the starting point to study the influence of driving time and the acceleration of the preceding vehicle on the drivers' reaction time, and develop a coarse-grained drivers' reaction time model. Secondly, considering the impact of fatigue driving on car-following decisions, we utilize GAN to generate a driving decision database based on reaction time and use Euclidean distance as a decision search indicator. Finally, we conduct experiments on a real data set, and the results indicate that our DeepCF model is superior to baseline models.

1. Introduction

Vehicle following, the most common drivers' behavior in traffic, exerts more important influence on many factors including traffic flow characteristics, traffic safety, and traffic simulation results. The car-following model serves as a basic algorithm of traffic simulation tools (such as SUMO and VISSIM) and an indispensable control algorithm for automated vehicles [1, 2]. The model is aimed at replicating drivers' car-following behavior. The kinematics-based car-following model attempts to describe the kinematic mechanism of vehicle-following maneuver [3–12]. Most of the parameters have obvious physical meaning. The output of the model can be easily controlled by adjusting the model parameters, so as long as the appropriate parameters, it can perform better in car safety. As the car-following model based on machine learning attempts to learn the human drivers' vehicle-following motion from a large number of human drivers' vehicle-tracking data [13–17], this category of model has a high accuracy in simulating the human drivers' vehicle following.

However, as the existing car-following models have become increasingly accurate in predicting driving decisions, they overlook the dynamic time for the drivers to execute the decision [18]. They set the drivers' reaction time to a fixed value [19]. This setting will greatly affect the car-following simulation performance. For example, in the car-following case, the front car brakes suddenly, causing the rear drivers to decide to slow down [20]. If this decision is 1.3 seconds late, it is likely to collide with the car ahead. Khodayari et al. use the performance characteristics of the drivers' stimulus and reaction while driving to calculate the drivers' reaction time in NGSIM data [21]. And they add the reaction time as known information to the existing car-following

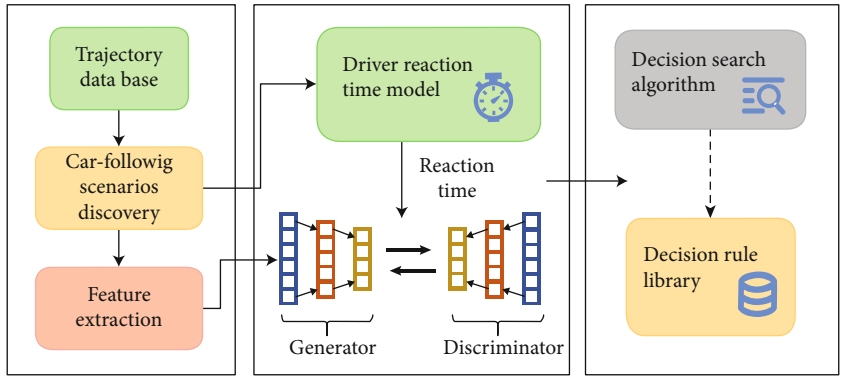


Figure 1: The framework of DeepCF model.

model for simulation experiments [22]. Finally, they confirmed that the drivers' reaction time existing in the car-following model will greatly improve the accuracy of the model simulation. However, they failed to build a model capable of calculating the drivers' reaction time. In addition, in predicting the driving behavior of the drivers, the existing car-following model does not involve two crucial factors. One is the impact of driving fatigue on driving decisions; the other is that driving fatigue causes the drivers' sensitivity and judgment to decline [23]. For example, driving for 1 hour continuously, a driver who faces the sudden acceleration of the vehicle ahead may be inclined to follow the same behaviour. However, when driving continuously for 3 h, in the face of the same scenario, the drivers' decision may be conservative acceleration, that is, the throttle will be much lighter than that before 2 h.

In this paper, we propose Deep Feature Learning-based Car-Following Model (The model frame diagram is shown in Figure 1). First, we study the influence of driving time and the acceleration of the preceding vehicle on the drivers' reaction time and establish a coarse-grained drivers' reaction time model. Secondly, grounded in the impact of driving time on car-following decisions, we generate a driving decision database based on GAN and use Euclidean distance as a decision indicator. Then, the decision search algorithm is proposed. Finally, we conduct contrast experiments on a real trajectory data, and the performance of DeepCF is evaluated.

The main contributions of this paper are listed as follows:

(1) A car-following decision algorithm based on Generative Adversarial Networks is proposed, and a method for establishing a driving decision database based on Euclidean distance as a decision index is proposed

(2) We design the framework based on the evidence that driving for a longer time will lead to a longer reaction time of the drivers

(3) Analyze the model using China's online car-hailing trajectory data

The paper is organized as follows: Section 2 introduces the related work of the car-following model. Section 3 designs the car-following model. Section 4 provides experimental results and compares them with traditional regression models. Section 5 summarizes the whole article.

2. Related Work

In this section, we first introduce the existing car-following model based on kinematics and the car-following model using machine learning algorithms. Then, we illustrate that the fatigue state will affect the drivers' reaction time.

2.1. Car-Following Model. Car-following models can be divided into kinematics-based models and models with machine learning algorithms. In the kinematics-based car-following models, Chandler et al. [3] first proposed the General Motors (GM) model. This model puts forward the relative speed of the front and rear vehicles to calculate the acceleration of the rear vehicle [4, 5]. The Gipps model takes the safety distance into account [6]. The optimal speed model obtains the expected rear vehicle speed based on the distance between the front and rear car heads [7, 8]. The action point model divides car following into different stages and sets space or speed thresholds separately [11, 12]. The car-following models based on machine learning algorithms are data-driven models. Wewerinke [13] uses neural networks to model the car-following behavior and achieves high performance. Khodayari et al. [14] propose an improved neural network car-following model with response time as input and verified it using NGSIM data [21]. The results show that the error is significantly less than other neural network models. Wei et al. [16] established a car-following model based on least squares support vector machine (LS-SVR) and used the microscopic traffic simulation system dataset [17] to verify the model. The experimental results show that LS-SVR car-following model is more accurate than the Gipps car-following model [6] and neural network car-following model.

2.2. Drivers' Reaction Time. In driving tasks, the drivers' brain nerve activity caused by exogenous stimuli is correlated with the drivers' reaction time [24], and there are many factors that can significantly affect the drivers' reaction time. Weng [23] divide the drivers' reaction behavior into three stages of perception, determination, and action. And this

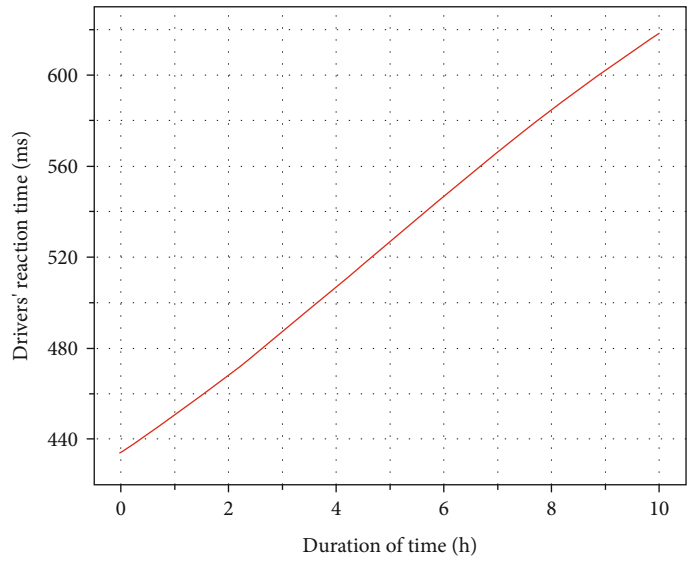


Figure 2: Drivers' reaction time function. Reaction time is positively correlated with driving duration.

paper verified that the drivers' sensitivity and judgment ability would weaken on a long road under fatigue state, and it would take longer time to make driving decisions. Visual distraction can further cause drivers to lose their attention on the road and affect their reaction time [25, 26]. Ru et al. [27] confirm that the subjective conditions that interfere with the drivers' response include driving experience, mental and physical conditions, and adaptability. Petermeijer et al. [28] explore the interaction between nondriving task types and take-over request methods. The test experiment of 101 volunteers confirm that nondriving tasks will increase the response time of the corresponding take-over request, and the initial response time of tactile and auditory take-over requests is lower than that of visual. In a traffic accident scenario, the drivers' reaction is related to the speed of obstacles before the accident [29], and the drivers' reaction time is linearly related to the collision time [30]. Xue et al. [31] analyze the simulation data of 47 volunteers in simulated car-following scenarios and found that in the case of high traffic flow density, the response time of drivers is usually shorter than that of low to medium traffic density [32, 33], while the response time of male and nonprofessional drivers tends to be slightly longer [34].

3. Methodology

This section introduces the DeepCF model in detail, including driver reaction time model and car-following decision algorithm based on GAN.

3.1. Drivers' Reaction Time Model Based on Fatigue Driving Phase Combination Model. It is intuitively obvious that the drivers' reaction time of fatigue driving is longer than that of normal driving. And in the assumption of this paper, the car-following decision is closely related to the drivers' reaction time. We need to reveal the relationship between the drivers' reaction time and driving time to build drivers' reaction time model. The reaction time model can quantitatively represent the relationship between driving time and reaction time. They investigate 294 drivers and test their reaction time [35]. The final statistical results showed a strong correlation between duration of driving and reaction time. On the basis of their experimental results, we use the cubic function to fit the relationship between duration of driving and the slowest reaction time. Our experiments show that cubic polynomials are the best choice to fit this relation. Due to the lack of original experimental data, the accuracy of the fitted function remains to be verified. The formulas are shown in Equation (1) and function visualization is shown in Figure 2.

Table 1: Training set data structure for GAN.

Attribute name	Speed difference between front and rear cars	Rear vehicle speed	Front and rear distance	Acceleration of rear car after reaction time
Unit	Km/h	Km/h	m	m/s^2

According to the results of our experiment, in Equation (1), α_3 is -0.067, α_2 is 0.9769, α_1 is 15.27 and, α_0 is 434.4. These parameters can fit the duration of time and drivers' reaction time well. And there is no overfitting in this model.

$$\tau(x) = \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0. \quad (1)$$

3.2. Car-following Decision Algorithm Model Based on Generative Adversarial Network. We need to design the drivers' car-following decision algorithm. First, we need to establish a decision library. We will use GAN (Generative Adversarial Network) to generate the decision library rules. GAN, an unsupervised deep learning model, is composed of two parts: generator and discriminator. The generator generates data close to the characteristics of the training set as much as possible, and the discriminator should attempt to determine the authenticity of the generator. It can find out the internal statistical law of the given observation data and

Input:
Training set data distribution $P_{\text{data}}(x)$; random noise distribution $P_g(z)$; Total training times epochs; the number of iterations of the discriminator k; the learning rate of the discriminator s_1; the learning rate of the generator s_2; the amount of training data per batch n.
Output:
The network parameters of the discriminator θ_d; network parameters of the generator θ_g.
Begin
1. Initialize θ_d,
2. For epochs do
3. For k do
4. Sample n samples $\{z^{(j)}\}_{j=1}^{n}$ from the random noise distribution $P_g(z)$
5. Sample n samples $\{x^{(j)}\}_{j=1}^{n}$ from the real data distribution $P_{\text{data}}(x)$
6. Update θ_d by boosting the stochastic gradient:
7. $\nabla_{\theta_d}(1/n)\sum_{j=1}^{n}[\log D(x^{(j)}) + \log(1 - D(G(z^{(j)})))]$
8. End for
9. Sample n samples $\{z^{(j)}\}_{j=1}^{n}$ from the random noise distribution $P_g(z)$
10. Update ∇_{θ_g} with gradient by decreasing:
11. $\nabla_{\theta_d}(1/n)\sum_{j=1}^{n}[\log D(x^{(j)}) + \log(1 - D(G(z^{(j)})))]$
12. End

Algorithm 1. Minibatch stochastic gradient descent training algorithm for generating adversarial networks.

can generate brand new data similar to the observation data based on the obtained probability distribution model. The target formula is shown in Equation (2).

$$\min_G \max_D V(G,D) = E_{x\sim P_{\text{data}}(x)}[\log(D(x))] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))], \tag{2}$$

where G is the differentiable function of the generator, D is the differentiable function of the discriminator, P_{data} is a real sample, x is the sample taken from P_{data}, and $D(x)$ means to distinguish x; we hope that the result of this discrimination is closer to 1, as possible the bigger the loss function $\log(D(x))$. P_z is the sample generated by G, z is the sample taken from P_G, and $D(G(z))$ means to distinguish $G(z)$; we hope that the smaller the result, as possible the bigger the loss function $\log(1 - D(G(z)))$. The structure of the training set is shown in Table 1.

The network is divided into two parts: generator and discriminator. The generator is composed of input layer, hidden layer, activation layer, and output layer. The dimension of the input and output layers is $1 * 4$. The hidden layer contains $2 * 2 * 11$ neural network units. The activation layer is composed of the Maxout activation function and sets the k value to 2. Maxout has a strong fitting ability; it can fit any convex function. Maxout has the advantages of ReLU, such as no linear saturation, but also does not have the disadvantage that the ReLU unit is fragile and may die.

Let the distribution generated by the generator (G) be $P_G(x;\theta)$, where θ is the parameter of the distribution. And let x^i be derived from the true distribution in the generator to calculate the likelihood $P_G(x^i;\theta)$ as shown in Formula (3).

$$L = \prod_{m}^{1} P_G(x^i;\theta). \tag{3}$$

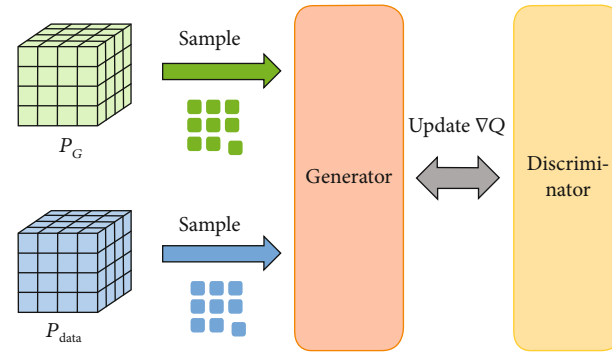


Figure 3: Structure diagram of Algorithm 1.

Then, you need to find a θ^* to maximize the likelihood, as shown in Formula (4).

$$\theta^* = \arg\min_{\theta} KL(P_{\text{data}}(x))||P_G(x;\theta). \tag{4}$$

The discriminator (D) is composed of an input layer, a hidden layer, an activation layer, and an output layer. The activation function is the sigmoid function. The discriminator's data consists of two parts, the first part is the real data set P_{data}, and the second part is the fake data P_G generated by the generator. If x comes from P_{data}, $D(x)$ should be as close to 1 as possible. If x comes from P_G, $D(x)$ should be as close to 0 as possible. The pseudocode of GAN algorithm is shown in Algorithm 1. The structure diagram of Algorithm 1 is shown in Figure 3.

The decision index is used to evaluate the similarity of the input features and the data of the decision database, that is, to calculate the similarity of the two feature vectors. There are many ways to calculate vector similarity, such as Pearson correlation coefficient, cosine similarity, and Manhattan distance. The decision index is mainly used to calculate the

Input:
Decision library D; Searched feature vector x
Output:
The fourth feature of the most similar variable y_4
Begin
1. Standardized D and x
   Set the threshold
2. Brush the database according to the feature vector x
3. Pick out the vector set V within the threshold
4. for i in V
5. use the decision index algorithm to calculate the distance between x and i, $c_1 = s(x, i)$
6. Sort the vector in V according to c_1
7. Output the fourth feature of the closest vector
End

Algorithm 2. Decision database retrieval algorithm s_su.

Table 2: Data attributes.

Field	Type of data	Example	Remarks
Driver ID	String	glox.jrrlltBMvCh8nxqktdr2dtopmlH	Data encryption processing
Order ID	String	jkkt8kxniovIFuns9qrrlvst@iqnpkwz	Data encryption processing
Timestamp	String	1501584540	Unix timestamp, in seconds
Longitude	String	104.04392	GCJ-02 coordinate system
Latitude	String	30.6863	GCJ-02 coordinate system

distance between two feature vectors, so this paper uses Euclidean distance. The index formula is shown in Formula (5). The purpose of the retrieval algorithm of this machine is to search the decision database for the decision plan (the fourth feature of Table 1) that is closest to the current driving state (the first three features of Table 1) to reach the goal of driving decision-making.

$$s(x, y) = \sqrt{\sum_{j=1}^{3} \left(x_j - y_j\right)^2}, \tag{5}$$

where y represents a feature vector in the decision library, x represents the input feature vector, and the first three features of x and y correspond to the first three features of the data structure shown in Table 1.

Due to the relatively large amount of rule data in the decision database, traversing the entire decision database to find the most similar vectors may affect the algorithm performance, so the retrieval of the decision database should be optimized. The statistics show that, aside from extreme cases, the value range of y_1 is approximately [-5,10], the value range of y_2 is approximately [0,70], and the value range of y_3 is [1,80]. Then, there are about 80,000 combinations of these three feature vectors. The following defines the retrieval algorithm $s\text{-}su$:

3.3. Deep Feature Learning-Based Car-Following Model. The driver reaction time model and the car-following decision algorithm have been obtained above, then we propose Deep Feature Learning-based Car-Following Model (DeepCF) here. Equations (6) and (7) describe the model details.

$$\tau_n = \tau(x), \tag{6}$$

$$a_x(t + \tau) = S_{su}(D, x), \tag{7}$$

where x is the feature vector.

4. Experiments

This chapter mainly introduces the experimental part, first introduces the comparative experiment, then defines the modulus evaluation index, and finally, analyzes the experimental results.

4.1. Experiment Data. The data set used in this article is the trajectory data of Didi drivers in Xi'an on October 26 and 27, 2016, provided by Didi Travel (please visit: https://outreach.didichuxing.com/research/opendata/).

The data set tracks approximately 18,000 vehicles, including the vehicle number, location, and time. The trajectory tracking time interval is 1 s. We screened the vehicles driving continuously for more than 3 h between 4:00-22:00, and finally extracted 5748 following scenes. The car-following duration is 9 s. We further add attributes such as speed, acceleration, driving duration, and distance to each piece of data. We divided the data set into two groups, the first group is the following vehicles in the scene where the driving time of the rear vehicle is less than two hours, and the second group is the vehicle that exceeds two hours.

Besides that, we select 100 follow-up scenes in each group as the test set. Data attribute list refers to Table 2.

4.2. Comparative Experiment. T. Cover and P. Hart proposed k-nearest neighbor in 1967. The working principle is there is a sample data set, and each data in the sample set has a label, that is, we know the correspondence between each data in the sample set and its classification. After inputting new data without labels, we compare each feature of the new data with the features corresponding to several types of data in the sample, and then, the algorithm extracts the classification label of the most similar data (nearest neighbor) of the sample. It uses the following Formula (8) to calculate the distance D. x and y are the two features that need to be calculated.

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{8}$$

Random forest is a model composed of many decision trees. In the training process, each tree in the random forest will learn from randomly sampled data points. The idea is to train each tree on different samples. Although for a specific training data set, the variance of each tree may be high, but in general, the variance of the entire forest will be very low without increasing the offset. In the test, predictions are made by averaging the predictions of each decision tree. This process of training a single learner on different self-sampled data subsets and averaging predictions is called bagging. The cart tree is used in the random forest algorithm in Sklearn.

The Gini index reflects the probability that two samples are randomly selected from the data set, and their category labels are inconsistent. Therefore, the smaller the Gini index, the higher the purity of the data set. The Gini index (Formula (9)) can be used to measure any uneven distribution. It is a number between 0 and 1.0 that is completely equal, and 1 is completely unequal.

$$\text{Gini}(D \mid A) = \sum_{K=1}^{K} \frac{|C_k|}{|D|}\left(1 - \frac{|C_k|}{|D|}\right), \tag{9}$$

where k represents the category.

The CART classification tree uses the size of the Gini coefficient to measure the division points of features. In the regression model, we adopt the common sum variance measurement method. For any partition feature A, the corresponding arbitrary partition point s is divided into data sets D_1 and D_2 on both sides, and the mean square error of each set of D_1 and D_2 is minimized. The feature and feature value division point corresponding to the minimum sum of mean square error of D_1 and D_2. The expression is Formula (10):

$$(y_i - c_1)\min_{a,s}\left[\min_{c_1}\sum_{x_i \in D_1}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in D_2}(y_i - c_2)^2\right]. \tag{10}$$

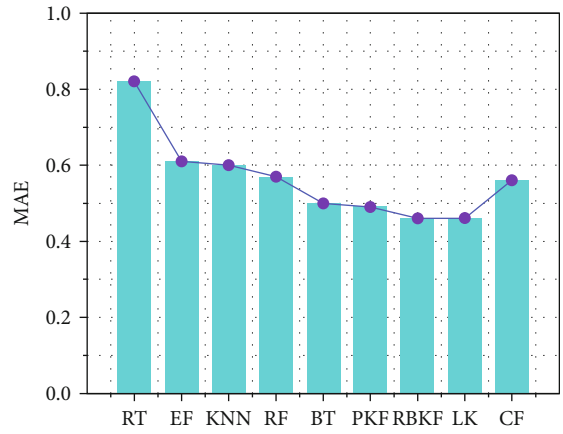


Figure 4: Comparison of CF and regression model with MAE as an evaluation index.

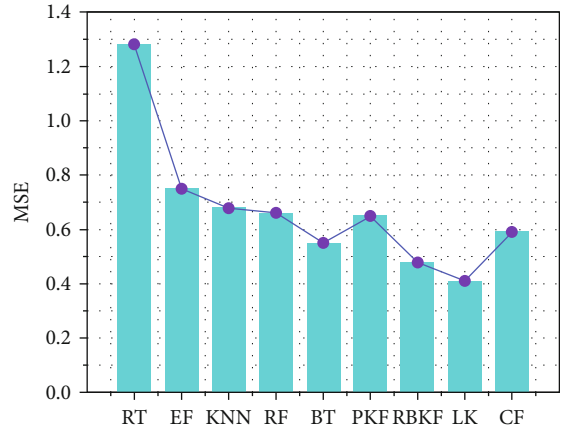


Figure 5: Comparison of CF and regression model with MSE as an evaluation index.

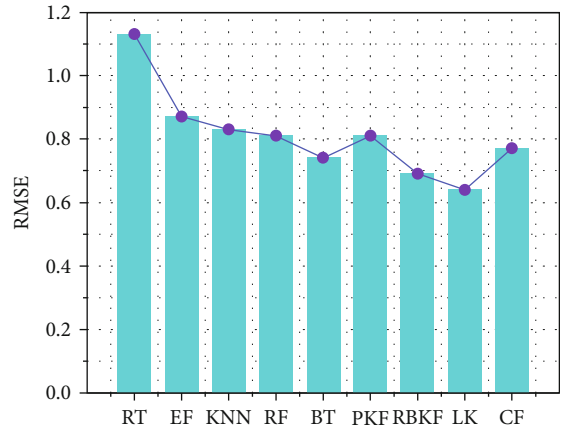


Figure 6: Comparison of CF and regression model with RMSE as an evaluation index.

Among them, c_1 is the sample output average of the D_1 data set, and c_2 is the sample output average of the D_2 data set.

4.3. Model Evaluation. In this paper, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) are used for measurement and evaluation [36]. The indicator formulas are shown in Equations (11), (12), and (13).

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|(x_i - \widehat{x}_i)|, \tag{11}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x\wedge_i)^2, \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x\wedge_i)^2}, \tag{13}$$

where N is the total number of samples, x_i represents real data value of car i, and $\widehat{x}_i$ means the predicted value of the model.

5. Results and Discussion

This paper proposes a car-following model based on driving fatigue and generative confrontation network, namely, dynamic car-following model (DeepCF). We use 474 real car-following scenes for experiments, and the experiments combine the DeepCF model proposed in this article with the regression model regression tree (RT), polynomial kernel function (PKF), radial basis kernel function (RBKF), boost tree (BT), extreme forest (EF), linear kernel (LK), k-nearest neighbor (KNN), and random forest (RF) for comparison. We use MAE, MSE, and RMSE as evaluation indicators. We use 474 real car-following scenes as the test set.

First, we use all real 8000 car-following scenes as a decision library. Besides, we adopt the decision database retrieval algorithm *s¯su*; the acceleration of the following vehicle after the reaction time is matched according to the speed of the following vehicle, the distance between the front and rear vehicles, and the speed difference between the front and rear vehicles in the test set. And we calculate the evaluation index of the comparison between the real car-following scene and the matching result (for the convenience of subsequent experiments, we will name the experiment CF here). Then, we use the above regression model to obtain the acceleration of the following vehicle after the reaction time. It also calculates the evaluation index comparing the real car-following scene with the regression result. Finally, we compare the evaluation index of the matching result with the evaluation index of the regression result. The results are shown in Figures 4, 5, and 6.

According to Figures 3, 4, and 5, we can find that when using a limited number of real car-following scenes as the decision-making database, using the decision-making database retrieval algorithms, the matching results are better than RT, EF, KNN, and other regression models. However, compared with RBKF, LK, and other regression models, there is still a certain gap.

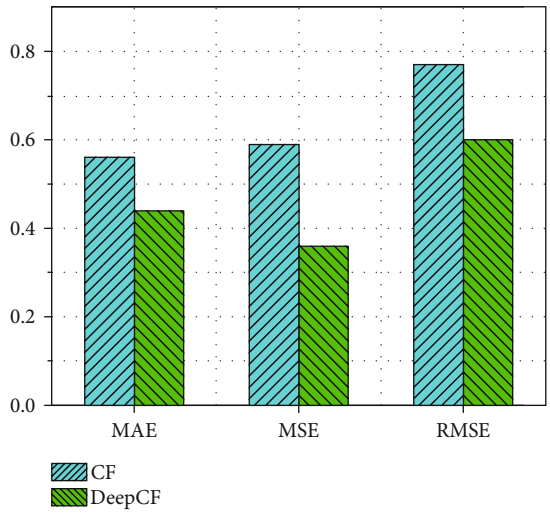


Figure 7: DeepCF compared with CF.

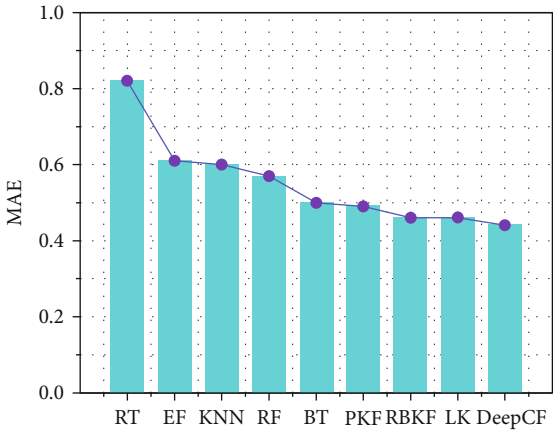


Figure 8: Comparison of DeepCF and regression model with MAE as an evaluation index.

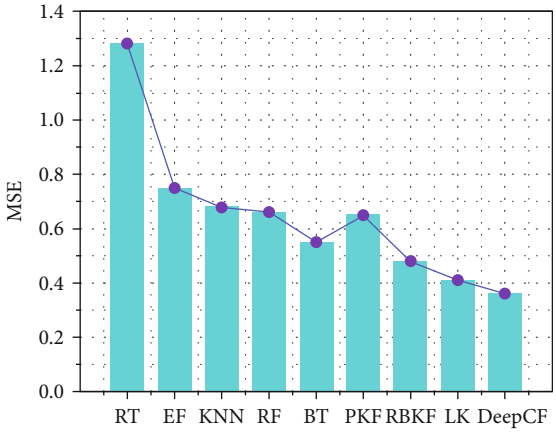


Figure 9: Comparison of DeepCF and regression model with MSE as an evaluation index.

We use 8000 real car-following scenes as raw data to use the GAN to generate 80,000 car-following scenes, and combine the real and generated car-following scenes as a decision

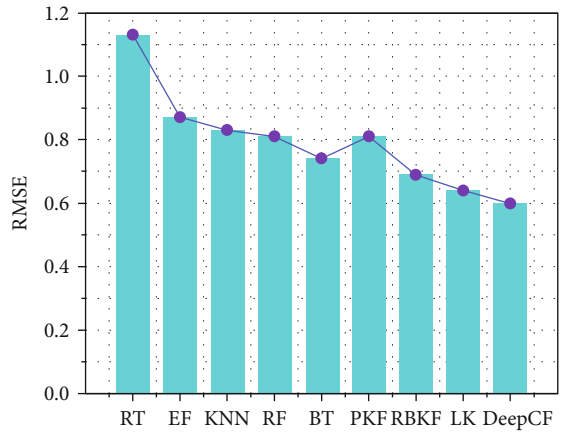


Figure 10: Comparison of DeepCF and regression model with RMSE as an evaluation index.

library. Using the decision database retrieval algorithm $s^{-}su$, the acceleration of the following vehicle after the reaction time is matched according to the speed of the following vehicle, the distance between the front and rear vehicles, and the speed difference between the front and rear vehicles in the test set. And we calculate the evaluation index of the comparison between the real car-following scene and the matching result (our model DeepCF). We compare the evaluation indicators obtained after the DeepCF model experiment with the above CF model Figure 7.

We find that after using the GAN to expand the decision-making database, the matching results are significantly better than before data generation. This proves the necessity of using GAN to generate car-following scenes.

We compare the evaluation indicators obtained after the DeepCF model experiment with the regression model, and the results are shown in Figures 8, 9, and 10.

In the premise of using the confrontation generation network to generate 80,000 car-following scenes with 8000 real car-following scenes as raw data, combining the real and generated car-following scenes as a decision-making database and adopting the decision-making database retrieval algorithms, the matching data performs better than most data in the regression model. Thus, the effectiveness of our DeepCF model is proved.

6. Conclusions

In view of the fact that existing car-following models fail to consider the impact of driving fatigue on driver reaction time and decision-making, this paper proposes a car-following model (DeepCF) based on driving fatigue and generating a confrontation network. A car-following decision algorithm based on a generative confrontation network is proposed, and we build a driving decision database with GAN. Besides a comparative experiment, we further conduct a comparative evaluation of the model. The results demonstrate that our car-following model (DeepCF) is closer to the real scene than the regression model.

Data Availability

The data used to support the findings of this study were supplied by Didi Travel under license and so cannot be made freely available. Requests for access to these data should be made to the corresponding author.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the Researchers Supporting Project No. (RSP-2020/102) King Saud University, Riyadh, Saudi Arabia, and by the National Natural Science Foundation of China (Grant nos. 62072409 and 62073295). The author would like to thank Mengyuan Wang and Zhen Ren from the School of Software, Dalian University of Technology, for their help.

References

[1] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2018.

[2] L. Li and W. Ma, "A collision-free car-following model for connected automated vehicles," *Transportation Research Board 96th Annual Meeting*, pp. 1–19, 2017.

[3] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (May, 2018)," *IEEE Access*, vol. 6, pp. 51756–51765, 2018.

[4] X. Kong, X. Liu, B. Jedari, and M. Li, "Mobile crowdsourcing in smart cities : technologies, applications and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.

[5] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, pp. 1–16, 2020.

[6] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: studies in car following," *Operations Research*, vol. 6, no. 2, pp. 165–184, 1958.

[7] R. Herman, E. W. Montroll, R. B. Potts, and R. W. Rothery, "Traffic dynamics: analysis of stability in car following," *Operations Research*, vol. 7, no. 1, pp. 86–106, 1959.

[8] D. C. Gazis, R. Herman, and R. B. Potts, "Car-following theory of steady-state traffic flow," *Operations Research*, vol. 7, no. 4, pp. 499–505, 1959.

[9] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[10] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical review E*, vol. 51, no. 2, pp. 1035–1042, 1995.

[11] R. Jiang, Q. Wu, and Z. Zhu, "Full velocity difference model for a car-following theory," *Physical Review E*, vol. 64, no. 1, article 017101, 2001.

[12] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.

[13] L. Evans and R. Rothery, "Perceptual thresholds in car-following—a comparison of recent measurements with earlier results," *Transportation Science*, vol. 11, no. 1, pp. 60–72, 1977.

[14] P. Wewerinke, "Modeling human learning involved in car driving," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1968–1973, San Antonio, TX, USA, USA, 1994.

[15] A. Khodayari, A. Ghaffari, R. Kazemi, and R. Braunstingl, "A modified car-following model based on a neural network model of the human driver effects," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 6, pp. 1440–1449, 2012.

[16] D. Wei, F. Chen, and T. Zhang, "Least square-support vector regression based car-following model with sparse sample selection," in *2010 8th World Congress on Intelligent Control and Automation*, pp. 1701–1707, Jinan, China, 2010.

[17] D. Wei, F. Chen, and X. Sun, "An improved road network partition algorithm for parallel microscopic traffic simulation," in *2010 International Conference on Mechanic Automation and Control Engineering*, pp. 2777–2782, Wuhan, China, 2010.

[18] X. Kong, F. Xia, J. Li, M. Hou, M. Li, and Y. Xiang, "A shared bus profiling scheme for smart cities based on heterogeneous mobile crowdsourced data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1436–1444, 2020.

[19] Z. Ning, P. Dong, X. Wang et al., "When deep reinforcement learning meets 5G-enabled vehicular networks: a distributed offloading framework for traffic big data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1352–1361, 2019.

[20] A. Khodayari, R. Kazemi, A. Ghaffari, and R. Braunstingl, "Design of an improved fuzzy logic based model for prediction of car following behavior," in *2011 IEEE International Conference on Mechatronics*, pp. 200–205, Istanbul, Turkey, 2011.

[21] M. Weng, "A study on evaluation of automobile driver fatigue performance," in *2010 International Conference on Machine Vision and Human-machine Interface*, pp. 92–94, Kaifeng, China, 2010.

[22] Z. Ning, K. Zhang, X. Wang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[23] X. Kong, J. Cao, H. Wu, and C. Hsu, "Mobile crowdsourcing and pervasive computing for smart cities," *Pervasive and Mobile Computing*, vol. 61, p. 101114, 2020.

[24] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[25] H. Zhang, R. Chavarriaga, L. Gheorghe, and J. D. R. Millán, "Brain correlates of lane changing reaction time in simulated driving," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3158–3163, Kowloon, China, 2015.

[26] Y. Liang and J. D. Lee, "Combining cognitive and visual distraction: less than the sum of its parts," *Accident Analysis & Prevention*, vol. 42, no. 3, pp. 881–890, 2010.

[27] G. K. Kountouriotis, P. Spyridakos, O. M. Carsten, and N. Merat, "Identifying cognitive distraction using steering wheel reversal rates," *Accident Analysis & Prevention*, vol. 96, pp. 39–45, 2016.

[28] G. Ruhai, Z. Weiwei, and W. Zhong, "Research on the driver reaction time of safety distance model on highway based on fuzzy mathematics," in *2010 International Conference on Optoelectronics and Image Processing*, vol. 2, pp. 293–296, Haikou, China, 2010.

[29] S. Petermeijer, F. Doubek, and J. de Winter, "Driver response times to auditory, visual, and tactile take-over requests: a simulator study with 101 participants," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1505–1510, Banff, AB, Canada, 2017.

[30] M. Sieber and B. Färber, "Driver perception and reaction in collision avoidance: implications for adas development and testing," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 239–245, Gothenburg, Sweden, 2016.

[31] R. S. Jurecki and T. L. Stańczyk, "Analyzing driver response times for pedestrian intrusions in crash-imminent situations," in *2018 XI International Science-Technical Conference Automotive Safety*, pp. 1–7, Casta, Slovakia, 2018.

[32] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[33] X. Kong, S. Tong, H. Gao et al., "Mobile edge cooperation optimization for wearable Internet of things: a network representation-based framework," *IEEE Transactions on Industrial Informatics*, 2020.

[34] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, pp. 411–425, 2021.

[35] P. Philip, J. Taillard, M. Quera-Salva, B. Bioulac, and T. Åkerstedt, "Simple reaction time, duration of driving and sleep deprivation in young versus old automobile drivers," *Journal of Sleep Research*, vol. 8, no. 1, pp. 9–14, 2002.

[36] A. Khodayari, A. Ghaffari, R. Kazemi, and N. Manavizadeh, "Anfis based modeling and prediction car following behavior in real traffic flow based on instantaneous reaction delay," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 599–604, Funchal, Portugal, 2010.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8869385, 19 pages
https://doi.org/10.1155/2020/8869385

Research Article

Big Data Aspect-Based Opinion Mining Using the SLDA and HME-LDA Models

Ling Yuan,[1] **JiaLi Bin**,[1] **YinZhen Wei**,[2] **Fei Huang**,[3] **XiaoFei Hu**,[3] **and Min Tan**[3]

[1]*School of Computer Science, Huazhong University of Science and Technology, 430074, China*
[2]*Huanggang Normal University, 438000, China*
[3]*Wuhan Fiberhome Technical Services Co., Ltd, 430205, China*

Correspondence should be addressed to YinZhen Wei; wyz_gs@163.com

Received 20 July 2020; Revised 1 September 2020; Accepted 23 October 2020; Published 19 November 2020

Academic Editor: Amr Tolba

In order to make better use of massive network comment data for decision-making support of customers and merchants in the big data era, this paper proposes two unsupervised optimized LDA (Latent Dirichlet Allocation) models, namely, SLDA (SentiWordNet WordNet-Latent Dirichlet Allocation) and HME-LDA (Hierarchical Clustering MaxEnt-Latent Dirichlet Allocation), for aspect-based opinion mining. One scheme of each of two optimized models, which both use seed words as topic words and construct the inverted index, is designed to enhance the readability of experiment results. Meanwhile, based on the LDA topic model, we introduce new indicator variables to refine the classification of topics and try to classify the opinion target words and the sentiment opinion words by two different schemes. For better classification effect, the similarity between words and seed words is calculated in two ways to offset the fixed parameters in the standard LDA. In addition, based on the SemEval2016ABSA data set and the Yelp data set, we design comparative experiments with training sets of different sizes and different seed words, which prove that the SLDA and the HME-LDA have better performance on the accuracy, recall value, and harmonic value with unannotated training sets.

1. Introduction

With the development of the Internet, almost all the things of human living have become digitized. The information in network is in the form of structured data and unstructured data [1]. Big data analysis and mining is aimed at discovering implicit, previously unknown, and potentially useful information and knowledge from big databases that contain high volumes of valuable veracious data collected or generated at a high velocity from a wide variety of data sources [2], which is called "4V" of big data. In fact, the deeper mining of big data is to mine the user demand and other deep information; the text mining that this paper studied is one of the ways to mine valid information from big data of text. Therefore, the study proposes two optimized opinion mining methods for customers and merchants to extract valid information they need from massive textual data that satisfies the "4V" [2].

For example, when purchasing a product, people usually refer to others' comments in the specialized product comment area first. Although comments on different platforms have different forms of display, most of them are text-based. Due to most product comments on Taobao only have positive, neutral, and negative labels, what users can directly refer to is just the number of positive and negative comments. Since different users have different needs for the same product, they need to know which attributes of it perform well and which perform poorly. However, these attributes are not shown in detail in the comment interface. It is impossible and uneconomic for users to read more comments to find the attribute they need, because of the massive quantity and continuous growth of product comments. Thus, in order to assist clients and merchants for better decision-making, conducting opinion mining and sentiment analysis of big data is necessary, which will bring huge profits to some markets.

Data mining and analysis have been used in the tourism industry [3], groundwater potential mapping [4], and so on; it becomes more and more important in modern life. Liu and Zhang [5] divided the text mining and analysis tasks into

three levels of granularity: chapter level, sentence level, and attribute level. The chapter level, whose research unit is a document, usually uses algorithms to show whether the opinions expressed by the author are negative or positive, and it is often used to analyse blogs and news. While the sentence-level sentiment analysis is often used to analyse whether the expression of microblogs, tweets, etc., in social networks is negative or positive. Obviously, it is impossible to perform opinion mining at the chapter level and sentence level when analysing user comments. Please consider the following review:

The location of this restaurant is relatively remote, but the waiter has a good attitude and the dishes taste good.

In fact, this sentence contains three opinions. Therefore, we ought not to determine the sentiment polarity of this comment simply and should get more specific results, such as <location, remote, negative>, <service, good, positive>, <dish, good, positive>, i.e., the aspect-based opinion mining. In 2015 and 2016, SEMEVAL released the research topic of ABSA (Aspect-Based Sentiment Analysis), triggering a study boom among scholars. SEMEVAL divides ABSA into three subtasks: identifying entities and attributes, identifying the expression of opinions, and identifying the sentiment polarity of opinions. SEMEVAL also gives the annotated training sets and test sets, such as catering, hotel, and laptop.

In recent years, people devote a lot of energy to analyse comments on the Internet for obtaining the detailed opinions of users [6, 7]. The aspect-based opinion mining of online comments can be divided into two subtasks, namely, the opinion target extraction and the classification of sentiment polarity of comments. Figure 1 gives the overview of the study method classification of the aspect-based opinion mining.

With the development of supervised models, the deep learning, and the conditional random field, CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Networks) have been widely used in NLP (Natural Language Processing) [8], which have achieved good results in ABSA research. Some other supervised learning methods [9, 10] such as the BMAM [11] which was proposed in recent time also have achieved good results. In addition, Chen et al. [12] and Araque et al. [13] introduced deep learning into the ABSA and achieved satisfactory results. However, compared with the models proposed in this paper based on LDA, all the above models lack the adaptability in various fields and have high manpower costs for annotation. For example, the effects of these models will be greatly reduced when the aspect category of the comment is transferred from the *food* and *beverage* to the *laptop*. Moreover, supervised models such as BMAM [11] need a lot more manpower than the models proposed in this paper to annotate data due to the small number of annotated training sets given. In addition, the time complexity of SLDA and HME-LDA is better than the popular models above. The time complexity of SLDA and HME-LDA is just related to the number of documents, topics, and words, while the time complexity of the popular models such as RNN is related to the multilayer neural network structure, especially the fully connected layer, and the result of the previous time sequence, leading to a really lower speed than the SLDA and HME-LDA.

As for the unsupervised model, there are two basic models for latent semantic analysis: the probabilistic latent semantic analysis (PLSA) [14] model and the latent Dirichlet allocation (LDA) [15] model, which can be applied to extract attributes, assuming that each comment is a combination of attribute words and opinion. Mei et al. [16] proposed a topic-sentiment hybrid model based on the PLSA to extract aspect opinion target words and sentiment words from a group of blogs. Li et al. [17] proposed two kinds of joint models, sentiment LDA and dependence-sentiment LDA, to find positive or negative aspects of sentiment words. Due to the flexibility of the LDA topic model, it is extended and combined with other methods to obtain a topic model [18, 19], which can improve the result topics or the additional information of the model.

In view of the short content, wide coverage and the small number of the annotated corpus of the network comment and its need for aspect-based mining, this paper proposes two schemes based on the LDA topic model that have unsupervised features and good extensibility, making it possible for network comments to perform aspect-based opinion mining with as little annotated data as possible. As for the data cleaning, the amount of data, correctness, completeness, and time correlation [20] are all good evaluation indicators of data quality. As for the data amendment, the Markov Random Field performs well [21].

This paper regards opinion targets, aspects, and opinion expressions as aspect opinion targets that refer to entities or properties to which sentiment words modifies, and sentiment words related to aspect opinion targets are called sentiment opinion words or opinion words.

Moreover, the aspect-based opinion mining schemes proposed in this paper only require users to set corresponding seed words and introduce the classification layer to classify the opinion target words and sentiment opinion words. Meanwhile, in order to improve the effects of the models in schemes, we are biasing the parameters of the LDA model by calculating the similarity between the words in the corpus and the seed words we set.

To sum up, the main study contents of this paper are as follows:

(1) Introduce the NLP tools, WordNet and SentiWordNet, into the standard LDA model to design an optimized LDA-based topic model

(2) Introduce a maximum entropy classifier into the standard LDA model to design another optimized LDA-based topic model

(3) Implement the above two optimized models and design experiments to verify the feasibility and superiority of optimized models

The rest of the paper is organized as follows. Next, we will describe two schemes with two optimized LDA-based models in detail. Then, the experimental results will be given in the next section. At the same time, we will give some analysis about the results. Finally, we give the conclusion.

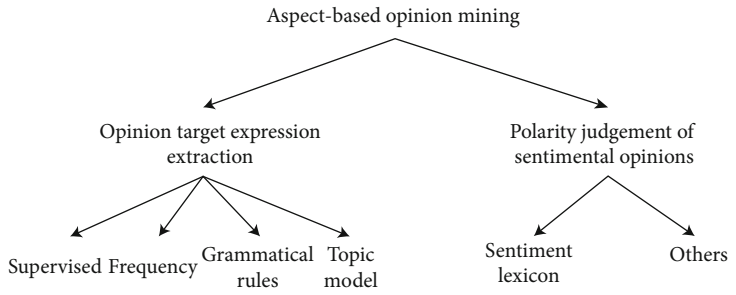


Figure 1: The classification of aspect-based opinion mining.

2. Two Optimized LDA Models for Aspect-Based Opinion Mining

This paper proposes two schemes for aspect-based opinion mining. The first scheme is based on the inverted list and the SLDA (SentiWordNet WordNet-Latent Dirichlet Allocation) model proposed in this paper. The second scheme is based on the inverted list and the HME-LDA (Hierarchical Clustering MaxEnt-Latent Dirichlet Allocation) model proposed in this paper.

The SLDA model is an optimized LDA model based on the WordNet and SentiWordNet, where the WordNet is for the similarity calculation of words and seed words and the SentiWordNet is for the separation of the opinion target words and the opinion words, while the HME-LDA model is an optimized LDA model based on SLDA and MaxEnt-LDA [22]. In fact, SLDA still has the disadvantage of relying on dictionary tools (WordNet and SentiWordNet), leading to the application failure of SLDA in other languages. Luckily, the HME-LDA model solves the problem.

Next, we will illustrate the two optimized models of schemes in detail.

2.1. Optimized Model, SLDA, Based on SentiWordNet and WordNet. Different from the text messages such as documents, blogs, and news, network comments tend to be shorter and often appear in the form of sentences. Please consider the following restaurant comments:

(1) "We, there were four of us, arrived at noon - the place was empty and the staff acted like we were imposing on them and they were very rude."

(2) "Everything is always cooked to perfection, the service is excellent, the decor cool and understated."

In nonaspect opinion mining, the only thing you need to do is to analyse the sentiment polarity of sentences. For example, the word "rude" in the first sentence is negative, so the sentiment polarity of the first sentence is negative. In the second one, the sentiment polarity of the word "perfection" is positive, so the sentiment polarity of the second sentence is positive. This way that only judges the sentiment polarity of sentences does not apply to network comments. More meaningful information should be specific to the word pairs of <Aspect Opinion Target-Opinion> such as <staff-rude>, <cook-perfect>, and <service-excellent>, while the algorithm based on the topic model lacks readability and appointed key words, where the relevant original content fails to be directly found by the final result.

The SLDA is an optimized LDA model based on SentiWordNet (a sentiment dictionary based on WordNet) and WordNet (a large database of English words). Since it is impossible for the LDA itself to separate opinions from opinion targets, this chapter adds a classification layer of opinion words and opinion target words based on the LDA to realize the separation of opinions and aspect opinion targets. The similarity between the word and the seed word in the text, which is reflected on LDA parameters, is calculated by setting the seed word and using the calculation tools of the vocabulary similarity in WordNet. Meanwhile, the opinion target is separated from the sentiment opinion words using tools that calculate the vocabulary sentiment in SentiWordNet. Aiming at the lack of readability of LDA results, we establish a belonging relationship among the clustering results, seed words, and original texts. Also, the SLDA model needs to set seed words, but has no need for the additional annotated data sets.

In order to achieve the goal that users can quickly find what they want from massive comments by inquiring the index rather than by reading, there are three steps to do:

(1) Construct an inverted index to number sentences and words

(2) Determine the aspect category of comments, separate the aspect-based attribute words from the sentiment opinion words, and determine the aspect category to which they belong

(3) Enhance the readability of results. Users can see an overview of Domain-Aspect-Opinion and find the specific sentence by the inverted index of a word

It is worth noting that the premise of our study is that the aspect category of the original corpus is known. The aspect category, which has the belonging relationship with the aspect opinion target, is usually given in advance by the original corpus, which can be learned by the user guide of the corpus. For example, the aspect opinion target "steak" belongs to the aspect category "Food." In addition, both words and sentences can belong to an aspect category. Thus, what only need to do is to label aspect opinion target words and sentences with a specific aspect category.

2.1.1. Implementation Process. Figure 2 is an overall flow chart of the scheme that conducts aspect-based opinion mining based on SLDA and the inverted index.

(1) Construct an inverted index. The words in the corpus are numbered in the form of binary group $< a, b >$, where a is the serial number of the sentence and b is the serial number of the words in the sentence. In addition, the generation of the inverted list requires the removal of duplicate word and the recording of their numbers. The inverted list reserves the sentence number that contains the word and the position information of the word in the sentence, making it easier to retrieve with context information from the original corpus later

(2) Data preprocessing, whose main tasks are to extract the data required and remove stop words from the original corpus for making a training set. The formats of the original data sets are XML and CSV. It is necessary to extract comment statements by the corresponding labels and fields. And the text in the corpus contains many useless stop words, such as "is" and "a," which should be removed before further processing to avoid interference with the training of the SLDA model

(3) Introduce preprocessed data into SLDA for model training and get clustering results. The setup of seed words, as well as the assist of WordNet and SentiWordNet, is required in this process

(4) Process the clustering results for better readability. The results in the topic model are probability matrix, whose readability is poor. To solve this problem, it is necessary to find the word corresponding to the result with higher probability and find its original sentence by the inverted index

In short, the SLDA model, trained with the preprocessed data of step 2 above, queries the original sentences that contain keywords from the original corpus by the inverted index of step 1 above.

2.1.2. The Optimized Direction of LDA. The expectation of every random variable μ_i of the Dirichlet distribution can be expressed as $E(\mu_i)$. The value of $E(\mu_i)$ can be calculated by equation (1), where α represents the parameter of the Dirichlet distribution and K represents the number of topics:

$$E(\mu_i) = \frac{\alpha_i}{\sum_{i=1}^{K} \alpha_i}, \quad (1)$$

where α is a fixed value and the $E(\mu_i)$ of each topic is same. Based on this, a biasing method of α can be explored to make the expectations of the corresponding probabilities of each topic different, so as to generate the topic bias. More visually, when α is fixed, it means that we fail to know which topic word to use before generating the document. When α is biased, the ideal topic is determined before the document is generated.

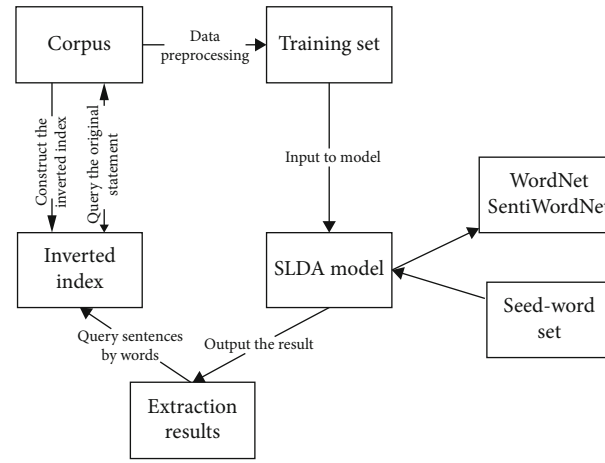


Figure 2: The overall flow chart of the scheme that is based on SLDA and inverted index.

Actually, the topic number $Z_{m,n}$ can be used as an indicator variable in the LDA model to control the selection of topic-word distribution. Based on this, more indicator variables can be introduced to refine the topic and obtain more topic-word distributions.

From the above, there are two aspect opinion targets that the LDA model can optimize: the first is to bias the parameters, α and β, which can generate topic biases to improve the classification effect; the second is to introduce more indicator variables like $Z_{m,n}$, which can generate more detailed topic classifications.

2.1.3. The Description of the SLDA Model. The standard LDA uses the document as the unit of topic allocation, while in the aspect-based opinion mining of the network comments, the sentence is often used as the unit of topic allocation, because there is no document with large contents in network comments and the topic allocation of vocabulary in network comments is actually more meaningful. In the aspect-based opinion mining of network comments, it is necessary to extract the aspect category of the comment, the opinion target, and the comment opinion (sentiment polarity) from the text. For example, in restaurant comments, "food," "service," and "ambiance" are the aspect categories of comments. In the "food" category, "steak" is the opinion target, and the evaluation of "good" for "steak" is the opinion of the comments (sentiment polarity).

In the SLDA model, seed words are directly used as aspect categories of comments, while the opinion targets of comments and the comment opinions are separated by the introduced sentiment layer, and the positive and negative polarities of comments are classified as well. The PGM (Probabilistic Graphical Model) of SLDA is shown in Figure 3.

2.1.4. The Generation Process of the SLDA Model. Only the parameters α and β, which, respectively, belong to the document-topic Dirichlet distribution and the topic-word

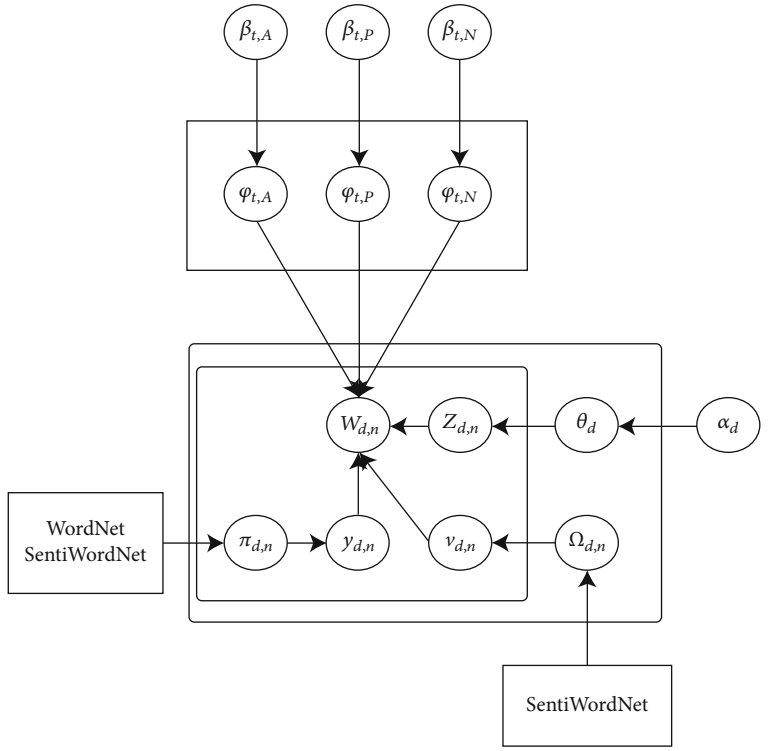


Figure 3: The probabilistic graphical model of SLDA.

Dirichlet distribution, are introduced into the standard LDA. In the SLDA model, seed words have been identified as the aspect category of comments. The variable $y_d \in \{A, O\}$ is introduced to represent the separation of the opinion target and the comment opinion. When $y_d = A$, the current word is the opinion target of comments. When $y_d = O$, the current word is the comment opinion. Meanwhile, the variable $v_d \in \{P, N\}$ is introduced into SLDA. When $v_d = P$, the sentiment polarity of the current vocabulary is positive, and when $v_d = N$, the sentiment polarity of the current vocabulary is negative. Both y_d and v_d are determined by the corresponding algorithms based on the WordNet and SentiWordNet. In the standard LDA model, the values of parameters, α and β, are fixed. In the SLDA model, different α_d will be set for each sentence, and the parameters, $\beta_{t,A}$, $\beta_{t,P}$, and $\beta_{t,N}$, will be set for the opinion target, the positive comments and the negative comments, respectively.

In SLDA, the first step is to generate the sentence-topic distribution and determine a topic for each sentence by the multinomial distribution. Then, two influence factors, y_d and v_d, are determined by WordNet and SentiWordNet to indicate the aspect categories of words. By the SLDA generation method above, we finally select a topic-word distribution and determine the final word.

The concrete generation process of SLDA is as follows:

Firstly, the distributions of the opinion target $\varphi_{t,A}$, the positive opinion word $\varphi_{t,P}$, and the negative opinion word $\varphi_{t,N}$ are, respectively, extracted from the parameters, $\beta_{t,A}$, $\beta_{t,P}$, $\beta_{t,N}$, where $\varphi_{t,A} \sim \mathrm{Dir}(\beta_{t,A})$, $\varphi_{t,P} \sim \mathrm{Dir}(\beta_{t,P})$, and $\varphi_{t,N} \sim \mathrm{Dir}(\beta_{t,N})$.

Similar to the standard LDA, for each sentence (document in the standard LDA), a topic distribution, $\theta_d \sim \mathrm{Dir}(\alpha_d)$, is extracted from the Dirichlet distribution with the parameter α_d.

Then, the word $w_{d,n}$ in the sentence d extracts a topic number t, i.e., extracts $z_{d,n} \sim \mathrm{Multi}(\theta_d)$.

After determining the topic number of the word $w(d, n)$, the classification of it remains to be determined. After determining the topic t in SLDA, in order to further classify the word as the opinion target word or the opinion of comments, the variables, $y_{d,n} \in \{A, O\}$ and $v_{d,n} \in P, N$, are introduced into the SLDA model. The $y_{d,n}$ and $v_{d,n}$ point to the n-th word $w_{d,n}$ in the sentence d jointly. The $y_{d,n}$ is used to indicate that the word $w_{d,n}$ is the opinion target of comments or the comment opinion, which is extracted from the Bernoulli distribution on $\{0, 1\}$ with the parameter $\pi_{d,n}$. The $v_{d,n}$ is used to indicate that the word $w_{d,n}$ is a positive or a negative comment opinion, which is extracted from the Bernoulli distribution on $\{0, 1\}$ with the parameter $\Omega_{d,n}$. The above two Bernoulli distributions are calculated by the WordNet and SentiWordNet. Finally, the word $w_{d,n}$ can be extracted according to equation (2):

$$Wd, n \sim \begin{cases} \mathrm{Multi}(\varphi t, A), & \text{if } yd, n = A, \\ \mathrm{Multi}(\varphi t, A), & \text{if } yd, n = O, \text{and } vd, n = P, \\ \mathrm{Multi}(\varphi t, A), & \text{if } yd, n = O, \text{and } vd, n = N. \end{cases} \tag{2}$$

Table 1 gives the description of the related symbols in the SLDA model, which is useful for readers in reading.

2.1.5. The Inference Process of the SLDA Model. The SLDA model consists of two major parts. One is the classification of the opinion target words and sentiment opinion words composed of the WordNet and SentiWordNet as well as the classification of positive and negative sentiment opinion words. The other is the LDA topic model. Besides, the seed words should be set before the inference of SLDA. Next, several modules will be introduced in turn.

(1) The Setting of Seed Words. In SLDA, seed words are set as aspect categories of comments. For example, in the classic English comment set of the Restaurant, the aspect categories are *food*, *service*, *ambiance*, etc. If the corpus is the Restaurant English comment set, the seed words can be directly set as *food*, *service*, and *ambience*. The seed word is recorded as w_t, where $t \in \{1, \cdots, T\}$, that is to say, the number of seed words determines the number of topics in the SLDA model.

(2) The Inference of the Word Classification Model Based on the WordNet and SentiWordNet. In the SLDA model, the Bernoulli distribution with parameter $\pi_{d,n}$ and Bernoulli distribution with parameter $\Omega_{d,n}$, which are, respectively, used to separate the opinion target words from sentiment opinion words and separate positive sentiment opinion words from negative sentiment opinion words, are related to the WordNet and SentiWordNet. The calculation of $\pi_{d,n}$ depends on the seed word w_t.

The words in WordNet have the feature of polysemy. To calculate the similarity between words, it is necessary to determine the exact meaning of a word. Therefore, before the model inference, it is necessary to determine the semantic interpretation $s_{t,k0}$ of the seed word w_t in the WordNet. When the current word is $w_{d,n}$, its semantic interpretation in the WordNet is $s_{d,n,k}$, where $k \in \{1, \cdots, K\}$ and K is the number of semantic interpretations. After determining the semantics of the seed words, we can regard the seed word as the topic and the aspect category of the final opinion target. All nonsentiment opinion words will be grouped into a certain aspect category of comments. Therefore, the similarity between the semantics of the current word and the semantics of the seed word can be calculated, and the semantics with the greatest similarity can be determined as the meaning expressed by the word in the sentence finally. The degree of semantic similarity between different words in the WordNet is recorded as $\mathrm{Sim}(s_1, s_2)$; then, the semantic similarity, $\mathrm{Sim}(s_{d,n,k}, s_{t,k0})$, between the $s_{d,n,k}$ of $w_{d,n}$ and the $s_{t,k0}$ of each seed word w_t can be calculated, where $k \in \{1, \cdots, K\}$. Besides, K is the number of semantic interpretations, $t \in \{1, \cdots, T\}$, and T is the number of seed words. In all calculation results, we choose the semantics with the max value of the similarity result and determine the largest result k' as the semantics $s_{d,n,k'}$ to which the current word $w_{d,n}$ belongs.

After determining the semantic $s_{d,n,k'}$ of the word $w_{d,n}$, the sentiment polarity of $s_{d,n,k'}$ can be queried in the SentiWordNet. In the SentiWordNet, the semantics of a word has three sentiment propensity probabilities: ρ_o indicates the probability that the semantics is objective (excluding sentiment polarity), p_P indicates the probability that the semantics is positive, and ρ_N indicates the probability that the semantics is negative. Besides, $p_o + p_P + p_N = 1$. It is believed that if the semantics is objective, the word is the opinion target, whose corresponding probability is p_o; otherwise, it is a sentiment opinion word. The sentiment scores of the semantics $s_{d,n,k'}$ of the word $w_{d,n}$ are recorded as $p^o_{d,n,k}$, $p^P_{d,n,k}$, and $p^N_{d,n,k}$. In Section 2.1.4, the Bernoulli distributions with parameter $\pi_{d,n}$ and parameter $\Omega_{d,n}$ can be determined by equation (3) and equation (4):

$$\pi_{d,n} = p^o_{d,n,k}, \tag{3}$$

$$\Omega_{d,n} = \frac{p^P_{d,n,k}}{p^P_{d,n,k} + p^N_{d,n,k}}. \tag{4}$$

So far, the model inference based on the WordNet and SentiWordNet has been completed. Algorithm 1 is the pseudocode for calculating $\pi_{d,n}$, $\Omega_{d,n}$.

(3) The Inference of the SLDA Model. In the standard LDA, the parameters, α and β, of the Dirichlet distribution are fixed. While in the SLDA model, the parameters, α and β, are biased by calculating the similarity between the input corpus and seed words. The fixed parameters are recorded as α_{base} and β_{base}, and the biased parameters are recorded as α_d, $\beta_{t,A}$, $\beta_{t,P}$, and $\beta_{t,N}$. In this paragraph, the semantic similarity between the word w and the seed word t is recorded as $\mathrm{sim}(w, t)$. The probability that w is a positive word is recorded as $\mathrm{sim}(w, P)$. The probability that w is a negative word is recorded as $\mathrm{sim}(w, N)$. The w is recorded as $\mathrm{sim}(w, A)$ if it belongs to the opinion target word. The w is recorded as $\mathrm{sim}(w, O)$ if it belongs to the sentiment opinion word.

In the standard LDA, the parameter α is used to control the topic distribution probability of the document. For all documents in the corpus, the values of α are the same, leading to determine which topic word will be the document topic before generating the document hardly. However, the parameter α_d in SLDA, which can be calculated by equation (5), will be set separately for each document (for each sentence in SLDA) based on the similarity between vocabulary and seed words, leading to determine a more ideal topic in advance before generating the document.

$$\alpha_d = \frac{\sum_i^{N_d} \mathrm{sim}(w_{d,i}, t)}{\sum_{t'}^{T} \sum_i^{N_d} \mathrm{sim}\left(w_{d,i}, t'\right)} \times \alpha_{\text{base}}. \tag{5}$$

In equation (5), N_d is the number of all words in the current sentence, T is the number of topics, $w_{d,i}$ is the i-th word in the current sentence, and t is the seed word.

In the standard LDA, the parameter β is used to control the word distribution of each topic, and the value of β is the same for each topic. In SLDA, parameters which can be

Table 1: The description of related symbols in the SLDA model.

Symbol	Description
D	The total number of comments in the corpus. The unit of corpus is the sentence
T	The number of topics
V	The number of words in the corpus
$w_{d,n}$	The n-th word in the d-th comment in the corpus
$z_{d,n}$	The topic of d-th comment. The value is $\{1, \cdots, T\}$
$y_{d,n}$	An indicator variable. The value is $\{A, O\}$. It is used to indicate the opinion target words and the sentiment opinion words.
$v_{d,n}$	An indicator variable. The value is $\{P, N\}$. It is used to indicate positive and negative sentiment opinion words.
A, O, P, N	The opinion target, the sentiment opinion word, the positive sentiment word, the negative sentiment word
$\varphi_{t,A}$	The distribution of the opinion target generated by a priori Dirichlet distribution with the parameter $\beta_{t,A}$
$\varphi_{t,P}$	The distribution of positive comment words generated by a priori Dirichlet distribution with parameters $\beta_{t,P}$
$\varphi_{t,N}$	The distribution of negative comment words generated by a priori Dirichlet distribution with parameters $\beta_{t,N}$
θ_d	The distribution of sentence topic terms generated by a priori Dirichlet distribution with parameter α_d

1: Query the semantic list S_{list} of $w_{d,n}$ in WordNet
2: //*Record the semantic value of the max similarity*
3: $s_w = 0$;
4: //*Record the maximum similarity*
5: $sim_{\max} = 0$;
6: //*Each semantic s of* $w_{d,n}$
7: **for** $s \in s_{list}$ **do**
8: **for** $s_t \in s_{t_{list}}$ **do**
9: //*Sim() is a function provided by WordNet*
10: **if** $\text{sim}_{\max} < \text{Sim}(s, s_t)$ **then**
11: $s_w = s$;
12: $sim\_\max = Sim(s, s_t)$;
13; **end if**
14: **end for**
15: **end for**
16: Use SentiWordNet to query the sentiment polarity of semantic S_w
17: Calculate $\pi_{d,n}$ using equation (3)
18: Calculate $\Omega_{d,n}$ using equation (4)
19: **return** s_w
20: **return** $\pi_{d,n}$
21: **return** $\Omega_{d,n}$

Algorithm 1. The calculation of $\pi_{d,n}$, $\Omega_{d,n}$.

calculated by equation (6), equation (7), and equation (8) will be set separately for each topic based on the similarity between vocabulary and seed words.

$$\beta_{t,A} = \text{sim}(w, A) \times \beta_{\text{base}}, \tag{6}$$

$$\beta_{t,P} = \text{sim}(w, P) \times \beta_{\text{base}}, \tag{7}$$

$$\beta_{t,N} = \text{sim}(w, N) \times \beta_{\text{base}}. \tag{8}$$

Similar to the standard LDA, SLDA is solved by the Gibbs sampling method. The variables involved in the solution process are explained in Table 2.

Equation (9) is used to sample the topic of each sentence.

$$p\left(Z_{d,n} = t \middle| z_{\neg d,n}, y_{\neg d,n}, v_{\neg d,n}, \cdot\right) \propto \frac{n_{w_{d,n}}^{t,A} + \beta_{w_{d,n}}^{t,A}}{\sum_v^V \left(n_v^{t,A} + \beta_v^{t,A}\right)} \times \frac{n_{w_{d,n}}^{t,P} + \beta_{w_{d,n}}^{t,P}}{\sum_v^V \left(n_v^{t,P} + \beta_v^{t,P}\right)} \times \frac{n_{w_{d,n}}^{t,N} + \beta_{w_{d,n}}^{t,N}}{\sum_v^V \left(n_v^{t,N} + \beta_v^{t,N}\right)} \times \left(n_{d,t} + \alpha_{d,t}\right). \tag{9}$$

Table 2: The partial symbolic description of SLDA model inference.

Symbol	Description
$n_v^{t,A}$	The number of words v in topic t and category A
$n_{d,t}$	The number of words with the topic t in the d-th sentence
$\beta_v^{t,u}$	The number of words v in topic t and category A

Equation (10) and equation (11) are used to sample $y_{d,n}$ and $v_{d,n}$.

$$p\left(y_{d,n}=u\middle|z_{d,n}=t,\cdot\right)\alpha\frac{\left(n_{w_{d,n}}^{t,u}+\beta_{w_{d,n}}^{t,u}\right)\times\mathrm{sim}(w_{d,n},u)}{\sum_v^V\left(n_v^{t,u}+\beta_v^{t,u}\right)},\ u\in\{A,O\},\tag{10}$$

$$p\left(v_{d,n}=q\middle|z_{d,n}=t,\cdot\right)\propto\frac{\left(n_{w_{d,n}}^{t,q}+\beta_{w_{d,n}}^{t,q}\right)\times\mathrm{sim}(w_{d,n},q)}{\sum_v^V\left(n_v^{t,q}+\beta_v^{t,q}\right)},\ q\in\{P,N\}.\tag{11}$$

In the corpus, the approximate probability of the topic t and the sentence d can be calculated by equation (12).

$$\theta_d=\frac{n_{d,t}+\alpha_{d,t}}{n_d+\sum_{t'}^T\alpha_{d,t'}}.\tag{12}$$

With t as the topic, the approximate probability that the word $w_{d,n}$ is the opinion target can be calculated by equation (13), the approximate probability that the word $w_{d,n}$ is the positive opinion can be calculated by equation (14), and the approximate probability that the word $w_{d,n}$ is the negative opinion can be calculated by equation (15).

$$\varphi_{w_{d,n}}^{t,A}=\frac{n_{w_{d,n}}^{t,A}+\beta_{w_{d,n}}^{t,A}}{\sum_v^V\left(n_v^{t,A}+\beta_v^{t,A}\right)},\tag{13}$$

$$\varphi_{w_{d,n}}^{t,P}=\frac{n_{w_{d,n}}^{t,P}+\beta_{w_{d,n}}^{t,P}}{\sum_v^V\left(n_v^{t,P}+\beta_v^{t,P}\right)},\tag{14}$$

$$\varphi_{w_{d,n}}^{t,N}=\frac{n_{w_{d,n}}^{t,N}+\beta_{w_{d,n}}^{t,N}}{\sum_v^V\left(n_v^{t,N}+\beta_v^{t,N}\right)}.\tag{15}$$

(4) Gibbs Sampling Implementation of the SLDA Model. The Gibbs sampling process of SLDA mainly has the following steps:

(1) *Random initialization*: randomly assign a topic number z to all sentences in the corpus, the topic numbers of all words in the sentence are also set to z, then the values of the indicator variables, y and u, are randomly set for all words in the sentence, where $y\in\{A,O\}$, $u\in\{P,N\}$.

(2) Traverse the corpus again, resample the topics of all words according to equation (9), and update the relevant values. Then, resample the indicator variables, y and u, according to equation (10) and equation (11), and update their values

(3) Repeat step 2 until the Gibbs sampling results converge

(4) Process the final results to improve the readability and save the results

In the initialization phase of the Gibbs sampling, a topic number is randomly assigned to each sentence in the corpus. For each word in the sentence, three categories are randomly assigned, which are determined by the indicator variables, y and u. Then, add 1 to the corresponding statistics. A part of the pseudocode in the initialization phase is shown in Algorithm 2.

In the repeated iteration phase of Gibbs sampling, the topic of each sentence in the corpus and the category of each word in the sentence are resampled. Then, update the relevant statistics after each sampling. Repeat the process until the end of the iteration.

In the result processing stage, the relevant value φ can be calculated by equation (13), equation (14), and equation (15). The calculated φ value is a digital, and the variables required in the calculation are the topic number, the category to which the word belongs, and the number of the word in the vocabulary. The relevant sentence information is missing; accordingly, it is necessary to effectively organize various types of information for the user to view. We use *result* to represent the final result. The *result.topic* is the topic information, that is, the seed word set by ourselves, which also can be regarded as the comment category word. The *result.word* saves the original word. The *result.wordType* is the category of the current word (aspect target words, positive and negative opinion words). The *result.sentences* is the sentence to which the word belongs. The *result.prob* is the probability that the word becomes a member of the category which the comment belongs to. The first m results are generated for each category under all the topics, and the relevant pseudocode is shown in Algorithm 3.

After getting the finalResults, we can query the results according to both the topic and the wordType.

2.2. Optimized Model, HME-LDA, Based on MAXENT-LDA. Because the WordNet and SentiWordNet only support English, SLDA has no linguistic adaptation. Therefore, we propose an optimized model in this chapter, namely, the HME-LDA model that has the linguistic adaptation. The

```
1: for d = 1 to D do
2:     //Randomly assign topic t to sentences in Corpus
3:     t = randomint from (1, T);
4:     documentTopics [d] = t;
5:     documentTopicsCount [d][t]++;
6:     for w = 1 to N do
7:       //Randomly assign value to y
8:       y = randomint from (0,1);
9:       Y[d][w] = y;
10:      //Randomly assign value to u
11:      u = randomint from (0,1);
12:      U[d][w] = u;
13:      if y ==0 then
14:        //statistic of aspect targets with topic t plus 1
15:        aspectWordCount [w][t]++;
16:      end if
17:      if y ==1 and u ==0 then
18:        //statistic of positive opinion with topic t plus 1
19:        positiveWordCount [w][t]++;
20:      end if
21:      if y ==1 and u ==1 then
22:        //statistic of negative opinion with topic t plus 1
23:        negativeWordCount [w][t]++;
24:      end if
25:    end for
26: end for
```

Algorithm 2. The initialization of Gibbs sampling.

optimized model, HME-LDA, is proposed by combining with the MaxEnt-LDA [22] model and the SLDA model. And the HME-LDA uses an unsupervised hierarchical clustering method to generate the annotated data set required by the maximum entropy model. Based on these, the new model can be used for comment opinion mining in many other languages.

2.2.1. Implementation Process. The overall process of the scheme that conducts aspect-based opinion mining based on the HME-LDA and the inverted index is shown in Figure 4:

(1) This step is the same as the implementation of step 1 in Section 2.1.1

(2) Data preprocessing, whose main task is to remove stop words. The text in the corpus contains many useless stop words, such as "is" and "a," which should be removed before further processing to avoid interference with the results

(3) Automatically generate annotated data sets by hierarchical clustering and train maximum entropy models for classification of the opinion target words and sentiment opinion words

(4) Enter the data into the HME-LDA model and perform the training to get the results

(5) Process the results for better readability. The results in the topic model are probability matrixes, whose readability is poor. To solve this problem, it is necessary to find the word which corresponds to the result that has a higher probability and find its original sentence by the inverted index

As for Step 3 above, there are a lot of word order information in the corpus; thus, when the category of a word in the corpus can be determined, the feature $\{w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}\}$ can be extracted. In HME-LDA, seed words are divided into topic seed words of aspect categories, seed words expressing positive sentiment, and seed words expressing negative sentiment. By scanning corpus, annotated feature sets, $\{w_{i-2}, w_{i-1}, w_i^u, w_{i+1}, w_{i+2}\}$, can be obtained, where $u \in \{A, O\}$. When $u = A$, w_i^A is the topic seed words, and when $u = O$, w_i^O is the sentiment seed words, and u is regarded as the label. However, the number of seed words is limited, so the training set obtained by scanning corpus may be insufficient in size. Therefore, the words in the corpus are considered to be clustered, and all the words in the category of the seed words are considered to have the same category as the seed words, and the word w_i^u in the scanned annotated feature set $\{w_{i-2}, w_{i-1}, w_i^u, w_{i+1}, w_{i+2}\}$ is replaced with the word in the category of seed words to get the new annotated data.

2.2.2. MaxEnt-LDA Model. In order to realize the aspect-based opinion mining of reviews, Zhao et al. [22] proposed a MaxEnt-LDA model based on LDA. Figure 5 is the probability model diagram of MaxEnt-LDA [22].

Adopting the second thought of optimizing LDA in Section 2.1.2, MaxEnt-LDA further divides topics and increases the topic number by introducing another two indicator variables, $y_{d,s,n}$ and $u_{d,s,n}$, that are similar to $Z_{m,n}$. Three new topic-word distributions are generated by the Dirichlet distribution with the parameter β. Meanwhile, the original topic is further divided into categories A (aspect-term) and O (opinion word).

In the MaxEnt-LDA model, the indicator variable $y_{d,s,n}$ is generated by the sampling of the maximum entropy model. In this model, the selected features of the maximum entropy model include the word and the part of speech, while there are three labels which are background word B, opinion word O, and the opinion target word A. The annotated training set is partially extracted from the SemEval data set and partially annotated by manual. The indicator variable $y_{d,s,n}$ is the same as $Z_{m,n}$, both of which are sampled from the multinomial distribution generated by the Dirichlet distribution.

MaxEnt-LDA increases both the type and number of classification and introduces the maximum entropy model. However, the MaxEnt-LDA model increases the dependence on the annotated data simultaneously. In order to pursue the unsupervised features of the model, there are two ways to improve the MaxEnt-LDA: one is to replace the maximum entropy model with other unsupervised classification models; the other is to use unsupervised methods to automatically label data sets to avoid the dependence on annotated data. Meanwhile, the parameters α and β can be considered for bias.

2.2.3. The Description of the HME-LDA Model. In the HME-LDA model, seed words are directly set as an aspect category

```
1: //Create a 3d array to store results
2: init results[][][]
3: for topic in TopicList do
4:   for wordType in {A, P, N} do
5:     for v in V do
6:       result = new result
7:       result.topic = topic
8:       result.word = v
9:       result.wordType = wordType
10:        Calculate φ according to Equation. (13), Equation. (14), Equation. (15)
11:        result.prob = φ
12:        //Add result to the array
13:        results[topic][wordType][v] = result
14:    end for
15:  end for
16: end for
17: //Next, sort the results
18: for topic in TopicList do
19:  /Store sentences contain results
20:  finalResults[][][]
21:  for wordType in {A, P, N} do
22:    //Sort by φ value and get the results of the top m
23:    finalResults[topic][wordType] = getTopKByPhi(results[topic][wordType], m)
24:    fork = 1 to m do
25:      Query the corresponding sentence of result.v from the inverted index described in Section 3.1.1 and add it to the result
26:    end for
27:  end for
28: end for
29: return finalResults
```

Algorithm 3. The result processing of the Gibbs sampling.

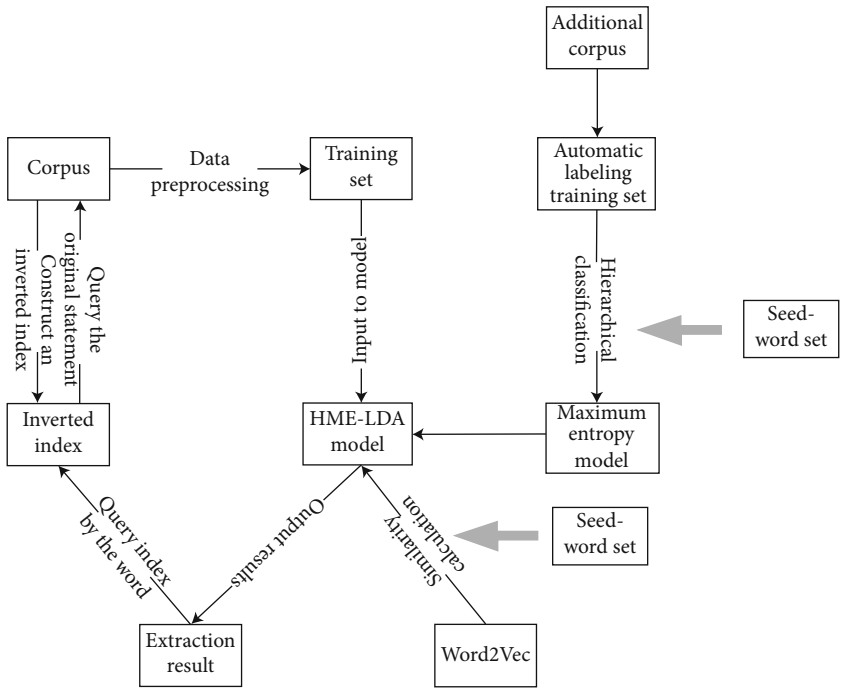


Figure 4: The flow chart of the scheme that is based on the HME-LDA and inverted index.

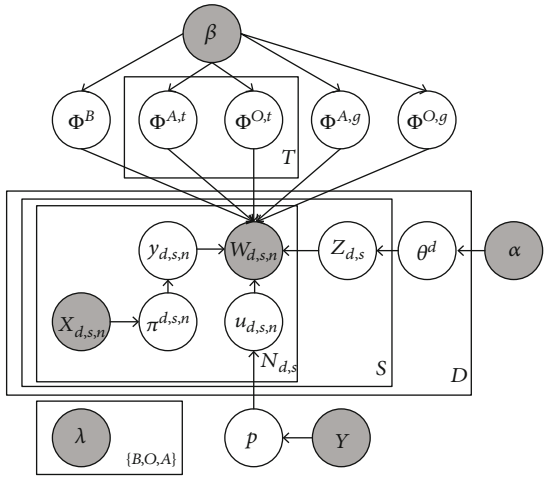


Figure 5: The probability diagram of MaxEnt-LDA.

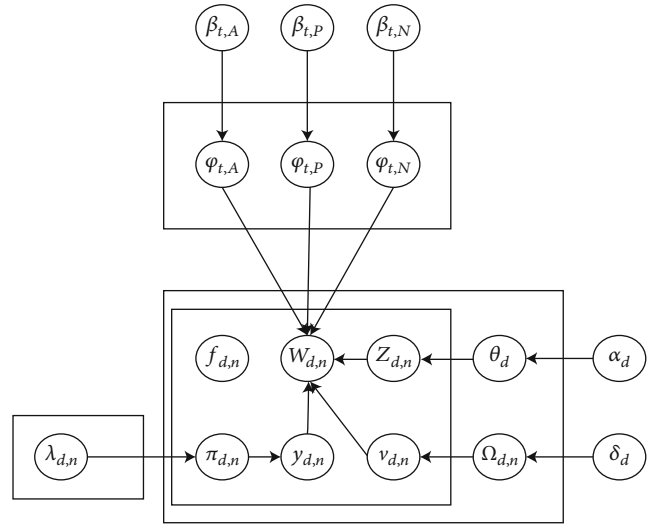


Figure 6: The probabilistic graphical model of HME-LDA.

of reviews. Therefore, it only needs to classify the opinion targets of reviews and the review opinions by introducing the maximum entropy model as a classifier. The Bernoulli distribution with the parameter $\pi_{d,n}$, where d indicates the d-th sentence and n indicates the n-th word, is jointly determined by the weight $\lambda_{d,n}$ and the eigenvector $f_{d,n}$ of the maximum entropy model. A beta distribution with a parameter δ_d will be introduced as a priori to generate a Bernoulli distribution with a parameter Ω_d for the classification of positive and negative sentiment opinion words. Figure 6 is the PGM.

The generation process of the HME-LDA model is similar to that of the SLDA model in Section 2.1. The difference is that $y_{d,n}$ is determined by the maximum entropy model rather than by the WordNet and SentiWordNet. And $v_{d,n}$ is determined by the parameter δ_d rather than by the calculation of the WordNet and SentiWordNet. The generation process of the HME-LDA model can refer to Section 2.1.4.

In addition, because there are no methods to calculate the sentiment polarity in HME-LDA, it is necessary to set a sentiment opinion word whose sentiment polarity is positive or negative for each comment category.

2.2.4. The Automatic Data Annotation Method Based on Hierarchical Clustering. The MaxEnt-LDA [22] model uses the word feature with a window whose size is 3 to extract the features from the annotated words. The selected features include the word and the word order $\{w_{i-1}, w_i, w_{i+1}\}$, where w_i is the current word. The selected features also include the features of grammatical rules of words $\{POS_{i-1}, POS_i, POS_{i+1}\}$, where POS_i indicates the part of speech of the current word (adjectives, nouns, verbs, etc.). The part-of-speech tagging requires the use of additional tools, while it is different for the accuracy of part-of-speech tagging due to different kinds of languages, and there is a possibility that the tagging tool is lacking. Therefore, the features selected in this chapter are only the words themselves.

(1) The Process of Automatically Annotating Data. After identifying the feature information obtained from the training, the next step is to consider how to automatically label it. In order to extract the feature $\{w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}\}$, the only thing that needs to do here is to determine the category of the word w_i in the corpus with information of word order. The seed-word setting of HEM-LDA is explained in Section 2.2.3. In the HME-LDA, seed words can be divided into three categories: topic seed words of a review category, seed words expressing positive sentiments, and seed words expressing negative sentiments. And the opinion target words belong to the corresponding category of comments, both of which are the same kind of topic. Therefore, seed words can be divided into two categories: opinion target words and sentiment opinion words. The categories of these seed words are able to be determined. The annotated feature sets $\{w_{i-2}, w_{i-1}, w_i^u, w_{i+1}, w_{i+2}\}$ can be obtained by scanning the corpus, where $u \in \{A, O\}$. When $u = A$, w_i^A is the topic seed word. When $u = O$, w_i^O is the sentiment seed word, and u is the label.

However, the number of seed words is limited, and the size of the training set obtained by scanning corpus may be insufficient. Therefore, we attempt to cluster the words in the corpus and treat that all words in the category of seed words have the same category as the seed words. Besides, the word in this category is used to replace the word w_i^u in the scanned annotated feature set $\{w_{i-2}, w_{i-1}, w_i^u, w_{i+1}, w_{i+2}\}$, so as to obtain the new annotated data. The pseudocode is shown in Algorithm 4.

(2) The Selection of Clustering Method. In this section, in order to improve the domain adaptability of the model and avoid the different effects of the same value of K when using the data of different fields, and to avoid more parameter adjustments as well, we select the hierarchical clustering method which has no need for the number of clusters. The result of hierarchical clustering is shown in Figure 7.

In the tree structure generated by the hierarchical clustering results, the leaf nodes of the tree are words in the corpus.

```
1: for t in Topics do        //Process each seed word
2:   wordList = getWrodListFromCorpus(t)    //Find the location where t appears from the corpus and get the corresponding word order
3:   wordCluster = getWordCluster(t)    //Get all the words of the category t from the clustering results
4:   trainSet = new Set    //Used to save labelled training samples
5:   for wOrder in wordList do
6:     for w in wordCluster do
7:       replaceWord(wOrder, t, w)    //Replace the word t in the word order with w
8:       trainSet.add(wOrder, t.Type)    //Add the label to which wOrder and t belong to the training set
9:     end for
10:  end for
11: end for
```

Algorithm 4. The process of automatically labelling data.

When looking for words of the same category, the intermediate node of the upper layer can be found by the current word. All the leaf nodes of the subtree with this intermediate node as the root node belong to the same category.

2.2.5. The Inference of the HEM-LDA Model. The reasoning of the HME-LDA model mainly includes two parts. The first part is the inference of the maximum entropy model for the classification of the opinion target words and sentiment opinion words. And the second part is the inference of the optimized LDA model.

(1) The Maximum Entropy Model. The maximum entropy model solves the classification problem actually. When the input of the model is x, the probability $p(y \mid x)$ of the category y can be calculated by equation (16).

$$p(y \mid x) = \frac{e^{\sum_{i=1}^{n} \lambda_i f_i(x,y)}}{\sum_y e^{\sum_{i=1}^{n} \lambda_i f_i(x,y)}}. \tag{16}$$

In equation (16), λ_i represents the weight vector, $f_i(x, y)$ is the eigenfunction, and n is the number of categories. When using the maximum entropy model for the classification of the opinion target words and sentiment opinion words, it is necessary to select appropriate features. The MaxEnt-LDA [22] model uses two features, i.e., word order and part of speech. The part of speech tagging relies on some other tools that is different with different languages. In order to avoid using tools that rely on languages, this chapter chooses the word order as a feature. Section 2.2.4 gives the method of automatically annotating the training set.

By training the maximum entropy model, the weight λ_u of the feature set $f_{d,n}$ can be obtained, and $\pi_{d,n}^{u}$ can be obtained by equation (17), where d represents the d-th sentence, n is the n-th word in the sentence, $u \in \{0, 1\}$ is the label collection, 0 represents the opinion target whose type is A, and 1 represents the sentiment opinion whose type is O.

$$p\left(y_{d,n} \mid f_{d,n}\right) = \pi_{d,n}^{u} = \frac{e^{\lambda_u \times f_{d,n}}}{\sum_{i=0}^{1} e^{\lambda_i \times f_{d,n}}}. \tag{17}$$

(2) The Model Inference of HME-LDA. In SLDA, the parameters, α and β, are offset by calculating the similarity between the input corpus and the seed words. The fixed parameters are denoted as α_{base} and β_{base}, and the offset parameters are denoted as α_d, $\beta_{t,A}$, $\beta_{t,P}$, and $\beta_{t,N}$. In Chapter 2, the above parameters are offset by using the semantic similarity calculation of the WordNet and the sentiment polarity calculation of the SentiWordNet. In this section, both the Word2Vec model and the cosine distance are used to calculate the similarity between words. The training corpus of the Word2Vec model is the all content in the corpus. The vector of the current word in the Word2Vec is represented by v_w, and the vector of the seed word w_t whose topic is t in Word2Vec is represented by v_{w_t}. The similarity between the word w and the topic word w_t can be calculated by equation (18).

$$\text{sim}(w, w_t) = \cos(\theta) = \frac{v_w \cdot v_{w_t}}{|v_w| \times |v_{w_t}|}. \tag{18}$$

Similar to the SLDA model, the HME-LDA will set the parameter α_d separately for each document (each sentence in the SLDA model) based on the similarity between the vocabulary and seed words. The biased parameters can be calculated by equation (19).

$$\alpha_d = \frac{\sum_{i=1}^{N_d} \text{sim}(w_{d,i}, w_t)}{\sum_{t'}^{T} \sum_{i}^{N_d} \text{sim}\left(w_{d,i}, w_t'\right)} \times \alpha_{\text{base}}. \tag{19}$$

In equation (19), N_d is the number of all words in the current sentence, T is the number of topics, $w_{d,i}$ is the i-th word in the current sentence, and t is the seed word.

Based on the similarity between the vocabulary and seed words, the SLDA model will set parameters separately for each topic. These biased parameters can be calculated by equation (20), equation (21), and equation (22).

$$\beta_{t,A} = \text{sim}(w, w_t) \times \beta_{\text{base}}, \tag{20}$$

$$\beta_{t,P} = \text{sim}(w, w_{t,P}) \times \beta_{\text{base}}, \tag{21}$$

$$\beta_{t,N} = \text{sim}(w, w_{t,N}) \times \beta_{\text{base}}. \tag{22}$$

Different from SLDA, HME-LDA introduces the

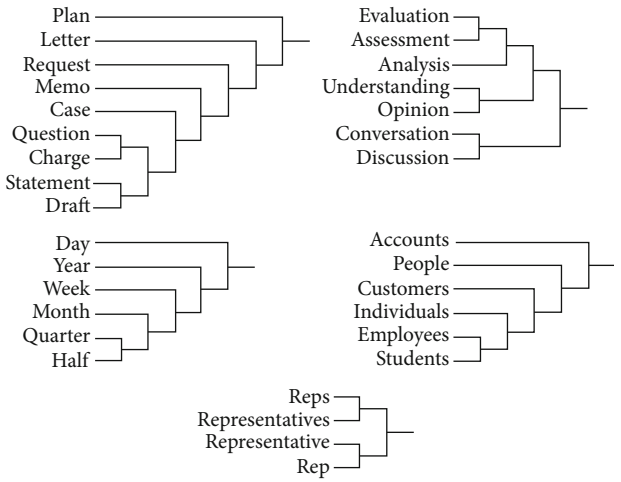


Figure 7: The results presentation of hierarchical clustering methods.

parameter δ to control the sentiment polarity of each sentence. The parameter $\delta_{d,q}$ can be calculated by equation (23).

$$\delta_{d,q} = \frac{\sum_i^{N_d} \mathrm{sim}\left(w_{d,i}, w_{t,q}\right)}{\sum_{q' \in \{P,N\}} \sum_i^{N_d} \mathrm{sim}\left(w_{d,i}, w_{t,q'}\right)} \times \delta_{\mathrm{base}}. \tag{23}$$

Similar to SLDA, HME-LDA uses the method of Gibbs sampling for solving. And the variables involved in the solution process have the same meanings as those in Tables 1 and 2.

Then, we use equation (24) to sample the topic of each sentence.

$$p\left(z_{d,n} = t \mid z_{\neg d,n}, y_{\neg d,n}, v_{\neg d,n}, \cdot\right) \propto \frac{n_{w_{d,n}}^{t,A} + \beta_{w_{d,n}}^{t,A}}{\sum_v^V \left(n_v^{t,A} + \beta_v^{t,A}\right)} \times \frac{n_{w_{d,n}}^{t,P} + \beta_{w_{d,n}}^{t,P}}{\sum_v^V \left(n_v^{t,P} + \beta_v^{t,P}\right)} \times \frac{n_{w_{d,n}}^{t,N} + \beta_{w_{d,n}}^{t,N}}{\sum_v^V \left(n_v^{t,N} + \beta_v^{t,N}\right)} \times \left(n_{d,t} + \alpha_{d,t}\right). \tag{24}$$

Equation (25) and equation (26) are used to sample $y_{d,n}$ and $v_{d,n}$,

$$p\left(y_{d,n} = u \mid z_{d,n} = t, \cdot\right) \propto \frac{n_{w_{d,n}}^{t,u} + \beta_{w_{d,n}}^{t,u}}{\sum_v^V \left(n_v^{t,u} + \beta_v^{t,u}\right)} \times \frac{e^{\lambda_u \times f_{d,n}}}{\sum_{u' \in \{A,O\}} e^{\lambda_{u'} \times f_{d,n}}}, u \in \{A, O\}, \tag{25}$$

$$p\left(v_{d,n} = q \mid z_{d,n} = t, \cdot\right) \propto \frac{n_{w_{d,n}}^{t,q} + \beta_{w_{d,n}}^{t,q}}{\sum_v^V \left(n_v^{t,q} + \beta_v^{t,q}\right)} \times \left(n_{d,q} + \delta_{d,q}\right). \tag{26}$$

In the corpus, the approximate probability of the topic t in sentence d can be calculated by equation (27),

$$\theta_d^t = \frac{n_{d,t} + \alpha_{d,t}}{n_d + \sum_{t'}^T \alpha_{d,t'}}. \tag{27}$$

With t as the topic, the approximate probability that the word $w_{d,n}$ is the opinion target can be calculated by equation (28),

$$\varphi_{w_{d,n}}^{t,A} = \frac{n_{w_{d,n}}^{t,A} + \beta_{w_{d,n}}^{t,A}}{\sum_v^V \left(n_v^{t,A} + \beta_v^{t,A}\right)}. \tag{28}$$

With t as the topic, the approximate probability that the word $w_{d,n}$ is the positive opinion word can be calculated by equation (29),

$$\varphi_{w_{d,n}}^{t,P} = \frac{n_{w_{d,n}}^{t,P} + \beta_{w_{d,n}}^{t,P}}{\sum_v^V \left(n_v^{t,P} + \beta_v^{t,P}\right)}. \tag{29}$$

With t as the topic, the approximate probability that the word $w_{d,n}$ is the negative opinion word can be calculated by equation (30),

$$\varphi_{w_{d,n}}^{t,N} = \frac{n_{w_{d,n}}^{t,N} + \beta_{w_{d,n}}^{t,N}}{\sum_v^V \left(n_v^{t,N} + \beta_v^{t,N}\right)}. \tag{30}$$

3. Results and Analysis

This chapter mainly trains the SLDA model in Section 2.1 and the HME-LDA model in Section 2.2 and, respectively, uses the above two models to extract the opinion targets and opinion review words on the *Restaurant* English data set, and then, this chapter will verify the feasibility of the models and analyse the experimental results.

3.1. Experimental Data Set. The data set of the experiment is from SemEval2016ABSA and Yelp. The original data of reviews on Yelp includes restaurants and hotels, so it needs to be screened. The test data is from the task B of SemEval2016ABSA. The test part of this experiment only pays attention to the classification of aspect-based opinion words, so the original test data need to be processed.

3.1.1. Original Data Set. SemEvel provides the training set and test set of the Restaurant reviews in XML format, where the size of the training set is 737 KB and the size of test set is 264 KB. The label structure of a sentence is shown in Figure 8.

The content in the label text is the original sentence, the attribute target in the label opinion is the opinion target, the category is the aspect category of reviews, and the polarity describes the sentiment polarity of the target. The model input proposed in this paper is the plain text without annotated information. The task of this paper is to extract the opinion target words and opinion sentiment opinion words and judge the polarity of the sentiment word. Here, the content in the text label needs to be extracted and added to the

```
<sentence id="1004293:1">
    <text>We, there were four of us, arrived at noon - the place was empty - and the
    staff acted like we were imposing on them and they were very rude.</text>
    <Opinions>
        <Opinion target="staff" category="SERVICE#GENERAL" polarity="negative" from="75"
        to="80"/>
    </Opinions>
</sentence>
```

Figure 8: The format of the SemEval data set.

corpus, and the content of target and category is extracted as test data. When verifying the experimental results based on the evaluation indicators, we only take the opinion target into account.

The size of the original data in Yelp is 231.2 MB, which contains lots of useless fields and covers a wide range of fields, including *Restaurant*, *Hotel*, and *Wine Bar*. There is a field, *business_categories*, which represents the realm of the review, in the Yelp data set. Therefore, the data can be filtered through the field above. The Yelp data is used to provide additional training sets and can be used to train Word2Vec and hierarchical clustering models to provide automatic annotated training sets for the maximum entropy model in HME-LDA.

3.1.2. Data Preprocessing. The Restaurant data set in SemEval is in XML format, while the format of Yelp data is CSV. Therefore, it is necessary to extract the data needed for this experiment from the files with two formats and carry out unified numbering. In the *Restaurant* data set, all sentences are annotated with text, so the sentences can be directly obtained from the labels. In the Yelp data set, the field, *business_categories*, stores the category of the review, and if the review contains the restaurant word, it is a restaurant-related review. The review of Yelp contains multiple sentences, so it can be split into multiple sentences by punctuations. Finally, the two txt files are used to store the extracted results, and one line in the file is a sentence. The xml.dom package and the csv package in python are used in the process. The amount of data finally extracted is shown in Table 3.

Repetitive words are removed from the statistics of the number of words, and no repetition words are removed in the statistics of the average length of sentences. When words are extracted from the sentences, they are separated by spaces and punctuation marks. Due to the need for additional tools for word type reduction, the word type reduction is not carried out here. Therefore, when counting the number of words, both the different tenses and the singular and plural forms of the same word are taken into account, resulting in the final number of words is larger than the fact.

There is additional annotated information in the SemEval corpus. The opinion target word and the aspect category of the word in the *Restaurant* review field can be extracted from the annotated information. The *opinion* label in the original xml file is extracted by python's xml.dom package, then the *attributes*, *target*, and *category* are extracted from the opinion label and make statistics. The statistical results are shown in Table 4.

Figure 9 generated by statistics in the training corpus of SimEval shows that the review category, *food*, accounts for a large proportion, while the review categories, *location* and *drinks*, account for a small proportion. The review category, *restaurant*, which contains a lot of semantics, is not usually the object of extraction. In this chapter, we only take the review categories, *food*, *service*, and *ambience* into account.

3.1.3. The Construction of the Experimental Data Set. The SemEval2016Restaurant review set is used in both the experimental data set and the test set. In the HME-LDA model, additional Yelp data sets are needed to train the hierarchical classification model and the Word2Vec model, and the amount of additional data provided will affect the final results as well. Therefore, the Yelp data set is divided into four training sets according to the number of sentences, which are shown in Table 5.

3.2. The Main Evaluation Indicators

3.2.1. The Evaluating Indicators of the Aspect Review Category of Sentences. Referring to the evaluation methods of Zhao et al. [22], this paper chooses accuracy P, recall R, and their harmonic value $F1$ as verification indicators, which are shown in equation (31), equation (32), and equation (33). In the MaxEnt-LDA [22] model, the topic of a sentence is undefined. According to the distribution of the topic words, the topics should be set to *food*, *service*, and *ambience* manually. Then, the sentence should be set to the corresponding topic according to the probability of the words appearing in each topic. The SLDA model proposed in this paper, as well as the HME-LDA model that has a fewer dependence on the language than SLDA, treats the seed word as the topic word and the aspect category of reviews. Therefore, the topic of a sentence can be determined directly by the distribution in the model.

$$P = \frac{\text{The number of correct predictions}}{\text{The total number of projections}}, \tag{31}$$

$$R = \frac{\text{The number of correct predictions}}{\text{The total correct number}}, \tag{32}$$

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{33}$$

3.2.2. The Evaluation Indicators of the Opinion Target. In the models proposed in this paper, the distribution information of the opinion target based on the specific topic is stored in $\varphi_{t,A}$. The $\varphi_{t,A}$ preserves the word probability based on the

Table 3: The information of the experimental data set.

Index	SemEval	Yelp
The number of reviews	2000	85000
The number of words	3373	67066
The average length of sentence	13	13

Table 4: The information of the annotating data.

The aspect category of reviews	The number of sentences	The number of words before repetition	The number of words after repetition
Food	952	952	420
Restaurant	258	258	90
Service	324	324	57
Drinks	96	96	51
Ambience	228	228	94
Location	22	22	9

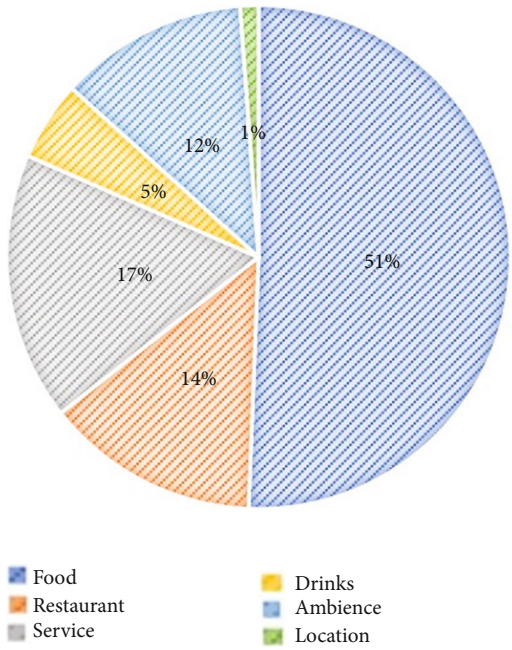


Figure 9: The proportion of review categories in the SemEval corpus.

Table 5: The information of the Yelp training set.

The name of the training sets	The number of sentences	The number of sentences
Yelp2K	2000	3809
Yelp4K	4000	5553
Yelp10K	10000	8948
Yelp20K	20000	12490

features of a topic model, and the same word may exist in different topics, so it is impossible to calculate the extracted accuracy and recall rate directly with $\varphi_{t,A}$. Referring to the scheme of Zhao et al. [22], n words with the highest probability of each topic in $\varphi_{t,A}$ are extracted and are treated as representatives, and then, their accuracy is calculated. In Section 1, the annotated data set contains information about the opinion targets and the categories of reviews they belong to. From this, the n words with the highest frequency of occurrence are selected as references for each review category. If n is 5, 10, 20, respectively, the accuracy rate $P_{t@n}$ can be calculated according to equation (31), where t is the topic number and n is the number of words taken. The average accuracy of extraction is expressed by $P_{t@n}$, which is calculated by equation (34).

$$P_{@n} = \frac{\sum_{t=0}^{T} P_{t@n}}{T}. \tag{34}$$

3.3. The Experimental Results and Analysis

3.3.1. The Setting of Experimental Parameters. In this experiment, the parameters related to the topic model are set as $\alpha_{\text{base}} = 50/T$, $\beta_{\text{base}} = 0.01$, and $\delta_{\text{base}} = T$, where T is the number of topics, and $T = 3$ in the subsequent experiment. In addition, the other one we need to set is the seed word and the cluster number of the hierarchical classification model. The seed word set is divided into two groups for comparison. One is the seed word set *A {food, service, ambience}*, and the other is seed word set *B {chicken, staff, atmosphere}*. The cluster number of the hierarchical classification model does not directly determine the number of categories. The cluster number is set to 200 according to experience.

The parameter setting of the comparison model MaxEnt-LDA [22] is the same as the original text. When training maximum entropy, the features of the model can be divided into three categories: word, part of speech, and part of speech plus word. The HME-LDA model proposed in this paper only uses the feature *word*. In the comparative experiment, the feature of the maximum entropy of the MaxEnt-LDA [22] model is chosen as the *word*.

3.3.2. The Influence of Seed Word Set on SLDA and HME-LDA. Figures 10 and 11 show the impact of seed words on the SLDA model and the HME-LDA model when they are, respectively, set to seed word set *A {food, service, ambience}* and seed word set *B {chicken, staff, atmosphere}*. When using the seed word set *A*, the model is recorded as SLDA-A, HME-LDA-A. When using the seed word set *B*, the model is recorded as SLDA-B, HME-LDA-B. Besides, the HME-LDA model additionally uses Yelp10K as the training set of the automatic annotated method.

It can be seen from the experimental results in Figure 10 that the selection of seed word sets has a greater impact on the SLDA model, while the impact on the HME-LDA model is not significant. Under the topics of *food* and *ambience*, the accuracy, recall rate and $F1$ value of the SLDA-A model are higher than those of SLDA-B, and when under the topic of *service*, the evaluation indexes above of SLDA-A model are

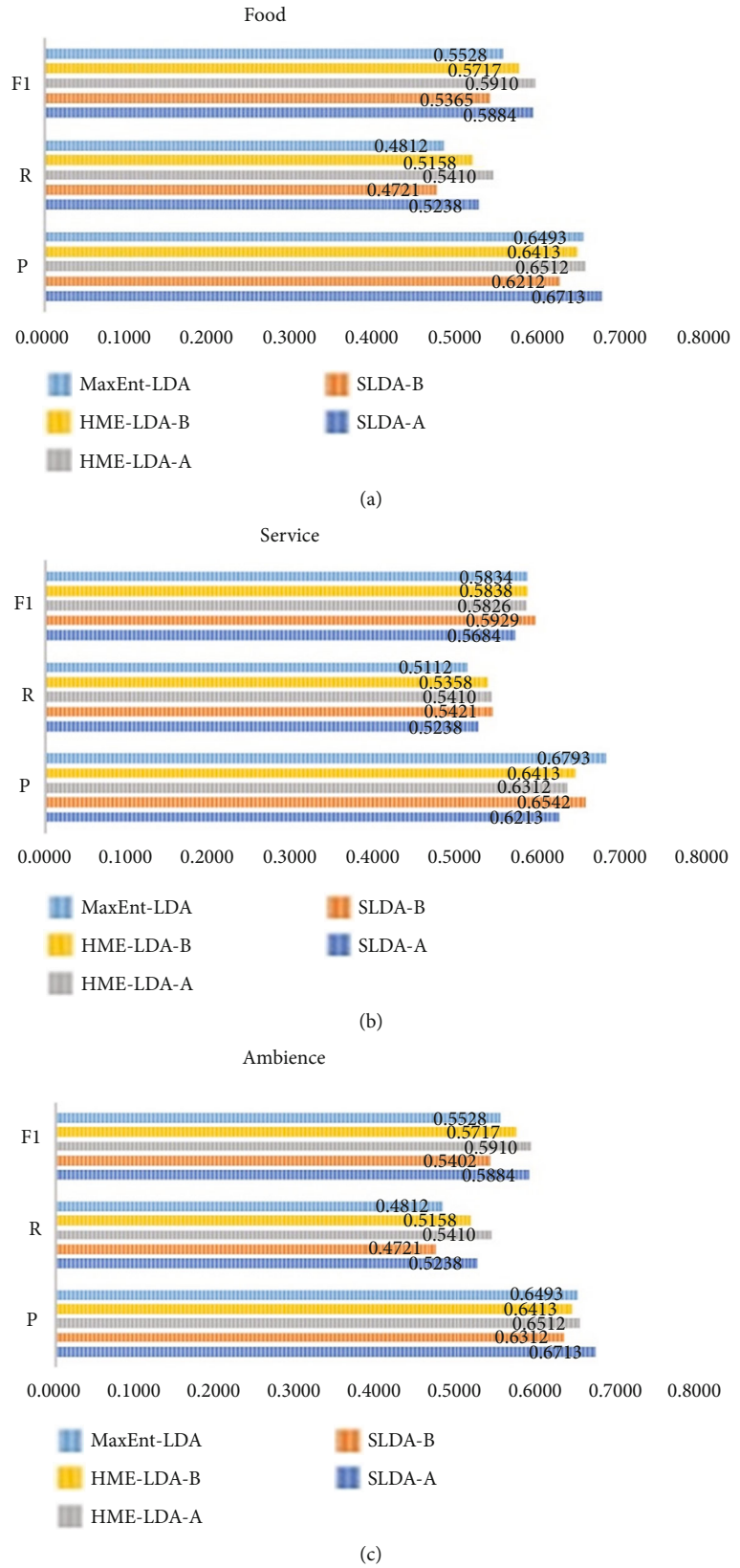


Figure 10: The influence of seed word sets on the extraction of sentence review categories: (a) evaluation index value of review category Food; (b) evaluation index value of review category Service; (c) evaluation index value of review category Ambience.

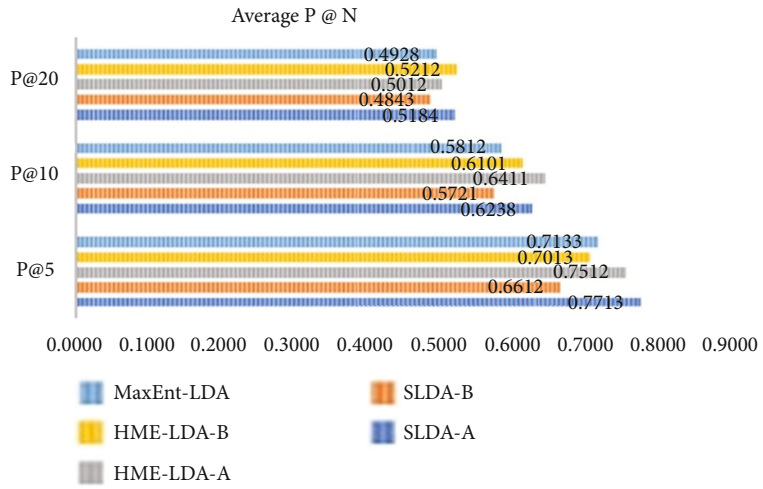


Figure 11: The influence of the seed word sets on P@N.

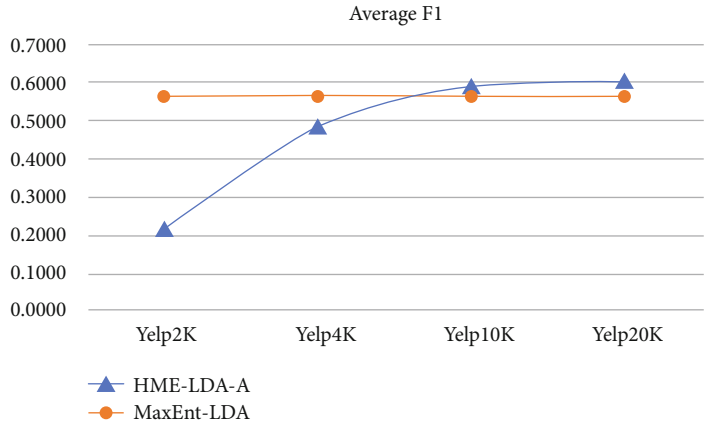


Figure 12: The F1 value of HME-LDA in training sets of different sizes.

lower than those of the SLDA-B model. This is related to the features of WordNet and SentiWordNet. In WordNet, the similarity between words is mainly calculated by the path between words. When the seed word *food* is used as the initial position, all branches only need to trace up to *food*. When a word in a branch of *food* is used as the seed word, the words of the other branches need to trace back to *food* first, and then searched down. This way makes the number of paths increase, which reduces the similarity of the words, thus affecting the results, while WordNet mainly preserves the concept of words and fails to perform knowledge reasoning, that is, it fails to infer the relationship between *waiter* and *service* to improve the similarity between them. When the seed word is replaced by *staff*, the explicit conceptual noun will make it easier to find noun words such as *waiter*, thus improving the effect of the SLDA. The eigenvector of the maximum entropy classifier in the HME-LDA model is taken as word order information, ignoring the conceptual problem of the word itself, so the choices of seed words have little effect on the final results. The evaluation indicators of SLDA-A and HME-LDA-A models are both slightly better than MaxEnt-LDA.

The results of Figure 11 show that the smaller the value of n, the higher the accuracy of the opinion target extraction in each model. And all models, namely, MaxEnt-LDA, SLDA, and HME-LDA, are suitable for extracting the opinion target words with the highest correlation, which is consistent with the comment habits of people. Most people only focus on a few aspect opinion targets of the comment objects, and the information people want to get from the comments is also based on their most concerned aspect. Similar to the results in Figure 11, when the seed word set is changed from A to B, the accuracy of the SLDA model decreases a lot, while the accuracy of the HME-LDA model decreases little, which has a great relationship with the word classification method of the SLDA. Meanwhile, it can be seen that SLDA-A and HME-LDA-A are slightly higher in the accuracy of the opinion target extraction than that of the MaxEnt-LDA.

3.3.3. The Influence of the Size of Training Set on HME-LDA. In the HME-LDA model, the effect of the maximum entropy model is related to the accuracy of automatically annotated data sets. The size of the training set that is used for training the hierarchical classification model will affect the accuracy

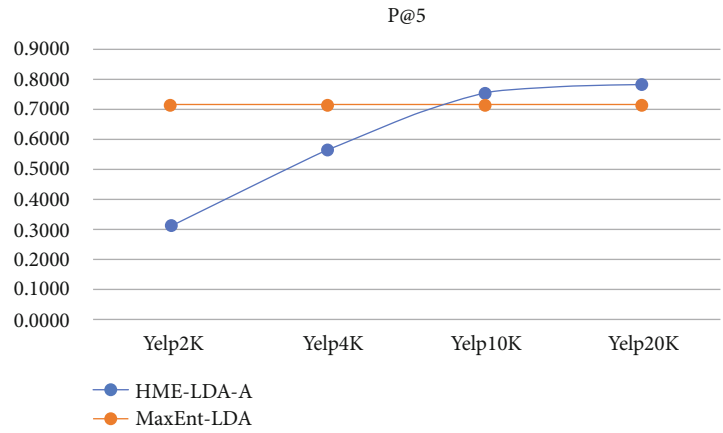


Figure 13: The P@5 value of HME-LDA in training sets of different sizes.

of automatic data annotation. Since it is difficult to calculate the accuracy of the automatic annotated data set, the impact of the training set size on the HME-LDA model is indicated by the evaluation indexes F1 and P@5 of the final HME-LDA model. Here, F1 takes the mean value of F1 under the three topics, and *P@5* indicates the mean value of the accuracy of the opinion target extraction in each review category when taking the top five values. In the experiment, the seed word is set to the seed word set *A {food*, *service*, *ambience}*. The experimental results are shown in Figures 12 and 13.

The value of the MaxEnt-LDA model is taken as a reference in the above figures. The Yelp data set is not used in the MaxEnt-LDA model, so the evaluation indicators of the MaxEnt-LDA model in these figures remain unchanged. With the increase of training set size, the effect of HEM-LDA is on the rise. When the training set reaches 10K, the effect of the HME-LDA model tends to be stable and slightly better than that of the MaxEnt-LDA model. The reason why the effect of the model is related to the size of the training set is because the larger the training set is, the more effective information it contains. Meanwhile, since the number of clusters in the hierarchical classification model is constant, when the amount of data is small, the number of words in each cluster is relatively small, and the accuracy of classification is relatively low. When the amount of data increases, the accuracy of classification will be improved relatively, which will affect the final effect of the model.

4. Conclusions

This paper mainly studies the methods to reduce the dependence of models on annotated data by focusing on the topic of aspect-based opinion mining. The unsupervised LDA topic model has good expansibility. Based on the LDA topic model, this paper introduces the two types of dictionary tools, WordNet and SentiWordNet, to propose the SLDA model. In order to further reduce the dependence on language tools, the maximum entropy model and the method of automatically annotating data are introduced to propose an optimized model HME-LDA. The experiments show that both the SLDA model and the HME-LDA model have good results on accuracy and recall rate without relying on the annotated data. Therefore, the two optimized models will give more detailed and accurate information for cryptocurrency investors in the blockchain to assist them in better decision-making.

Data Availability

Firstly, the Yelp data set used to support the findings of this study can be available from the https://www.yelp.com/dataset. Secondly, the SemEval2016ABSA data set used to support the findings of this study are included within the article: "SemEval-2016 Task 5: Aspect Based Sentiment Analysis". Also, this dataset can be available from the http://alt.qcri.org/semeval2016/task5/index.php?id=data-and-tools.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Social Science Fund Planning Project of Ministry of Education of People's Republic of China "Research on Data Service and Guarantee for the Fourth Paradigm of Social Science" (20YJA870017).

References

[1] C. Song, H. Jung, and K. Chung, "Development of a medical big-data mining process using topic modeling," *Cluster Computing*, vol. 22, no. S1, pp. 1949–1958, 2019.

[2] L. Carson K-S, "Big data analysis and mining," in *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics*, pp. 15–27, IGI Global, 2019.

[3] Q. Li, S. Li, S. Zhang, and J. Hu, "A Review of Text Corpus-Based Tourism Big Data Mining," *Applied Sciences*, vol. 9, no. 16, p. 3300, 2019.

[4] S. Lee, Y. Hyun, and M. J. Lee, "Groundwater potential mapping using data mining models of big data analysis in Goyang-si, South Korea," *Sustainability*, vol. 11, no. 6, article 1678, 2019.

[5] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining Text Data*, pp. 415–463, Springer, Boston, MA, 2013.

[6] K. Liu, L. Xu, and J. Zhao, *Opinion target extraction using word-based translation model*, pp. 1346–1356, 2012.

[7] Z. Chen, A. Mukherjee, and B. Liu, *Aspect extraction with automated prior knowledge learning*, 2014.

[8] H. Cheng, Z. Xie, Y. Shi, and N. Xiong, "Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM," *IEEE Access*, vol. 7, 2019.

[9] M. Pontiki, D. Galanis, H. Papageorgiou et al., *SemEval-2016 task 5: aspect based sentiment analysis*, pp. 19–30, 2016.

[10] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: aspect based sentiment analysis," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 27–35, 2014.

[11] Y. F. Zeng, T. Lan, and Z. F. Wu, "Bi-memory based attention model for aspect level sentiment classification," *Chinese Journal of Computers*, vol. 8, pp. 1845–1857, 2019.

[12] T. Chen, R. Xu, and X. Wang, "Improving Sentiment Analysis Via Sentence Type Classification Using BiLSTM-CRF and CNN," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[13] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2018.

[14] T. Hofmann, "Probabilistic latent semantic indexing," *ACM SIGIR Forum*, vol. 51, no. 2, pp. 211–218, 2017.

[15] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, p. 993, 2013.

[16] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, *Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs*, 2007.

[17] F. Li, C. Han, M. Huang et al., *Structure-Aware Review Mining and Summarization*, vol. 2, pp. 653–661, 2010.

[18] D. Blei and J. McAuliffe, "Supervised topic models," *Advances in Neural Information Processing Systems*, vol. 3, 2010.

[19] D. Ramage, D. Hall, R. Nallapati, and C. Manning, *Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora*, vol. 1, pp. 248–256, 2009.

[20] H. Cheng, D. Feng, X. Shi, and C. Chen, "Data quality analysis and cleaning strategy for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 61, 2018.

[21] H. Cheng, Z. Su, N. Xiong, and Y. Xiao, "Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model," *Information Sciences*, vol. 329, pp. 461–477, 2016.

[22] W. Zhao, J. Jiang, H. Yan, and X. Li, *Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid*, pp. 56–65, 2010.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8834450, 14 pages
https://doi.org/10.1155/2020/8834450

Research Article

Leveraging Social Relationship-Based Graph Attention Model for Group Event Recommendation

Guoqiong Liao[1] **and Xiaobin Deng**[1,2]

[1]*School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, China*
[2]*Department of Construction Engineering, Jiangxi Water Resources Institute, Nanchang 330013, China*

Correspondence should be addressed to Xiaobin Deng; dengxiaobin83@163.com

Received 9 September 2020; Accepted 13 October 2020; Published 30 October 2020

Academic Editor: Amr Tolba

Recently, event-based social networks(EBSN) such as Meetup, Plancast, and Douban have become popular. As users in the networks usually take groups as an unit to participate in events, it is necessary and meaningful to study effective strategies for recommending events to groups. Existing research on group event recommendation either has the problems of data sparse and cold start due to without considering of social relationships in the networks or makes the assumption that the influence weights between any pair of nodes in the user social graph are equal. In this paper, inspired by the graph neural network and attention mechanism, we propose a novel recommendation model named leveraging social relationship-based graph attention model (SRGAM) for group event recommendation. Specifically, we not only construct a user-event interaction graph and an event-user interaction graph, but also build a user-user social graph and an event-event social graph, to alleviate the problems of data sparse and cold start. In addition, by using a graph attention neural network to learn graph data, we can calculate the influence weight of each node in the graph, thereby generating more reasonable user latent vectors and event latent vectors. Furthermore, we use an attention mechanism to fuse multiple user vectors in a group, so as to generate a high-level group latent vector for rating prediction. Extensive experiments on real-world Meetup datasets demonstrate the effectiveness of the proposed model.

1. Introduction

With the rapid development of the Internet and information technology, social networks are becoming more and more necessary and diversified. As a new type of social networks combining online and offline interactions, event-based social networks (EBSN) have received more and more attentions in recent years. In EBSN, recommending events to groups become an important task since users usually take groups as an unit to participate in events. Recently, a lot of research works on group recommendation have appeared [1–7]. For example, Li et al. [1] proposed a collective matrix factorization and event-user neighbor (CMF-EUN) model to recommend events. Purushotham et al. [2] proposed a Bayesian model based on collaborative filtering for personalized group event recommendation. Yuan et al. [4] introduced the COnsensus Model (COM) probability model to simulate the generation process of group preferences for events. However, these studies do not take into account the social relationship information of users in the group, they have problems of data sparse and cold start. Pham et al. [5] proposed a general graph-based model called HeteRS, in which a random walk method was used to solve different recommendation tasks on EBSN. Yin et al. [6] proposed a general graph-based embedding model (GEM) to solve the event-partner recommendation problem. Although these studies consider the social relationship of users in the group and can alleviate the problems of data sparse and cold start to some extent, when they learn feature, they assume that the influence weights between any pair of users are equal, lessening the accuracy of results they obtain.

In EBSN, there is abundant interactive information and social information, as shown in Figure 1. As it can be seen, EBSN is a heterogeneous social network which includes three kinds of nodes (i.e., groups, users and events), two kinds of interactive information (i.e., group-user interaction and

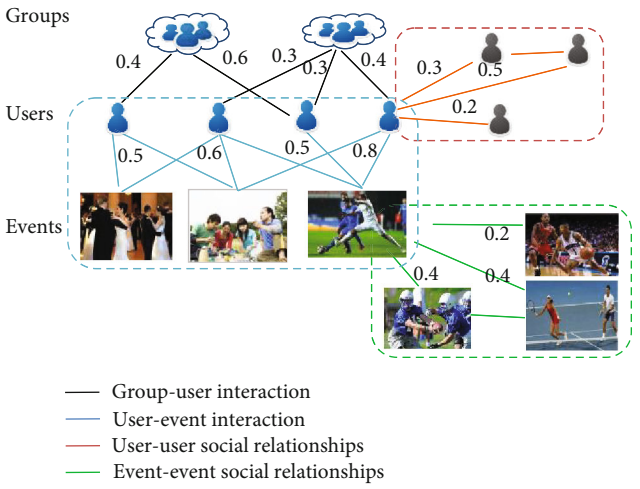


Figure 1: Heterogeneous social network of EBSN.

user-event interaction), and two kinds of social relationship information (i.e., user-user relationship and event-event relationship). The numbers on the interaction line between the users and the events represent the user's preference for scoring events, and the numbers on other line segments in the figure represent the influence weights. For example, as shown in the upper right corner of the figure, a user has three friends, and the influence weights of these three friends on the user are: 0.3, 0.2, 0.5, and the sum of the weights is 1.

In recent years, the deep neural network technology of graph data has made great progress [8]. Now, graph convolution neural networks and graph attention networks have been getting more and more attention, since they can both construct complex heterogeneous graph relational data and also capture the different influence weights of the nodes in the graph. To this end, we propose a novel method leveraging social relationship-based graph attention model (SRGAM) for group event recommendation in EBSN. Specifically, we first build a mathematical model for users in the groups. We construct user latent vectors from two aspects: event aggregation and user social aggregation. We use the attention mechanism to weight and sum the latent vectors of different users to obtain the latent vectors of the group. Then, we build a mathematical model for the events. Graph attention networks are used for event modeling, and event-user interaction graphs and event-event social graphs are used to generate event latent vectors. Finally, we use the group latent vectors and the event latent vectors to perform a dot product to generate a predicted score for recommendation. Our experimental results have demonstrated that the suggested model performs better than state-of-the-art methods on real-world datasets. In summary, the key contributions of the paper are listed as follows:

(i) We propose a novel group event recommendation model SRGAM, which uses a graph neural network and attention mechanism. To the best of our knowledge, this is the first work to recommend events to groups using the heterogeneous graph attention network in EBSN

(ii) We construct a heterogeneous social network for EBSN, which not only contain a user-event interaction graph and an event-user interaction graph but also has a user-user social graph and an event-event social graph, to further alleviate the problems of data sparse and cold start

(iii) We use a graph attention neural network for learning graph data and calculate the different influence weights between the nodes in the heterogeneous network, which facilitate to generate more reasonable latent user vectors and event vectors. The attention mechanism is also used to perform weighted fusion on different user vectors in the group to generate high-level group latent vectors

(iv) We conduct extensive experiments on the real-world datasets to demonstrate the effectiveness of our proposed method

The remainder of this paper is organized as follows. Section 2 introduces our model framework. Section 3 shows the model realization. The comparison experiment on the real-world datasets is presented in Section 4. Related work is reviewed in Section 5. Finally, we conclude the paper and discuss the future work in Section 6.

2. The Proposed Model

In this section, we first introduce the definitions and notations used in this paper and then give an overview about the model framework.

2.1. Definitions and Notations. Let $G = \{g_1, g_2, \cdots, g_{|G|}\}$, $U = \{u_1, u_2, \cdots, u_{|U|}\}$, and $E = \{e_1, e_2, \cdots, e_{|E|}\}$ be the sets of groups, users, and events in EBSN, respectively, where $|G|$, $|U|$, and $|E|$, are the numbers of groups, users, and events, respectively, and g_i is composed of the users from U.

$\mathbf{M} \in \mathbb{R}^{|G|\times|E|}$ is a group-event rating matrix, which is also called a group-event graph. The value in the matrix, denotes as r_{ij}, is the rating score that g_i gives to e_j. We employ 0 to represent an unknown rating from g_i to e_j.

$\mathbf{N_1} \in \mathbb{R}^{|U|\times|E|}$is a user-event rating matrix, which is also called a user-event graph. s_{ij} is the score value of u_i for e_j. If user u_i did not participate in the event e_j, then $s_{ij} = 0$.

$\mathbf{N_2} \in \mathbb{R}^{|E|\times|U|}$is an event-user rating matrix, which is also called an event-user graph. $\mathbf{N_2}$ is the transposed matrix of $\mathbf{N_1}$, so s_{ji} is the score value of u_i for e_j. In addition, users can establish social relationships with each other.

$\mathbf{H} \in \mathbb{R}^{|U|\times|U|}$ is a user-user relationship matrix, which is also called a user-user social graph, where $H_{ij} = 1$ indicates that there is a friend relationship between u_i and u_j, and $H_{ij} = 0$ indicates that there is no friend relationship between them.

$\mathbf{T} \in \mathbb{R}^{|E|\times|E|}$ is an event-event relationship matrix, which is also called an event-event social graph, where $T_{ij} = 1$ indicates that there is an influence relationship between e_i and e_j, and $T_{ij} = 0$ indicates that there is no influence relationship between them.

Let A_i be the set of social friends of user u_i, B_i be the set of events which user u_i participated in, C_j be the set of users who participated in event e_j, D_j be the set of social events of event e_j, and F_i be the set of users who are in the group g_i. Given $\mathbf{M}$, $\mathbf{N_1}$, $\mathbf{N_2}$, $\mathbf{H}$, and $\mathbf{T}$, the goal of the paper is to predict the missing score value in $\mathbf{M}$. We use $\mathbf{p}_i \in \mathbb{R}^d$ to represent the embedding vector of user u_i and $\mathbf{q}_j \in \mathbb{R}^d$ to represent the embedding vector of the event e_j, where d is the dimension of the embedding vector.

The notations used in the paper are listed in Table 1.

2.2. Model Framework. The framework of our proposed leveraging social relationship-based graph attention model (SRGAM) is shown in Figure 2, which consists of three modules, i.e., group modeling, event modeling, and rating prediction.

The first module is group modeling, which is used to learn the latent vector of the groups. This module is divided into two stages: user modeling and user fusion. During user modeling, the user-event graph and the user-user social graph are used to learn user representation from two different graph perspectives. Therefore, we introduce two kinds of aggregations to process these two different graphs separately. The one is event aggregation, which can understand users through the interaction between the users and events in the user-event graph. The other is social aggregation, and the relationship between users in a social graph can model users from a social perspective. Then, we intuitively obtain user latent vectors by combining information from the event space and social space. At last, the attention-weighted summation of different user latent vectors is performed to obtain the latent vector of the group.

The second module is event modeling, used to learn the latent vector of events. Due to the mutual influence between different events, similar to the user-user social graph, we propose the event-event social graph. This part uses the event-user graph and event-event social graph to learn the representation of events from two different graph perspectives.

The last module is the rating prediction, to learn model parameters for prediction by integrating the information both of the group modeling and event modeling.

3. Model Inference and Learning

In this section, we will introduce the implement techniques of the three modules in detail.

3.1. Group Modeling. The goal of group modeling is to generate the latent vector of each group $g_i \in G$, represented as $\mathbf{g}_i \in \mathbb{R}^d$. This component is divided into two stages: user modeling and user fusion. The user modeling stage involves integrating the information of the user-event interactive graphs and the user-user social graphs. As shown in the upper part of Figure 2, there are two kinds of aggregations, i.e., event aggregation and social aggregation, to learn $u_i \in U$'s latent vectors $\mathbf{u}_i^E \in \mathbb{R}^d$ in the event space and $\mathbf{u}_i^S \in \mathbb{R}^d$ in the social space. Then, we combine $\mathbf{u}_i^E$ and $\mathbf{u}_i^S$ to form u_i's final latent vector $\mathbf{u}_i$. In the user fusion stage, the latent vectors $\mathbf{u}_i$ of different users in the group are weighted and summed through an attention mechanism, to generate the group latent vector $\mathbf{g}_i$ finally.

3.1.1. User Modeling Stage. This stage includes three parts: event aggregation, user social aggregation, and user latent vectors learning.

(1) Event Aggregation. The user-event graph contains not only the user's interaction with the event but also the user's opinion on the event (or the user's rating of the event). We provide a way to jointly capture the interactions and opinions in the user-event graph to learn the user latent vectors $\mathbf{u}_i^E$ in the event space. To represent this aggregation mathematically, we use the following function:

$$\boldsymbol{u}_i^E = \sigma\left(\boldsymbol{W} \cdot \mathrm{AggreE}\left(\left\{\boldsymbol{x}_{ij}, \forall j \in B_i\right\}\right) + \boldsymbol{b}\right), \tag{1}$$

where σ represents the nonlinear activation function, AggreE() represents the event aggregation function, and $\mathbf{x}_{ij}$ is a representation vector of interaction and opinions between u_i and e_j. B_i is the set of events participated by the user u_i. $\boldsymbol{W}$ and $\mathbf{b}$ represent the weight and bias in the neural network, respectively.

For an event, the user can express his/her opinion (or rating), denoted as r. These opinions about the event can capture the user's preference for the event, which can help model the latent vector of the event space user. To model opinions, for each opinion r, we introduce an opinion embedding vector $\mathbf{s}_r \in \mathbb{R}^d$, which represents each opinion r as a dense vector representation. For the interaction between user u_i and event e_j with r, we combine the event embedding vector $\mathbf{q}_j$ and the opinion embedding vector $\mathbf{s}_r$ through a multilayer perceptron (MLP) to generate an opinion-aware interaction representation $\mathbf{x}_{ij}$. MLP takes the series of event embedding $\mathbf{q}_j$ and opinion embedding $\mathbf{s}_r$ as input, and the output of MLP is an opinion perception representation $\mathbf{x}_{ij}$ of the interaction between u_i and e_j, as follows:

$$\boldsymbol{x}_{ij} = \mathrm{MLP}\left(\left[\boldsymbol{q}_j \oplus \boldsymbol{s}_r\right]\right), \tag{2}$$

where $\oplus$ represents the connection of two vectors.

For the aggregation function AggreE(), the vector in $\{\mathbf{x}_{ij}, \forall j \in B_i\}$ is generally averaged; then, equation (1) becomes

$$\mathbf{u}_i^E = \sigma\left(\mathbf{W} \cdot \left\{\sum_{j\in B_i} \alpha_i \mathbf{x}_{ij}\right\} + \mathbf{b}\right), \tag{3}$$

where $\alpha_i = 1/|B_i|$, and all events use a fixed equal weight value. This method assumes that all events interacting with

Table 1: Notation.

Symbols	Descriptions
G	The set of group g_i $(i = 1, 2, \cdots \mid G \mid)$
U	The set of user u_i $(i = 1, 2, \cdots \mid U \mid)$
E	The set of event e_j $(j = 1, 2, \cdots \mid E \mid)$
M	The group-event graph
N_1	The user-event graph
N_2	The event-user graph
H	The user-user graph
T	The event-event graph
A_i	The set of social friends of user u_i
B_i	The set of events interacted by user u_i
C_j	The set of users who participated in event e_j
D_j	The set of social events of event e_j
F_i	The set of users who are in the group g_i
p_i	The embedding of user u_i
q_j	The embedding of event e_j
s_r	The embedding of the rating value r, $r \in [0, 1]$
$\mathrm{u_i}^{\mathrm{E}}$	The event space user latent factor from event set B_i of user u_i
u_i^S	The social space user latent factor from social friends set A_i of user u_i
u_i	The user latent factor of user u_i, combining from event space $\mathbf{u}_i^E$ and social space $\mathbf{u}_i^S$
x_{ij}	The interaction and opinions representation of event e_j for user u_i
e_j^U	The user space event latent factor from user set C_j of event e_j
e_j^S	The social space event latent factor from social events set D_j of event e_j
e_j	The event latent factor of event e_j, combining from user space $\mathbf{e}_j^U$ and social space $\mathbf{e}_j^S$
y_{ji}	The interaction and opinions representation of user u_i for event e_j
g_i	The group latent vector of g_i
α_{ij}	The attention weight of event e_j in contributing to $\mathbf{u_i}^{\mathrm{E}}$
β_{ij}	The social attention weight of user u_j in contributing to $\boldsymbol{u}_i^S$
θ_{ji}	The attention weight of user u_i in contributing to $\boldsymbol{e}_j^U$
γ_{ji}	The social attention weight of event e_i in contributing to e_j^S
μ_{ij}	The attention weight of user u_j in contributing to g_i
r_{ij}	The rating value of event e_j by group g_i
r_{ij}'	The predicted rating value of event e_j by group g_i
$\oplus$	The concatenation operator of two vectors
W	The weight in neural network
b	The bias in neural network

a user have the same effect on the user. However, since the impact of the interaction on the user may vary greatly, this may not be reasonable. Therefore, we need to allow interactions to make different contributions to users' latent factors by assigning a weight to each interaction.

In order to alleviate the shortcomings of the mean-based aggregation method, we use an attention mechanism to learn how different events affect users.

$$\mathbf{u}_i^E = \sigma\left(\mathbf{W} \cdot \left\{\sum_{j \in B_i} \alpha_{ij} \mathbf{x}_{ij}\right\} + \mathbf{b}\right), \tag{4}$$

where α_{ij} is the weight of event e_j to user u_i's event space

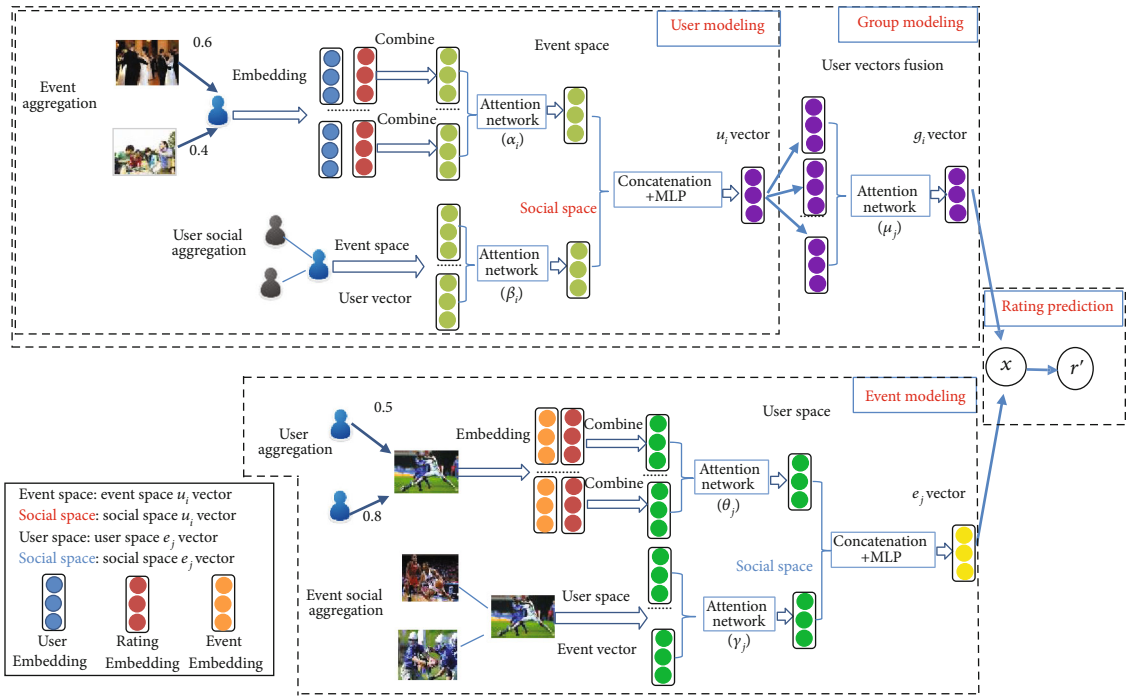


FIGURE 2: The overall architecture of the proposed model. It contains three major components: group modeling, event modeling, and rating prediction. α_i, β_i, θ_j, γ_j, and μ_i are the attention weights. Note that the numbers on the edges of the graph denote the rating score.

latent vector. We use an attention neural network to learn α_{ij}. The input of the attention network is the opinion perception representation $\mathbf{x}_{ij}$ of the user interaction and the embedding vector $\mathbf{p}_i$ of the target user. The attention network is defined as follows:

$$\alpha'_{ij} = \sigma\left(\mathbf{W} \cdot \left[\mathbf{x}_{ij} \oplus \mathbf{p}_i\right] + \mathbf{b}\right) \tag{5}$$

The final attention weight is obtained using the Softmax function to normalize the above attention score, which can be interpreted as the contribution of the event interaction to the user u_i's event space user latent vector $\mathbf{u}_i^E$, specifically:

$$\alpha_{ij} = \frac{\exp\left(\alpha'_{ij}\right)}{\sum_{j \in B_i} \exp\left(\alpha'_{ij}\right)}. \tag{6}$$

(2) User Social Aggregation. The user's preferences are similar to or affected by his/her social friends. Therefore, we need to incorporate social information to more accurately simulate user latent factors. At the same time, the strength of the connection between users can further influence the user's behavior from the social graph. In other words, the learning of user latent factors in social space should consider the heterogeneity of social relations. Therefore, we introduce an attention mechanism to select representative social friends to characterize the user's social information and then summarize the information. In order to represent user latent vectors from social perspective, a user latent vector in social space, i.e., $\mathbf{u}_i^S \in \mathbb{R}^d$, is introduced for u_i. Specifically, $\boldsymbol{u}_i^S$ is a vector which aggregates the latent vectors in event space of the users in friend set A_i of u_i, as follows:

$$\boldsymbol{u}_i^S = \sigma\left(\boldsymbol{W} \cdot \mathrm{AggreUS}\left(\left\{\boldsymbol{u}_j^E, \forall j \in A_i\right\}\right) + \boldsymbol{b}\right), \tag{7}$$

where AggreUS() represents the aggregation function of the user's social friends.

As with event aggregation, for the aggregation function AggreUS(), the vector in $\{\mathbf{u}_j^E, \forall j \in A_i\}$ is generally averaged. Then, equation (7) becomes

$$\mathbf{u}_i^S = \sigma\left(\mathbf{W} \cdot \left\{\sum_{j \in A_i} \beta_i \mathbf{u}_j^E\right\} + \mathbf{b}\right), \tag{8}$$

where $\beta_i = 1/|A_i|$, and a fixed equal weight value is used for all social friends. This method assumes that all friends who have social interactions with the user have the same impact on the user. This is obviously unreasonable, so we also use the attention mechanism to learn the different influence weights of different friends on the user, as follows:

$$\mathbf{u}_i^S = \sigma\left(\mathbf{W} \cdot \left\{\sum_{j \in A_i} \beta_{ij} \mathbf{u}_j^E\right\} + \mathbf{b}\right), \tag{9}$$

$$\beta'_{ij} = \sigma\left(\mathbf{W} \cdot \left[\mathbf{u}_j^E \oplus \mathbf{p}_i\right] + \mathbf{b}\right), \tag{10}$$

$$\beta_{ij} = \frac{\exp\left(\beta'_{ij}\right)}{\sum_{j \in A_i} \exp\left(\beta'_{ij}\right)}, \tag{11}$$

where $\beta_{ij}{}'$ is the attention weight, and β_{ij} is the normalized attention weight.

(3) User Latent Vector Learning. In order to learn better user latent factors, since user social graphs and user-event graphs provide information about users from different perspectives, and event space user latent vectors and social space user latent vectors need to be considered together. We propose to connect them and learn through the multilayer perceptron (MLP) and finally obtain the user latent vector $\boldsymbol{u}_i$.

$$\boldsymbol{u}_i = \mathrm{MLP}\left(\boldsymbol{u}_i^E \oplus \boldsymbol{u}_i^S\right). \tag{12}$$

3.1.2. User Fusion Stage. A group contains several users, and the latent vectors of multiple users are fused to form the latent vector of the group. Because different users have different positions and weights in the group, we use the attention mechanism to learn the different weights of users and then use the weights and user latent vectors to perform weighted summation to obtain the group latent vector. The corresponding formula is as follows:

$$\mathbf{g}_i = \sum_{j \in F_i} \mu_{ij} \mathbf{u}_j, \tag{13}$$

$$\mu'_{ij} = \sigma\left(\mathbf{W} \cdot \mathbf{u}_j + \mathbf{b}\right), \tag{14}$$

$$\mu_{ij} = \frac{\exp\left(\mu'_{ij}\right)}{\sum_{j \in F_i} \exp\left(\mu'_{ij}\right)}, \tag{15}$$

where $\mu_{ij}{}'$ is the attention weight, and μ_{ij} is the normalized attention weight.

3.2. Event Modeling. As shown in the lower part of Figure 2, event modeling is used to learn the event latent vector $\boldsymbol{e}_j$. Event modeling also includes three parts: user aggregation, event social aggregation, and event latent vectors learning.

3.2.1. User Aggregation. We adopt the similar method to learn event latent vectors in user space through user aggregation. The information to be aggregated here comes from the user set C_j that interacts with the event e_j. Even for the same event, different users may express different opinions during the user-event interaction. These opinions from different users can capture the characteristics of the same event, which can help model the event latent vector. We connect the user vector interacting with the event and the user's opinion vector on the event and then learn through MLP to get the interactive user representation $\mathbf{y}_{ji}$ of opinion perception. The formula is as follows:

$$\boldsymbol{y}_{ji} = \mathrm{MLP}([\boldsymbol{p}_i \oplus \boldsymbol{s}_r]). \tag{16}$$

Then, in order to learn the event latent vector $\boldsymbol{e}_j{}^U$, we also aggregate different users interacting with the event e_j. The aggregation function is *AggreU()*. It mainly aggregates the user's opinion-aware interaction representation in $\{\boldsymbol{y}_{ji}, \forall i \in C_j\}$, as follows:

$$\boldsymbol{e}_j^U = \sigma\left(\boldsymbol{W} \cdot \mathrm{AggreU}\left(\left\{\boldsymbol{y}_{ji}, \forall i \in C_j\right\}\right) + \boldsymbol{b}\right). \tag{17}$$

In addition, we also use the attention mechanism to learn the different influence weights of different users on the same event.

$$\mathbf{e}_j^U = \sigma\left(\mathbf{W} \cdot \left\{\sum_{i \in C_j} \theta_{ji} \mathbf{y}_{ji}\right\} + \mathbf{b}\right), \tag{18}$$

$$\theta'_{ji} = \sigma\left(\mathbf{W} \cdot \left[\mathbf{y}_{ji} \oplus \mathbf{q}_j\right] + \mathbf{b}\right), \tag{19}$$

$$\theta_{ji} = \frac{\exp\left(\theta'_{ji}\right)}{\sum_{i \in C_j} \exp\left(\theta'_{ji}\right)}, \tag{20}$$

where $\theta_{ji}{}'$ is the attention weight, and θ_{ji} is the normalized attention weight.

3.2.2. Event Social Aggregation. Just as a social graph can be constructed between users, a social graph of events can also be constructed between events. We assume that if the similarity between a pair of events exceeds a certain threshold, it is considered that the pair have an event social relationship. Specifically, the social space event latent vector $\mathbf{e}_j{}^S$ of the event e_j is a user space event latent vector that aggregates the social event set D_j of the event e_j as follows:

$$\boldsymbol{e}_j^S = \sigma\left(\boldsymbol{W} \cdot \mathrm{AggreES}\left(\left\{\boldsymbol{e}_i^U, \forall i \in D_j\right\}\right) + \boldsymbol{b}\right), \tag{21}$$

where AggreES() represents the aggregation function of the social relationship of the event.

Events that have friendships with one event have different influence weights on this one event. Therefore, we also use an attention mechanism to learn the influence weights for event social aggregation and perform weighted summation for the purpose of aggregation.

$$\mathbf{e}_j^S = \sigma\left(\mathbf{W} \cdot \left\{\sum_{i \in D_j} \gamma_{ji} \mathbf{e}_j^U\right\} + \mathbf{b}\right), \tag{22}$$

$$\gamma'_{ji} = \sigma\left(\mathbf{W} \cdot \left[\mathbf{e}_j^U \oplus \mathbf{q}_j\right] + \mathbf{b}\right), \tag{23}$$

$$\gamma_{ji} = \frac{\exp\left(\gamma'_{ji}\right)}{\sum_{i \in D_j} \exp\left(\gamma'_{ji}\right)}, \tag{24}$$

where $\gamma_{ji}{}'$ is the attention weight, and γ_{ji} is the normalized attention weight.

3.2.3. Event Latent Vectors Learning. In order to learn better event latent factors, since event social graphs and event-user graphs provide information about events from different perspectives, we connect the user space event latent vectors and social space event latent vectors and learn through MLP to obtain the event latent vector $\boldsymbol{e}_j$.

$$\mathbf{e}_j = MLP\left(\mathbf{e}_j^U \oplus \mathbf{e}_j^S\right). \tag{25}$$

3.3. Rating Prediction and Model Training. In the work of this paper, we use the score prediction as a recommendation task. Specifically, we use the latent vector $\mathbf{g}_i$ of the group and the latent vector $\mathbf{e}_j$ of the event to perform a dot product to generate a predicted score:

$$r'_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{e}_j. \tag{26}$$

In order to estimate the parameters of the model, we need to specify an objective function for optimization. Since the task we focus on in this work is the rating prediction, the objective function is expressed as

$$\text{Loss} = \frac{1}{|N|}\sum_{i,j\in N}\left(r'_{ij} - r_{ij}\right)^2, \tag{27}$$

where $|N|$ is the number of observed scores, r_{ij} is the actual score of group g_i on event e_j, and $r_{ij}{}'$ is the predicted score of group i on event j.

To optimize the objective function, AdamOptimizer as the optimizer is used to optimize the mean square error function. Our model contains three embeddings: event embedding, $\mathbf{q}_j$, user embedding, $\mathbf{p}_i$, and opinion embedding, $\mathbf{s}_r$. During the training phase, they are randomly initialized and learned together. Because the original features are very large and sparse, we do not use one-hot encoding vectors to represent each user and each event, but we embed high-dimensional sparse features into low-dimensional latent space so that the model can be more easily trained. To alleviate the problem of overfitting in optimizing deep neural network models, we adopt the dropout strategy to randomly drop some neurons during the training process.

4. Experiments

In this section, we compare the experimental results of SRGAM with five baseline methods and three variant methods on two real-world datasets. Generally speaking, our experimental goal is to answer the following research questions (RQ):

(i) RQ1: how does SRGAM perform as compared to existing advanced methods?

(ii) RQ2: what are the effects of each component in our method on performance?

(iii) RQ3: how do the hyper-parameters affect our model's performance?

4.1. Experimental Settings

4.1.1. Datasets. We conducted experiments on the real-world datasets from two different cities in the event social network Meetup (https://www.meetup.com/): Chicago and San Diego. The actual score data of a group on an event is a floating point number between 0 and 5; data with a score of 0 was discarded since the 0 score most likely occurred as a result of a user not participating. For users in the Meetup dataset do not explicitly rate events, we calculate the similarity based on the theme of each user and the theme of each participating event and normalize it to obtain a value in the range of [0,1], which is used as the user's score data for the event. There is also no explicit friend relationship in the Meetup dataset, so we consider two users who have participated in three or more events together as friend relationship. We assume that if the degree of acquaintance between two events exceeds 0.4, then it is considered that there is an influence relationship between them. The statistics of the two datasets are shown in Table 2.

4.1.2. Evaluation Metrics. We use two indicators to evaluate the model performance: root mean square error (RMSE) and mean absolute error (MAE). The smaller the RMSE and MAE indicators, the higher the accuracy of the model.

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{(i,j)\in T}\left(r'_{ij} - r_{ij}\right)^2}, \tag{28}$$

$$\text{MAE} = \frac{1}{|T|}\sum_{(i,j)\in T}\left|r'_{ij} - r_{ij}\right|, \tag{29}$$

where (i, j) represents group i and event j, $|T|$ is the number of scores in the test set, $r_{ij}{}'$ is the score predicted by the model, and r_{ij} is the actual score of the test set.

4.1.3. Baselines. In order to evaluate the performance, our model is compared with the other five methods and three variant methods. These methods are described in detail below.

(i) GARec [9]: the model uses genetic algorithms to learn the interactions between users in a group and can capture the different importance of users in the group

(ii) HeteRS [5]: this method proposes to construct a heterogeneous graph model, that makes full use of the interaction between groups, users, events, and tags and complete three different recommendation tasks: recommend groups to users, recommend tags to groups, and recommend events to users

(iii) RRWR-S [10]: this method constructs a heterogeneous graph to represent the relationships between various entities and social networks and then uses restart to perform a reverse random walk to obtain user-event proximity values from the constructed graph

Table 2: Statistics of the datasets.

Dataset	Chicago	San Diego
Total groups	5675	8462
Total events	2365	3529
Total users	18164	35543
Ratings of groups to events	41427	47387
Ratings of users to events	148944	231029
Users social connections	58124	81748
Events social connections	13244	21526

(iv) AGREE [11]: for the first time, the attention mechanism in the neural network is used for group recommendation. The fusion strategy is dynamically learned based on the input data. At the same time, user-item interaction information is introduced to improve the effect of group recommendation

(v) MoSAN [12]: this model introduces the attention mechanism into group recommendation, and it proposed to use an attention mechanism to obtain the influence of each user in the group. The model can learn the influence weight of each user in the group, and consider different contexts, and then recommend items to groups based on their members' weighted preferences. This method can model complex group decision-making processes

(vi) SRGAM1: this is a variant of the model in this paper, which deletes the user's social relationships and retains the event's social relationships in the model. It recommends events to groups without considering the user's social relationships

(vii) SRGAM2: this is a variant of the model in this paper, which deletes the event's social relationships and retains the user's social relationships in the model. It is that recommends events to groups without considering the event's social relationships

(viii) SRGAM3: this is a variant of the model in this paper, which removes the attention mechanism in the model. It assumes that the impact of different events on users, the impact of different users on events, the impact of social interaction, and the impact of different users on groups are equal

4.1.4. Parameter Settings. For each dataset, we use x% as the training set to learn the parameters, (1-x%)/2 as the verification set to adjust the hyper parameters, and (1-x%)/2 as the test set, where x% takes the values {50%, 70%, 90%}. For the embedding size d, we test the value of [8, 16, 32, 64, 128]. The batch size and learning rate were searched in [32, 64, 128, 512] and [0.0001, 0.001, 0.01, 0.1], respectively. When training with neural networks, we uniformly used three hidden layers and ReLu as activation functions, the training period Epoch was 20, and the group size tested was the number of members of [1–20].

4.2. Overall Performance Comparison (RQ1). First, we compare the recommendation performance of our model with the other five models. Figure 3 shows the RMSE and MAE on the two datasets of Chicago and San Diego. We have the following main findings:

(i) The performance of the GARec model is the worst. Although this method considers the different weights of users in the group, it only uses the score information and ignores the user-user social relationship information, so it leads to poor performance

(ii) The HeteRS and RRWR-S models are better than the GARec model, because these two methods use a graph structure to model social relationships. The GARec model only uses rating information and ignores user-user social relationship information. These results indicate that social relationship information is necessary in group recommendation

(iii) The AGREE and MoSAN models are better than the HeteRS and RRWR-S models. Although the first two do not consider the user's social relationship information, they use the attention mechanism and neural network for deep learning, which can learn better groups and events. This shows that the attention mechanism and neural network play an important role in group recommendation

(iv) Our model SRGAM is superior to the other five baseline methods. Our model combines user-user social relationships and event-event social relationships. At the same time, our model also uses attention mechanisms and neural networks for learning to get a better representation of groups and events

Overall, the comparison results of various model methods show that (1) social information is helpful for group recommendation, (2) attention mechanism and neural network can improve recommendation performance, and (3) our model is better than other baseline methods.

4.3. Ablation Study(RQ2). In this section, we conduct ablation research from two aspects: social relationship ablation and attention mechanism ablation.

4.3.1. Social Relationship Ablation. SRGAM1 and SRGAM2 are two variants of the model in this paper, they delete user social relationship and event social relationship, respectively. Figure 4 shows the comparison of the RMSE and MAE indicators of the SRGAM1, SRGAM2, and SRGAM models on the two datasets. We have the following findings:

(i) The performance of the model SRGAM in this paper is better than that of the two models of the variant, indicating that both the user's social relationship information and the event social relationship information play a positive role in group event recommendation

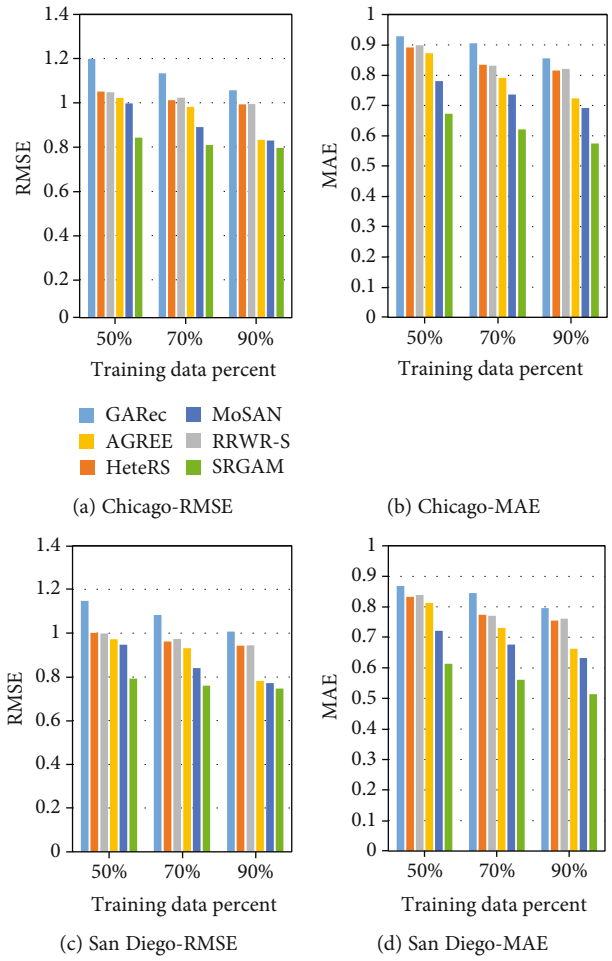


Figure 3: Performance comparison of different methods on two datasets.

(ii) The performance of the SRGAM2 model is better than that of the SRGAM1 model, which shows that the user's social relationship information is more important than the event social relationship information to a certain extent

4.3.2. Attention Mechanism Ablation. As shown in the previous chapter, SRGAM3 is another variant of the model in this paper, one which removes the attention mechanism in the model. Since there are five places in the model that use the attention mechanism, five variants were produced: SRGAM3-α, SRGAM3-β, SRGAM3-θ, SRGAM3-γ, and SRGAM3-μ. They are defined as follows:

(i) SRGAM3-α: when the user's events are aggregated, the event attention α of the model SRGAM is removed. This variant method uses the average aggregation method for event aggregation; that is, the impact of each event on the user is considered equal

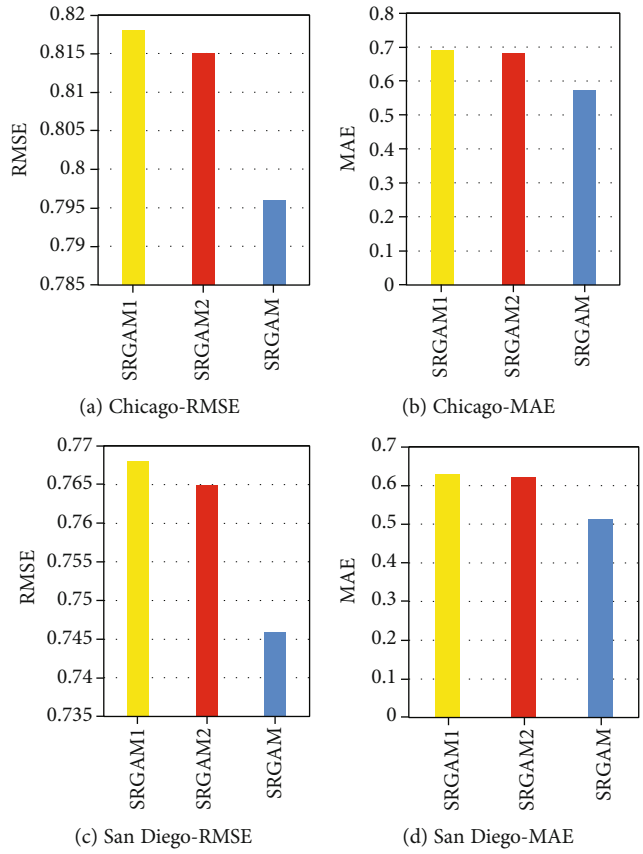
(ii) SRGAM3-β: when the user's social relationships are aggregated, the social attention β of the model SRGAM is removed. This variant method uses the average aggregation method for user social aggregation; that is, the impact of each friend on the user is considered equal

(iii) SRGAM3-θ: when the event's users are aggregated, the user attention θ of the model SRGAM is removed. This variant method uses the average aggregation method for user aggregation; that is, the impact of each user on the same event is considered equal

(iv) SRGAM3-γ: when the event's socials are aggregated, the social attention γ of the model SRGAM is removed. This variant method uses the average aggregation method for event social aggregation. That is, the impact of each event on the same event is considered equal

(v) SRGAM3-μ: when the users of the group are aggregated, the user attention μ of the model SRGAM is removed. This variant method uses the average aggregation method for user vector aggregation; that is, the impact of each user on the group is considered equal

Figure 5 shows the performance comparison of this model and the five attention variant models. From the results, we have the following findings:



(a) Chicago-RMSE (b) Chicago-MAE

(c) San Diego-RMSE (d) San Diego-MAE

Figure 4: Effect of social network on two datasets.

(i) The impact of multiple events involving a user on the latent vector of the user in the event space is not equal. Similarly, the impact of multiple users participating in an event on the latent vector of the event in the user space is not equal. Based on this assumption, our model was implemented by using two different attention mechanisms (α and θ). From the experimental results, SRGAM3-α and SRGAM3-θ did not perform as well as SRGAM. The results show that the attention mechanism plays a role in event aggregation and user aggregation

(ii) The influence of the user's friends on the user is not equal. Similarly, the influence of the event's friends on the event is not equal. Based on this assumption, our model was implemented by using two different attention mechanisms (β and γ). From the experimental results, SRGAM3-β and SRGAM3-γ did not perform as well as SRGAM. The results show that the attention mechanism plays a role in user social aggregation and event social aggregation

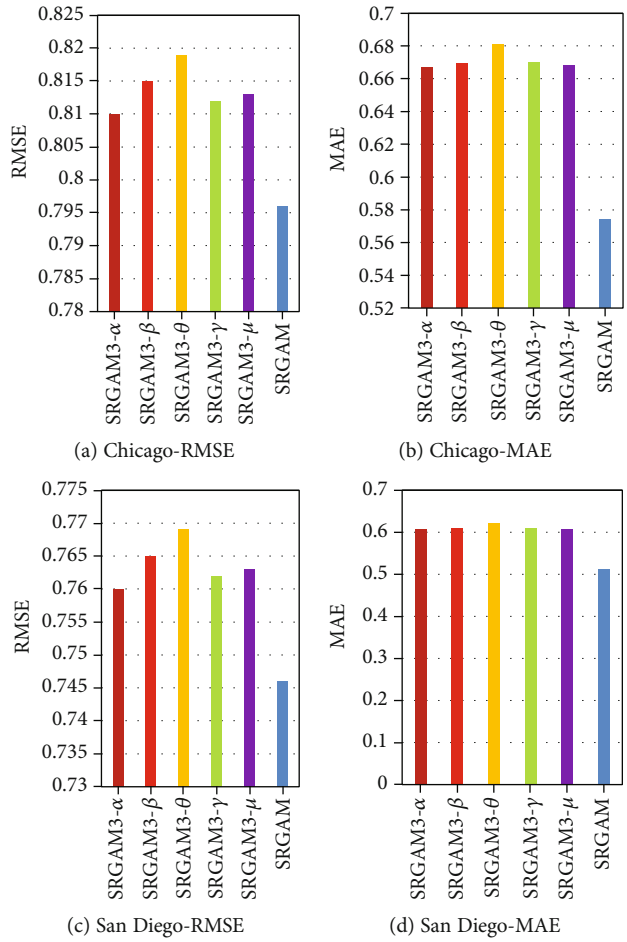
(iii) Different users in a group have different influences on group decisions. Based on this assumption, our model was implemented by using the attention mechanism (μ). From the experimental results, SRGAM3-μ is not perform as well as SRGAM. The results show that the attention mechanism plays a role in the aggregation of users in the group

To sum up, SRGAM can capture various influence weights in users, groups, and events through application of an attention mechanism, which improves the recommendation performance.

4.4. Effect of Hyper Parameters on the Model Performance (RQ3). There are two main hyper parameters in the SRGAM model: embedding size and group size.

4.4.1. Embedding Size. We select five values, {8, 16, 32, 64, 128}, for testing. As shown in Figure 6, for the two datasets, the small embedding size cannot fully express the characteristics of users, events, and ratings, while an excessively large embedding size will lead to overfitting of the model and a decrease in learning efficiency. Therefore, we set the embedding size to 32.

4.4.2. Group Size. When we train the model, we try four different group sizes (the number of members in the group): {1-5, 6-10, 11-15, 16-20} and found that the model achieve best performance when the group size was 6-10 users on two datasets. As shown in Figure 7, a large group size causes the model complexity to increase and reduces recommendation performance, while a small group size reduces the effect of



(a) Chicago-RMSE (b) Chicago-MAE

(c) San Diego-RMSE (d) San Diego-MAE

Figure 5: Effect of attention mechanism on two datasets.

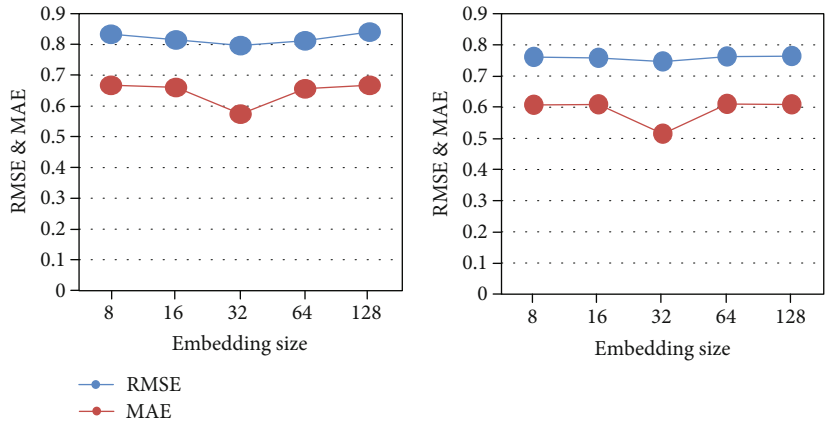
the attention mechanism and reduces the recommendation effect.

5. Related Work

In this section, we discuss some related work including general group recommendation, group recommendation based on the graph model, graph neural network, and graph attention network.

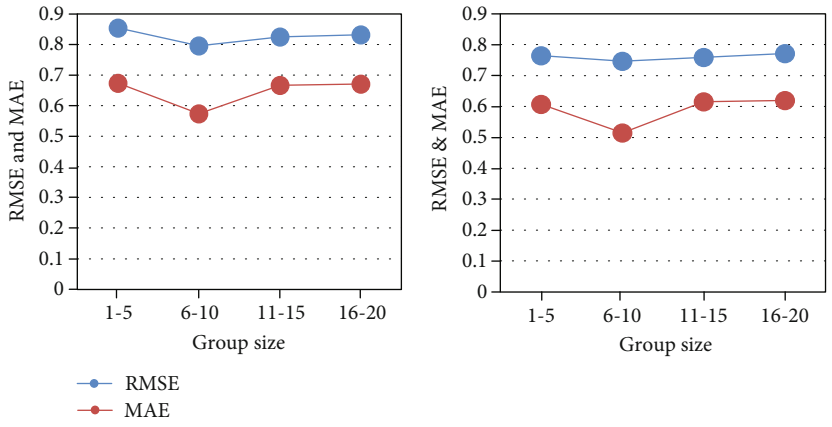
5.1. General Group Recommendation. The key issue of the general group recommendation is preference fusion. The preference fusion methods can be divided into two categories: model fusion and recommendation fusion [17]. Yu et al. [13] proposed a model fusion method based on item feature scoring. Yuan et al. [4] proposed a probabilistic model named COM to simulate the generation process of group preferences for events. Kagita et al. [14] proposed a fusion method based on priority sequence mining and a virtual user model. O'Connor et al. [16] used the least painful strategy to fuse a user recommendation list and obtain a group recommendation list. Chen et al. [9] used a genetic algorithm to optimize the weight of group members and proposed a group recommendation method combining collaborative filtering and genetic algorithm. Naamani-Dery et al. [15] show that it is possible to find a balance between the size of the recommendation list and the cost of group preference extraction and thereby, reduce the size of the group recommendation list using an iterative preference extraction method. Cao et al. [11] proposed the AGREE model. Where for the first time, the attention mechanism in the neural network was used for group recommendation. Lucas et al. [12] proposed the MoSAN model, which also introduced the attention mechanism to group recommendation. It proposes using an attention mechanism to identify the influence level of each user in the group. However, as these methods seldom consider social relationship information, they have to be confronted with the problems of data sparse and cold start.

5.2. Group Recommendation Based on the Graph Model. The recommendation method based on a graph model is widely used in the recommendation field. Meng et al. [20] divided graph-based social recommendation methods into graph-based recommendation methods and link prediction methods. The graph is the most natural and direct representation of social network relationships in EBSN. There have been many studies on group recommendation in EBSN based on graph models, including [5–7, 10, 18, 19, 21]. Li et al. [10,



(a) Chicago (b) San Diego

Figure 6: Effect of embedding size on two datasets.



(a) Chicago (b) San Diego

Figure 7: Effect of group size on two datasets.

18] and Liu et al. [19] constructed a heterogeneous graph to express the interaction between different types of entities in EBSN. The perform random walks on the graph to obtain candidate events with high convergence probability. Pham et al. [5] proposed a general graph-based model HeteRS, in which a random walk method was used to solve the different recommendation tasks on EBSN. Yin et al. [6] proposed a general graph-based embedding model (GEM) to solve the event-partner recommendation problem. Liu et al. [21] constructed a primary graph (PG) based on the entities and their relationships in the EBSN and calculated the user-event similarity score by applying the random walk restart algorithm on the PG. However, these methods assume that the social impact of each friend on users is equal in importance.

5.3. Graph Neural Network and Graph Attention Network. Graph neural network (GNN) and graph attention network (GAN) have become a hotspot in the field of deep learning in recent years. Recently, the related work has focused on using CNN to model more general graph structure data [8, 22–25]. Specifically, Thomas et al. [8] proposed a graph convolution network for semisupervised graph classification. The model learns the node representation by using node attributes and graph structure. It consists of multiple graph convolutional layers, and each layer updates the node representation using the representation of the current node and its neighbors. Through this process, it can capture the dependencies between nodes. However, in the original formula, when updating the node representation, all neighbors are given a static weight. Existing studies use GAN to solve social impact analysis [26], graph node classification [27], dialogue generation [28], and association matching [29]. In addition, some research work such as [28, 30, 31] not only uses GAN technology to build user-friend social networks and user-project network graphs but also can calculate different influence weights of friends on users.

Zhang et al. [32] proposed the HetGNN model, which can jointly consider heterogeneous graph information as well

as heterogeneous contents information of each node effectively. Wang et al. [33] thought that a graph has strong relationship expression ability and proposed a user identity linkage method based on a heterogeneous graph attention network mechanism (UIL-HGAN). Wang et al. [34] proposed a novel heterogeneous graph neural network based on the hierarchical attention, including node-level and semantic-level attentions.

However, the goals of the above research are not about group event recommendations. Therefore, they are different from the problems studied in this paper.

6. Conclusion and Future Work

The paper present a leveraging Social Relationship based Graph Attention Model (SRGAM) for group event recommendation. The SRGAM model uses the social relationship information of users and events to alleviate the data sparse and cold start problems inherent in group event recommendation. We adopt an attention mechanism inside users, events and groups, and learn the different influence weights of various factors, and finally generate high-level comprehensive feature vectors of groups and events, which makes the prediction score of groups participation events more accurate. Experimenting on two real-world datasets, our SRGAM model performs best.

SRGAM has considered the social relationship between users and events, but does not consider the social relationship between groups. Thus, in future, we intend to integrate the group social relationship into the model, to further improve recommendation performance.

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 they have no conflict of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (No. 61772245), the Jiangxi Provincial Graduate Innovation Fund (No. YC2019-B093), and Science and Technology Project of Jiangxi Provincial Department of Education (No. GJJ181349).

References

[1] M. Li, D. Huang, B. Wei, and C. D. Wang, "Event recommendation via collective matrix factorization with event-user neighborhood," in *Intelligence Science and Big Data Engineering. IScIDE 2017. Lecture Notes in Computer Science, vol 10559*, Y. Sun, H. Lu, L. Zhang, J. Yang, and H. Huang, Eds., Springer, Cham, 2017.

[2] S. Purushotham and C. C. J. Kuo, "Modeling group dynamics for personalized group-event recommendation," in *Social Computing, Behavioral-Cultural Modeling, and Prediction. SBP 2015. Lecture Notes in Computer Science, vol 9021*, N. Agarwal, K. Xu, and N. Osgood, Eds., pp. 405–411, Springer, Cham, 2015.

[3] Y. Gu, J. Song, W. Liu, L. Zou, and Y. Yao, "CAMF: context aware matrix factorization for social recommendation," *Web Intelligence*, vol. 16, no. 1, pp. 53–71, 2018.

[4] Q. Yuan, G. Cong, and C. Lin, "COM: a generative model for group recommendation," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 163–172, New York, NY, USA, August 2014.

[5] T. Pham, X. Li, G. Cong, and Z. Zhang, "A general graph-based model for recommendation in event-based social networks," in *2015 IEEE 31st International Conference on Data Engineering*, pp. 567–578, Seoul, South Korea, April 2015.

[6] H. Yin, L. Zou, Q. V. H. Nguyen, Z. Huang, and X. Zhou, "Joint event-partner recommendation in event-based social networks," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 929–940, Paris, France, April 2018.

[7] M. M. Gonzalez, "A general recommendation model for heterogeneous networks," *Computing Reviews*, vol. 58, no. 7, pp. 418–418, 2017.

[8] T. N. Kipf and W. Max, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations*, pp. 1–10, Toulon, France, 2017.

[9] Y.-L. Chen, L.-C. Cheng, and C.-N. Chuang, "A group recommendation system with consideration of interactions among group members," *Expert Systems with Applications*, vol. 34, no. 3, pp. 2082–2090, 2008.

[10] Y. Mo, B. Li, B. Wang, L. T. Yang, and M. Xu, "Event recommendation in social networks based on reverse random walk and participant scale control," *Future Generation Computer Systems*, vol. 79, pp. 383–395, 2018.

[11] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," in *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 645–654, Ann Arbor, MI, USA, June 2018.

[12] V. Lucas, N. Tuan-Anh, T. Yi, Y. Liu, G. Cong, and X. Li, "Interact and decide: medley of sub-attention networks for effective group recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 255–264, Paris, France, July 2019.

[13] Z. Yu, X. Zhou, Y. Hao, and J. Gu, "TV program recommendation for multiple viewers based on user profile merging," *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63–82, 2006.

[14] V. R. Kagita, A. K. Pujari, and V. Padmanabhan, "Virtual user approach for group recommender systems using precedence relations," *Information Sciences*, vol. 294, no. 3, pp. 15–30, 2015.

[15] L. Naamani-Dery, M. Kalech, L. Rokach, and B. Shapira, "Preference elicitation for narrowing the recommended list for groups," in *Proceeding of the 8th ACM Conference on Recommender Systems*, pp. 333–336, Silicon Valley, CA, USA, October 2014.

[16] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl, "PolyLens: a recommender system for groups of users," in *ECSCW*

2001, W. Prinz, M. Jarke, Y. Rogers, K. Schmidt, and V. Wulf, Eds., Springer, Dordrecht, 2002.

[17] Y. Zhang, Y. Du, and X. Meng, "Research on group recommender systems and their applications," *Chinese Journal of Computers*, vol. 39, no. 4, pp. 745–764, 2016.

[18] B. Li, B. Wang, Y. Mo et al., "A novel random walk and scale control method for event recommendation," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 228–235, Toulouse, France, July 2016.

[19] S. Liu, B. Wang, and M. Xu, "Event recommendation based on graph random walking and history preference reranking," in *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pp. 861–864, New York, NY, USA, August 2017.

[20] X.-W. Meng, S.-D. Liu, Y.-J Zhang, and X. Hu, "Research on social recommender systems," *Journal of Software*, vol. 26, no. 6, pp. 1356–1372, 2015.

[21] S. Liu, B. Wang, and M. Xu, "SERGE: successive event recommendation based on graph entropy for event-based social networks," *IEEE Access*, vol. 6, pp. 3020–3030, 2018.

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proceedings of the 2th International Conference on Learning Representations*, pp. 1–14, Banff, Canada, 2014.

[23] D. Michael, B. Xavier, and V. Pierre, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th Neural Information Processing Systems Conference*, pp. 3844–3852, Barcelona, SPAIN, 2016.

[24] H. Mikael, B. Joan, and L. Yann, "Deep convolutional networks on graph-structured data," 2015, http://arxiv.org/abs/1506.05163.

[25] K. Alex, S. Ilya, and E. Geoffrey, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[26] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "DeepInf: socail influence prediction with deep learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2110–2119, New York, NY, USA, July 2018.

[27] L. Gong and Q. Cheng, "Adaptive edge features guided graph attention networks," 2018, http://arxiv.org/abs/1809.02709.

[28] Q. Wu, H. Zhang, X. Gao et al., "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proceedings of the 2019 World Wide Web Conference*, pp. 2091–2102, New York, NY, USA, May 2019.

[29] T. Zhang, B. Liu, D. Niu, K. Lai, and Y. Xu, "Multiresolution graph attention networks for relevance matching," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 933–942, New York, NY, USA, October 2018.

[30] W. Fan, Y. Ma, Q. Li et al., "Graph neural networks for social recommendation," in *Proceedings of the 2019 World Wide Web Conference*, pp. 417–426, New York, NY, USA, May 2019.

[31] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pp. 555–563, New York, NY, USA, January 2019.

[32] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 793–803, New York, NY, USA, July 2019.

[33] R. Wang, H. Zhu, L. Wang, Z. Chen, M. Gao, and Y. Xin, "User identity linkage across social networks by heterogeneous graph attention network modeling," *Applied Sciences*, vol. 10, article 5478, 16 pages, 2020.

[34] X. Wang, H. Ji, C. Shi et al., "Heterogeneous graph attention network," in *Proceedings of the 2019 World Wide Web Conference*, pp. 2022–2032, New York, NY, USA, May 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8880430, 9 pages
https://doi.org/10.1155/2020/8880430

Research Article

A Deep Fusion Gaussian Mixture Model for Multiview Land Data Clustering

Peng Li,[1] Zhikui Chen,[1,2] Jing Gao,[1,2] Jianing Zhang,[1] Shan Jin,[1] Wenhan Zhao,[1] Feng Xia,[1,2] and Lu Wang[3,4,5]

[1]*School of Software Technology, Dalian University of Technology, Dalian 116620, China*
[2]*Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China*
[3]*College of Natural Resources and Environment, South China Agricultural University, Guangzhou 510642, China*
[4]*Guangdong Province Key Laboratory for Land Use and Consolidation, South China Agricultural University, Guangzhou 510642, China*
[5]*Guangdong Province Engineering Research Center for Land Information Technology, South China Agricultural University, Guangzhou 510642, China*

Correspondence should be addressed to Zhikui Chen; zkchen@dlut.edu.cn

Received 9 August 2020; Revised 14 September 2020; Accepted 28 September 2020; Published 17 October 2020

Academic Editor: Xiaojie Wang

With the rapid industrialization and urbanization, pattern mining of soil contamination of heavy metals is attracting increasing attention to control soil contamination. However, the correlation over various heavy metals and the high-dimension representation of heavy metal data pose vast challenges on the accurate mining of patterns over heavy metals of soil contamination. To solve those challenges, a multiview Gaussian mixture model is proposed in this paper, to naturally capture complicated relationships over multiviews on the basis of deep fusion features of data. Specifically, a deep fusion feature architecture containing modality-specific and modality-common stacked autoencoders is designed to distill fusion representations from the information of all views. Then, the Gaussian mixture model is extended on the fusion representations to naturally recognize the accurate patterns of the intra- and inter-views. Finally, extensive experiments are conducted on the representative datasets to evaluate the performance of the multiview Gaussian mixture model. Results show the outperformance of the proposed methods.

1. Introduction

With the rapid industrialization and urbanization over the world, environmental contamination is attracting increasing attention nowadays, which is caused by unreasonable usage of natural resources, such as the overuse of coal [1]. Among the environmental contamination, the status of soil contamination of heavy metals is the core concern of the public, with the scare of the heavy metal security of agricultural products that easily have a direct influence on our health [2]. A large number of researchers force on the control of soil contamination of heavy metals by mining intrinsic patterns hidden over various heavy metals which can do a favor to the contamination control and environmental protection. However, the correlation over various heavy metals and the high-dimension representation of heavy metal data pose vast challenges on the accurate mining of patterns over heavy metals of soil contamination. With the continuous development of industrialization and urbanization, more research is still required to capture effective patterns of high-dimension representation of heavy metal data, to control the soil contamination.

In recent years, large amounts of research have been proposed to learn patterns of data to improve our lives [3–8]. For example, Chen et al. used multivariate statistics and geostatistics to explore distributions of heavy metals in the soil of northwest China, which can capture pollution sources of heavy metals based on patterns of distributions [9].

Additionally, the chemical mass balance model, factor analysis, target transformation factor analysis, and principal component analysis are used to capture the complicated relationship of heavy metals [10–12]. Those statistical methods are able to mine patterns of heavy metals in simple cases where there are not many kinds of heavy metals. Also, they can only mine contamination patterns in a single view. In other words, those traditional statistical methods cannot well learn complex contamination patterns of heavy metals in the current soil, which are expressed by high-dimension data. Thus, to explore the complicated patterns of various heavy metals requires novel computing methods.

Clustering, as a fundamental approach to pattern mining, divides data into several groups based on data similarity; hence, data in the same group are more similar than data in different groups [13]. It is widely used in various domains, such as text recognition and image processing [14–17]. Among various clustering algorithms, the Gaussian mixture model, as a generating method, captures each cluster by a probability distribution, which well fit multiview characteristics of data in a natural manner [18]. Inspired by this, a Gaussian mixture model is introduced to mine the multiview heavy metal data. However, the current Gaussian mixture model-based methods neglect the multiview information of data, especially the deep intrinsic fusion features of all views.

To solve those challenges, in this paper, a multiview Gaussian mixture model is proposed to naturally capture complicated relationships over multiviews on the basis of deep fusion features of data, which can potentially mine robust patterns of heavy metals in practice. In particular, a deep fusion feature architecture with modality-specific and modality-common stacked autoencoders is designed to distill fusion representations from the information of all views. Then, the Gaussian mixture model is extended on the fusion representations to naturally recognize the accurate patterns of the intra- and inter-views. Extensive experiments are conducted on the representative datasets to evaluate the performance of the multiview Gaussian mixture model. Results show that the proposed method can greatly outperform the compared methods.

Thus, the major contributions of this paper are threefold:

(i) To accurately capture complex patterns of heavy metal data, a multiview Gaussian mixture model is introduced based on the fusion representations, which fully considers information of each view in a nonlinear manner

(ii) To distill fusion representations from the information of all views, a deep fusion feature architecture is designed, which consists of modality-specific and modality-common stacked autoencoders

(iii) Extensive experiments with outperforming results are conducted to assess the performance on the representative datasets

The rest of the paper is organized as follows. Section 2 reviews common methods in statistical learning about the pattern mining of heavy metals. Sections 3 and 4 are the fundamentals of the proposed method. Section 5 describes the details of the proposed method, and Section 6 validates the proposed method. Finally, Section 7 concludes this work.

2. Related Works

To trace the source of the soil heavy metal pollution, a lot of statistical methods were proposed. Most of all can be grouped into the following:

Linear regression. Because of its simplicity and efficiency, linear regression is a frequently used method [19]. It tries to find the best linear projection function through updating the parameters of the function using the least square method or the gradient descent method. For example, Tian et al. [20] improved the multiple linear regression (MLR) method to quantitatively estimate relationships between soil properties and sources of heavy metals. In MLR, heavy metal concentrations were regarded as dependent variables while the scores of soil properties and sources were independent variables. However, due to the influence of various complex factors, such as climate, parent material, topography, and human activities, the linear projection cannot well model correlations between the environmental parameters and the soil properties in the practice of soil pollution research [21].

Decision tree. Decision tree methods such as classification and regression tree (CART) and random forest (RF) use a tree structure for deciding classification results by judging from the root to leaves [22, 23]. For example, Qiu et al. [24] applied stepwise linear regression (SLR), CART, and RF to the prediction of the soil Cd's spatial distribution. In that article, RF was the best method for handling the nonlinear and hierarchical relationships between soil Cd and influence factors. Wang et al. [25] aimed to use RF and the stochastic gradient boosting (SGB) method for identifying and apportioning heavy metal pollution. Both RF and SGB showed that the biggest reason for the concentrations of Pb and Cd was anthropogenic sources.

Neural network. The neural network imitates the mechanism of human brains, recombining the information of input to extract some simple and fuzzy features, producing the corresponding impression and judgment. Furthermore, nonlinear activation functions of each layer, such as the sigmoid function and Rectified Linear Unit (ReLU) function, play a great role in the nonlinear fitting ability. One representative work is [26]. Specifically, neural networks with Monte Carlo simulations are combined to address the uncertainties from data quality and measurement errors in predicting the copper's phytoavailability in contaminated soils against the soil input parameters.

Principal component analysis (PCA). The principal component analysis uses the covariance matrix of data matrix for choosing principal components of data so that it can eliminate the less important properties for reducing the dimension of data and extracting hidden subsets to detect possible sources. For surveying the Chinese farmland soil metal accumulation at the national scale, Niu et al. [27] performed multivariate statistical analysis on soil properties and metal concentrations using PCA and correlation analysis. Research results on 11 metals showed that Pb, Cd, Zn, and

Cu had the concentrations above reference values. At the same time, results indicated that the 4 metals' accumulation may be associated with artificial fertilization. Also, Sun et al. [28] used PCA and correlation coefficient analysis to mine the agricultural soil major and trace element accumulation in the Gannan area, China. More PCA-based research includes [29, 30].

Cluster analysis (CA). CA classifies the data points into several disjoint and nonempty clusters on the basis of the similarity or distance among data points. There are various clustering algorithms used in the heavy metal analysis, such as spectral clustering, K-means, and hierarchical clustering. For the characterization of heavy metals in soils, Chai et al. [31] performed PCA and clustering analysis on data from the surface and underlying horizons of grassland. Three principal components were extracted, and hierarchical clustering proved this result. Moreover, in the three clusters from hierarchical clustering, clusters 1 and 2 were merged at a higher level so that the heavy metals in clusters 1 and 2 had a similar source. Similarly, Liu et al. [32] applied PCA and clustering analysis on data from the outskirts of Changchun, China. Results showed that Pb, Cu, and Zn were from human activities, while Cr and Ni were from natural sources.

In summary, the above methods can mine patterns of heavy metals in soil in simple cases where there are not many kinds of heavy metals. However, they neglect the multiview characteristics of land data, leading to undesired result patterns in complicated cases. Also, those methods cannot capture intrinsic patterns within high-dimension representations of land data. To solve those challenges, a deep fusion Gaussian mixture model for multiview land data clustering is proposed in this paper.

3. The Deep Stacked Autoencoder

The deep stacked autoencoder is a neural network of the fully connected paradigm on the basis of autoencoders, as shown in Figure 1 [33–35]. It extracts instinct representations of data by data reconstruction between an encoder and a decoder where the encoder constructs deeper representations layer by layer with the decoder reconstructing the input [36–38]. The deep stacked autoencoder is trained by a greedy layer-wise method in which each layer in the encoder and the corresponding layer of the decoder are modeled as an autoencoder to obtain the pretrained parameters followed by an end-to-end fine-tuning training.

Specifically, in a deep stacked autocoder of l layers, the s -th layer is modeled as an autoencoder with the $(l-s+1)$-th layer to pretrain weights and biases in the following form:

$$\begin{aligned} h^s &= f\left(w^s \odot h^{s-1} + b^s\right), \\ h^{l-s+1} &= f\left(w^{l-s+1} \odot h^s + b^{l-s+1}\right), \end{aligned} \tag{1}$$

where w^s, w^{l-s+1}, b^s, and b^{l-s+1} are the weights and biases of the s-th layer and the $(l-s+1)$-th layer, respectively. $\odot$ is the matrix product. h denotes the hidden representation.

After the pretraining, each hidden layer in the deep stacked autocoder is fine-tuned as follows:

$$h^s = f\left(w^s \odot h^{s-1} + b^s\right), \tag{2}$$

which is based on the stochastic gradient descent algorithm.

4. The Gaussian Mixture Model

A Gaussian mixture model (GMM) is a generative probabilistic model with trainable parameters [16]. It uses several basis Gaussian components to naturally represent multimodal characteristics of collected data by a weighted superposition operation, where each Gaussian component denotes a modal source. Generally, the Gaussian mixture model is trained by the expectation-maximization method by maximizing the likelihood function, where the expectation step computes probability distributions of each sample generated from each basis component and the maximization step learns the mean, covariance, and weight parameters of each basis component. GMMs have been widely used in various applications, such as text clustering and image recognition.

Given a dataset $X = \{x_1, x_2, \cdots, x_N\}$ with $x_i \in R^d$, the Gaussian mixture distributions are denoted as

$$p(x_i) = \sum_{k=1}^{K} w_k g(x_i; \mu_k, \Sigma_k), \tag{3}$$

where w_k is the weight of each basis Gaussian component and $g(x_i; \mu_k, \Sigma_k)$ represents the basis distribution parameterized by the mean vector μ_k and the covariance matrix Σ_k with the following form:

$$g(x_i; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)\right\}. \tag{4}$$

The d is the dimension of data, and K is the number of basis Gaussian components.

Thus, to fit the given dataset $X = \{x_1, x_2, \cdots, x_N\}$, the logarithm likelihood function of GMM is expressed in the following form:

$$\begin{aligned} \log L &= \log\left(\prod_{i=1}^{N} p(x_i, z_i)\right) \\ &= \log\left(\prod_{i=1}^{N} \prod_{k=1}^{K} \left(w_k g(x_i; \mu_k, \Sigma_k)\right)^{z_{ik}}\right) \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} \left(z_{ik} \log\left(w_k\right) + z_{ik} \log\left(g(x_i; \mu_k, \Sigma_k)\right)\right), \end{aligned} \tag{5}$$

where $z_i \in \{0, 1\}^K$, $\sum_{k=0}^{K} z_{ik} = 1$, denotes the component from which x_i is generated. Then, setting the derivates of $\log L$ to

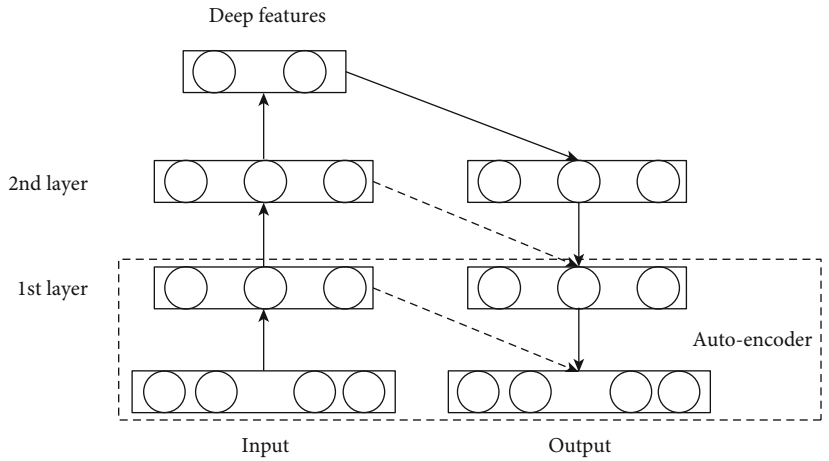


Figure 1: The computing paradigm of the stacked autoencoder.

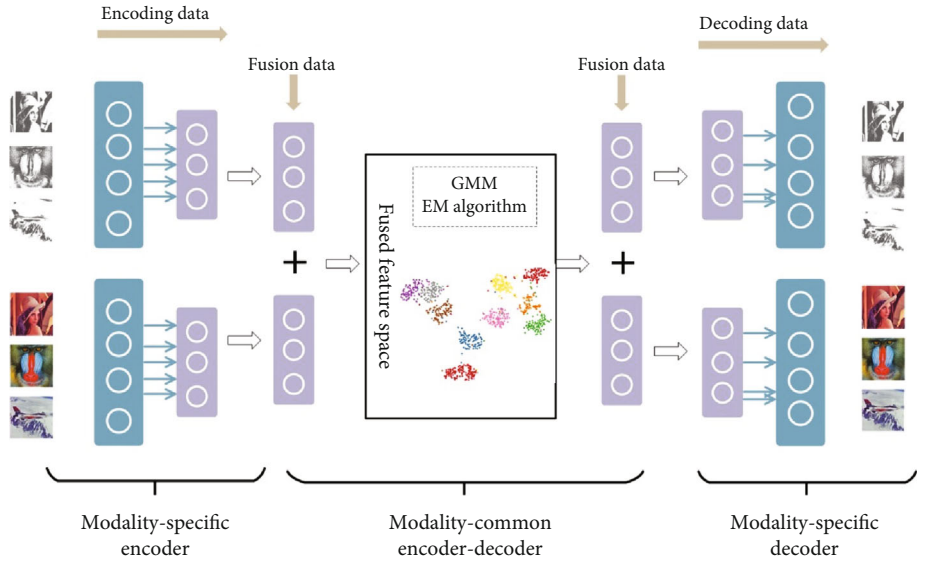


Figure 2: The computing paradigm of the multiview fusion Gaussian mixture model. Modality-specific encoders, modality-common encoder-decoder, and modality-specific decoders are linked in a cascaded manner where data are transferred into hidden representations of each view by modality-specific encoders; then, those hidden representations are concentrated, which are reconstructed via the modality-common encoder-decoder, and finally, the reconstructed hidden representations are decoded into the original data space by modality-specific decoders.

be zero, we can get the computing equations of the mean, covariance, and weight parameters of each basis component.

$$w_k = \frac{1}{N}\sum_{i=1}^{N} r_{ik},$$
$$\mu_k = \frac{\sum_{i=1}^{N} r_{ik} x_i}{\sum_{i=1}^{N} r_{ik}}, \tag{6}$$
$$\Sigma_k = \frac{\sum_{i=1}^{N} r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} r_{ik}},$$

in which

$$r_{ik} = p(z_{ik} = 1 \mid x_i; w_k, \mu_k, \Sigma_k) = \frac{w_k g(x_i; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} w_k g(x_i; \mu_k, \Sigma_k)}. \tag{7}$$

Generally, the expectation-maximization method is used to train GMM in an iterative maximization manner where current parameters are employed to estimate future parameters.

5. The Multiview Fusion Gaussian Mixture Model Algorithm

To mine complicated fusion relationships over multiview data, a deep fusion representation-based Gaussian mixture model is proposed, which is composed of the deep fusion feature learning and the expectation-maximization clustering. In the deep fusion feature learning, intrinsic view-specific features are first extracted by each view-specific stacked autoencoder. Then, those view-specific features are concentrated via a view-common stacked autoencoder, capturing fusion representations of multiview data. In the expectation-

maximization clustering, the Gaussian mixture model is used to recognize structure patterns of complicated shapes.

5.1. The Deep Fusion Feature Learning. To obtain the effective representations of multiview data, a deep fusion architecture is designed on the basis of the unsupervised encode-decode manner, which can avoid the dimensionality curse of data. As shown in Figure 2, in the deep fusion architecture, all the views of data are simultaneously fed into the corresponding view-specific stacked autoencoders, learning intrinsic view-specific features.

In detail, given the multiview dataset $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ in which each sample $\mathbf{x}_i$ is composed of v views $\mathbf{x}_i = (x_i^1, x_i^2, \cdots, x_i^v)$, each sample $\mathbf{x}_i$ is mapped to the view-specific feature space as follows:

$$\begin{aligned} h_i^1 &= f_l^1\left(f_{l-1}^1\left(\cdots\left(f_1^1\left(x_i^1\right)\right)\right)\right), \\ &\vdots \\ h_i^j &= f_l^j\left(f_{l-1}^j\left(\cdots\left(f_1^j\left(x_i^j\right)\right)\right)\right), \\ &\vdots \\ h_i^v &= f_l^v\left(f_{l-1}^v\left(\cdots\left(f_1^v\left(x_i^v\right)\right)\right)\right), \end{aligned} \tag{8}$$

where h_i^j is the feature of the j-th view and $f_l^j(f_{l-1}^j(\cdots(f_1^j())))$ is the corresponding encoding network function with the trainable parameters $w_l{}^j, \cdots, w_1{}^j$ and $b_l{}^j, \cdots, b_1{}^j$. To train those parameters, the features of all views are mapped to original data space as follows:

$$\begin{aligned} x_i^1 &= g_l^1\left(g_{l-1}^1\left(\cdots\left(g_1^1\left(h_i^1\right)\right)\right)\right), \\ &\vdots \\ x_i^j &= g_l^j\left(g_{l-1}^j\left(\cdots\left(g_1^j\left(h_i^j\right)\right)\right)\right), \\ &\vdots \\ x_i^v &= g_l^v\left(g_{l-1}^v\left(\cdots\left(g_1^v\left(h_i^v\right)\right)\right)\right), \end{aligned} \tag{9}$$

where $g_l^j(g_{l-1}^j(\cdots(g_1^j())))$ denotes the decoding network function. The view-specific encoder is cascaded by the corresponding decoder to get the pretrained weights and biases with the help of the stochastic gradient descent algorithm via the end-to-end training.

After the view-specific intrinsic representations $\{h_i^1, h_i^2, \cdots, h_i^v\}$ are obtained; they are concentrated in the following form:

$$\mathbf{h}_i = \mathrm{con}\left(h_i^1, h_i^2, \cdots, h_i^v\right), \tag{10}$$

where con() is the linear concentration function. Then, a view-common stacked autoencoder is used to transfer the concentrated representations to a fusion feature space, learning fused representations of multiview data via

$$\begin{aligned} \mathbf{h}_i^{\text{fusion}} &= \mathrm{encoder}(\mathbf{h}_i), \\ \mathbf{h}_i &= \mathrm{decoder}\left(\mathbf{h}_i^{\text{fusion}}\right), \end{aligned} \tag{11}$$

in which encoder() and decoder() are deep neural networks with the same number of layers.

5.2. The Clustering Pattern Mining. Specifically, after obtaining the fusion representations of the multiview dataset $\{f_1, f_2, \cdots, f_N\}$, the Gaussian mixture model with K basis components is defined as follows:

$$p(f_i) = \sum_{k=1}^{K} w_k g(f_i; \mu_k, \Sigma_k), \tag{12}$$

where w_k denotes the weight of the k-th basis Gaussian model, f_i represents the i-th fusion representation, and $g(f_i; \mu_k, \Sigma_k)$ is the basis distribution parameterized by the mean vector μ_k and the covariance matrix Σ_k with the following form:

$$g(f_i; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(f_i - \mu_k)^T \Sigma_k^{-1}(f_i - \mu_k)\right\}. \tag{13}$$

The d is the dimension of fusion representations of data.

Thus, the logarithm likelihood function of the given data is expressed in the following form:

$$\begin{aligned} \log L &= \log\left(\prod_{i=1}^{N} p(f_i, z_i)\right) \\ &= \log\left(\prod_{i=1}^{N}\prod_{k=1}^{K}\left(w_k g(f_i; \mu_k, \Sigma_k)\right)^{z_{ik}}\right) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K}\left(z_{ik}\log\left(w_k\right) + z_{ik}\log\left(g(f_i; \mu_k, \Sigma_k)\right)\right), \end{aligned} \tag{14}$$

where $z_i \in \{0, 1\}^K$, $\sum_{k=0}^{K} z_{ik} = 1$, denotes the component from which f_i is generated.

Then, setting the derivates of $\log L$ to be zero, we can get the computing equations of the mean, covariance, and weight parameters of each basis component.

$$\begin{aligned} w_k &= \frac{1}{N}\sum_{i=1}^{N} r_{ik}, \\ \mu_k &= \frac{\sum_{i=1}^{N} r_{ik} f_i}{\sum_{i=1}^{N} r_{ik}}, \\ \Sigma_k &= \frac{\sum_{i=1}^{N} r_{ik}(f_i - \mu_k)(f_i - \mu_k)^T}{\sum_{i=1}^{N} r_{ik}}, \end{aligned} \tag{15}$$

The multiview fusion Gaussian mixture model algorithm.
Input: the multiview dataset $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, the number of component models K, the hyperparameters of deep fusion architecture
Output: patterns of the input data
1. To randomly initialize parameters of each autoencoder in the deep fusion architecture;
2. To train each autoencoder layer by layer;
3. To fine-tune the deep fusion architecture in an end-to-end manner;
4. To randomly initialize model parameters and weight coefficients of Gaussian models;
5. To compute the probability of each sample generated from each Gaussian model;
6. To compute the model parameters and weight coefficients of each Gaussian model;
7. To update model parameters and weight coefficients of Gaussian models;
8. Go to 5 until convergence, then output the probability of each data sample generated from each Gaussian model as patterns of the input data.

Algorithm 1

Table 1: ARI results of models on the basis of raw representations.

Models	K-means-M	GMM-M
ARI	0.36	0.24

Table 2: NMI results of models on the basis of raw representations.

Models	K-means-M	GMM-M
NMI	0.49	0.37

in which

$$r_{ik} = p(z_{ik} = 1 \mid f_i; w_k, \mu_k, \Sigma_k) = \frac{w_k g(f_i; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} w_k g(f_i; \mu_k, \Sigma_k)}. \quad (16)$$

5.3. The Multiview Fusion Gaussian Mixture Model Algorithm. The multiview fusion Gaussian mixture model algorithm consists of two steps, i.e., fusion feature learning and pattern mining. In the former step, all view-specific stacked autoencoders and view-common stacked autoencoders are trained in a greedy layer-wise unsupervised manner. Then, an end-to-end fine-tuning training is conducted on the basis of SGD. In the latter step, the fusion features of multiview data extracted in the former step are fed into the multiview Gaussian mixture model with the predefined K. Then, the parameters in each component Gaussian model and weight coefficients between Gaussian models are learned based on the expectation-maximization algorithm. The details of the multiview fusion Gaussian mixture model algorithm are shown in Algorithm 1.

6. Experiments

To evaluate the performance of the multiview fusion Gaussian mixture model, extensive experiments are conducted on two datasets. Those experiments are implemented by Python, and the details of the experiments are described in the following.

Table 3: ARI results of models on the basis of deep representations.

Models	K-means-DM	GMM-DM	K-means-DE	GMM-DE	Clustering-DF
ARI	0.65	0.76	0.57	0.74	0.80

Table 4: NMI results of models on the basis of deep representations.

Models	K-means-DM	GMM-DM	K-means-DE	GMM-DE	Clustering-DF
NMI	0.71	0.81	0.62	0.80	0.85

6.1. Compared Methods. K -means. K-means is a typical clustering method that is widely used in practice as a representative baseline.

Gaussian mixture model. The Gaussian mixture model is a generative method based on the probability distribution. It mines cluster patterns of data by multiple Gaussian distributions.

In the experiments, the K-means and Gaussian mixture model are used as the based model, which are extended to modality-specific, modality-common, modality-fused methods with respect to raw, shallow, and deep representations of data.

6.2. Datasets. MNIST-EMNIST. MNIST [39] and EMNIST [40] are the representative datasets of images, which contain images of numbers from 0 to 9. They are widely used in image classification and image clustering. In the experiments, MNIST and EMNIST are fed into a fully connected neural network and a convolutional neural network, respectively, in feature learning to represent different views. The results are illustrated in Tables 1–4. Also, Figure 3 visualizes the feature learning processing.

6.3. Results. In the results of Tables 1–4, K-means-M and GMM-M are the traditional K-means and GMM clustering algorithms conducted on the raw representations of MNIST. K-means-DM and GMM-DM denote the K-means and GMM performed on the deep representations of MNIST, which is extracted by the modality-specific stacked autoencoder. K-means-DE and GMM-DE are similar models

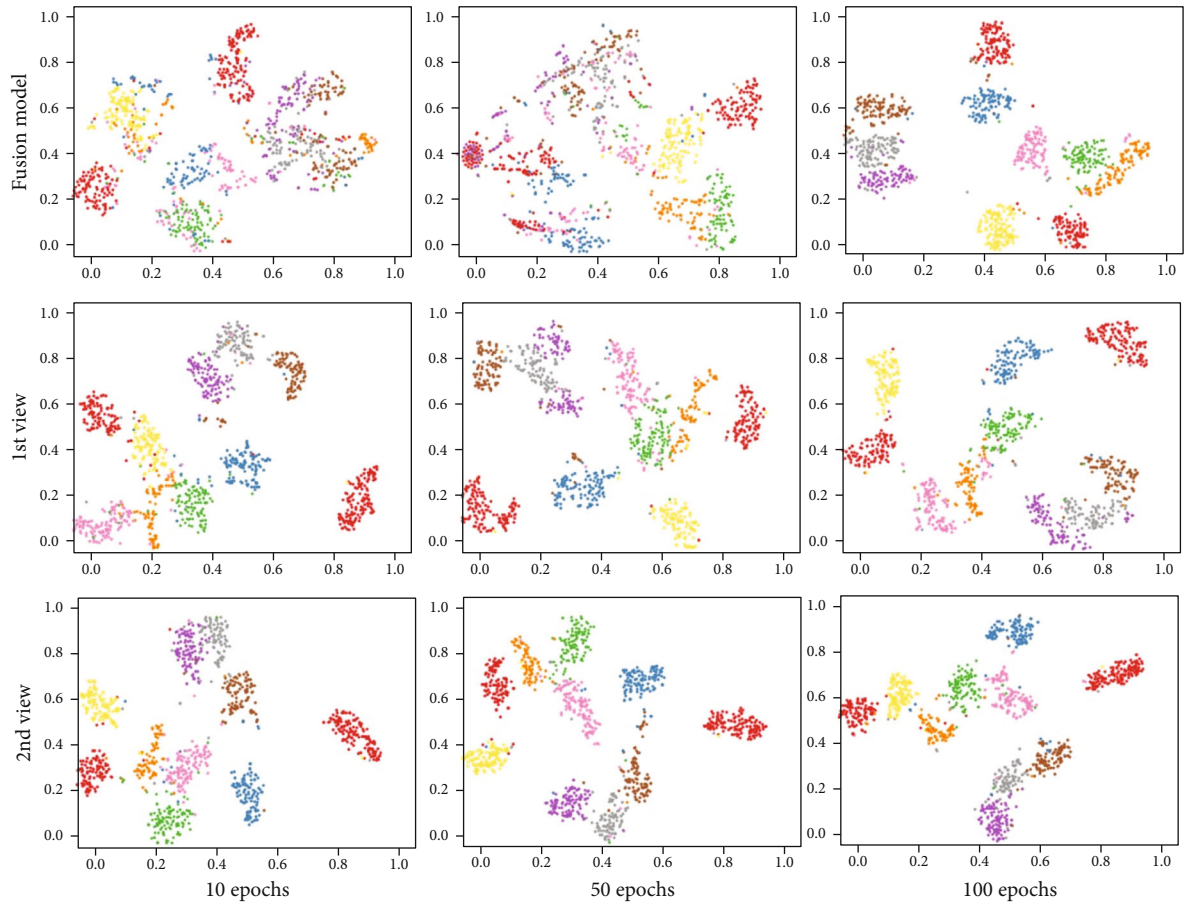


Figure 3: The t-SNE figures of each view model and the fusion model.

performed on EMNIST. Clustering-DF denotes the results of the proposed model.

From the above results, several observations can be concluded. In raw representations of data, K-means produced better results than GMM in terms of ARI and NMI. This is because the less important properties in raw representations are also modeled by the probability distributions of GMM, decreasing the clustering performance. The second observation is that the deep feature-based methods (K-means-DM, GMM-DM) outperform the shallow methods (K-means-M, GMM-M), since the proposed modality-specific stacked autoencoder can well extract intrinsic features of each view of data. Additionally, the clustering results of GMM-DM are better than those of K-means-DM, since the multiple Gaussian distributions in GMM can better fit patterns of data than the hard division in K-means with the clear features. The third observation is that the proposed method achieves the best results in terms of ARI and NMI, since it can distill information from all views by the designed deep fusion network. The observations of the results demonstrate the outperformance of the proposed method.

Figure 3 shows the t-SNE figures of the above models to visualize features learned by each model. There are two observations. First, the fusion model learns better representations than each single-view model. Specifically, the proposed model produces features where the distance of similar data is closer than that of dissimilar data, shown in the third column. Furthermore, the distance between different clusters is further. Second, the proposed model learns data representations faster than single-view models. In detail, the representations produced by the fusion model are more disorderly than those by the compared models at the beginning, while the fusion model achieves better representations after the same number of training epochs.

7. Conclusions

In this paper, a deep fusion Gaussian mixture model is proposed for multiview data clustering based on deep fusion representations, which can potentially capture intrinsic patterns of heavy metal data. In this model, a deep fusion feature architecture of modality-specific and modality-common stacked autoencoders is designed to merge fusion information of all views of data, which can well capture deep intrinsic fusion representations of data. Afterward, the Gaussian mixture model is extended on the fusion representations to naturally recognize the accurate patterns. Finally, results show the outperformance of the proposed methods by extensive experiments. In the future, more effective deep clustering methods will be explored, which are trained in an end-to-end manner.

Data Availability

The datasets used in this paper are public datasets which can be accessed by the following websites: MNIST and EMNIST (https://pytorch.org/docs/stable/torchvision/datasets.html).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Authors' Contributions

Peng Li, Jing Gao, Jianing Zhang, Shan Jin, and Wenhan Zhao are the first coauthors.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant No. 2016YFD0800300.

References

[1] X. Yang, L. Geng, and K. Zhou, "Environmental pollution, income growth, and subjective well-being: regional and individual evidence from China," *Environmental Science and Pollution Research*, vol. 27, no. 27, pp. 34211–34222, 2020.

[2] X. Zhao, Y. Sun, J. Huang, H. Wang, and D. Tang, "Effects of soil heavy metal pollution on microbial activities and community diversity in different land use types in mining areas," *Environmental Science and Pollution Research*, vol. 27, no. 16, pp. 20215–20226, 2020.

[3] R. Vamanan and K. Ramar, "Classification of agricultural land soils a data mining approach," *International Journal of Computer Science and Engineering*, vol. 3, no. 1, pp. 379–384, 2011.

[4] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[5] Z. Ning, K. Zhang, X. Wang, L. Guo, and R. Y. K. Kwok, "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[6] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 2020, pp. 1–16, 2020.

[7] Z. Ning, P. Dong, X. Wang et al., "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[8] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 411–425, 2021.

[9] T. Chen, Q. Chang, J. Liu, J. G. P. W. Clevers, and L. Kooistra, "Identification of soil heavy metal sources and improvement in spatial mapping based on soil spectral information: a case study in northwest China," *Science of the Total Environment*, vol. 565, pp. 155–164, 2016.

[10] G. Shi, J. Liu, H. Wang et al., "Source apportionment for fine particulate matter in a Chinese city using an improved gas-constrained method and comparison with multiple receptor models," *Environmental Pollution*, vol. 233, pp. 1058–1067, 2018.

[11] S. Jain, S. K. Sharma, T. K. Mandal, and M. Saxena, "Source apportionment of PM10 in Delhi, India using PCA/APCs, UNMIX and PMF," *Particuology*, vol. 37, pp. 107–118, 2018.

[12] K. Keerthi, N. Selvaraju, and L. A. Varghese, "Use of combined receptor modeling technique for prediction of possible sources of particulate pollution in Kozhikode, India," *International Journal of Environmental Science and Technology*, vol. 17, no. 5, pp. 2623–2636, 2020.

[13] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: from the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.

[14] M. Ibrar, J. Mi, S. Karim, A. A. Laghari, S. M. Shaikh, and V. Kumar, "Improvement of large-vehicle detection and monitoring on CPEC route," *3d Research*, vol. 9, no. 3, article 45, 2018.

[15] S. Karim, Y. Zhang, S. Yin, A. A. Laghari, and A. A. Brohi, "Impact of compressed and down-scaled training images on vehicle detection in remote sensing imagery," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 32565–32583, 2019.

[16] S. Karim, I. A. Halepoto, A. Manzoor, N. H. Phulpoto, and A. A. Laghari, "Vehicle detection in satellite imagery using maximally stable extremal regions," *International Journal of Computer Science and Network Security*, vol. 18, no. 4, 2018.

[17] A. A. Laghari, H. He, M. Shafiq, and A. Khan, "Assessment of quality of experience (QoE) of image compression in social cloud computing," *Multiagent and Grid Systems*, vol. 14, no. 2, pp. 125–143, 2018.

[18] C. E. Rasmussen, "The infinite Gaussian mixture model," *Advances in Neural Information Processing Systems*, vol. 12, pp. 554–560, 2000.

[19] J. A. Thompson, E. M. Pena-Yewtukhiw, and J. H. Grove, "Soil–landscape modeling across a physiographic region: topographic patterns and model transportability," *Geoderma*, vol. 133, no. 1-2, pp. 57–70, 2006.

[20] K. Tian, W. Hu, Z. Xing, B. Huang, M. Jia, and M. Wan, "Determination and evaluation of heavy metals in soils under two different greenhouse vegetable production systems in eastern China," *Chemosphere*, vol. 165, pp. 555–563, 2016.

[21] X. Zhang, F. Lin, Y. Jiang, K. Wang, and M. T. F. Wong, "Assessing soil Cu content and anthropogenic influences using decision tree analysis," *Environmental Pollution*, vol. 156, no. 3, pp. 1260–1267, 2008.

[22] G. De'ath and K. E. Fabricius, "Classification and regression trees: a powerful yet simple technique for ecological data analysis," *Ecology*, vol. 81, no. 11, pp. 3178–3192, 2000.

[23] J. M. Drake, C. Randin, and A. Guisan, "Modelling ecological niches with support vector machines," *Journal of Applied Ecology*, vol. 43, no. 3, pp. 424–432, 2006.

[24] L. Qiu, K. Wang, W. Long, K. Wang, W. Hu, and G. S. Amable, "A comparative assessment of the influences of human impacts on soil cd concentrations based on stepwise linear regression, classification and regression tree, and random forest models," *PLoS One*, vol. 11, no. 3, article e0151131, 2016.

[25] Q. Wang, Z. Xie, and F. Li, "Using ensemble models to identify and apportion heavy metal pollution sources in agricultural soils on a local scale," *Environmental Pollution*, vol. 206, pp. 227–235, 2015.

[26] N. Hattab, R. Hambli, M. Motelica-Heino, and M. Mench, "Neural network and Monte Carlo simulation approach to investigate variability of copper concentration in phytoremediated contaminated soils," *Journal of Environmental Management*, vol. 129, pp. 134–142, 2013.

[27] L. Niu, F. Yang, C. Xu, H. Yang, and W. Liu, "Status of metal accumulation in farmland soils across China: from distribution to risk assessment," *Environmental Pollution*, vol. 176, pp. 55–62, 2013.

[28] G. Sun, Y. Chen, X. Bi et al., "Geochemical assessment of agricultural soil: a case study in Songnen-Plain (Northeastern China)," *Catena*, vol. 111, pp. 56–63, 2013.

[29] Y. Shan, M. Tysklind, F. Hao, W. Ouyang, S. Chen, and C. Lin, "Identification of sources of heavy metals in agricultural soils using multivariate analysis and GIS," *Journal of Soils and Sediments*, vol. 13, no. 4, pp. 720–729, 2013.

[30] Y. Li, H. Gao, L. Mo, Y. Kong, and I. Lou, "Quantitative assessment and source apportionment of metal pollution in soil along Chao River," *Desalination and Water Treatment*, vol. 51, no. 19-21, pp. 4010–4018, 2013.

[31] Y. Chai, J. Guo, S. Chai, J. Cai, L. Xue, and Q. Zhang, "Source identification of eight heavy metals in grassland soils by multivariate analysis from the Baicheng–Songyuan area, Jilin Province, Northeast China," *Chemosphere*, vol. 134, pp. 67–75, 2015.

[32] L. Qiang, L. Jingshuang, W. Qicun, and W. Yang, "Source identification and availability of heavy metals in peri-urban vegetable soils: a case study from China," *Human and Ecological Risk Assessment*, vol. 22, no. 1, pp. 1–14, 2016.

[33] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 1, pp. 1–36, 2020.

[34] J. Gao, P. Li, and Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in internet of things," *Future Generation Computer Systems*, vol. 99, pp. 508–516, 2019.

[35] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.

[36] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[37] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An improved stacked auto-encoder for network traffic flow classification," *IEEE Network*, vol. 32, no. 6, pp. 22–27, 2018.

[38] Z. Ning, R. Y. K. Kwok, K. Zhang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[39] 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.

[40] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," 2017, https://arxiv.org/abs/1702.05373.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8891595, 10 pages
https://doi.org/10.1155/2020/8891595

Research Article

Aero Engine Gas-Path Fault Diagnose Based on Multimodal Deep Neural Networks

Liang Zhao,[1,2] Chunyang Mo,[1,2] Tingting Sun,[3] and Wei Huang[4]

[1]*School of Software Technology, Dalian University of Technology, Dalian 116600, China*
[2]*Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116600, China*
[3]*Department of Natural Resources Information, Dalian Natural Resources Affairs Service Center, Dalian 116011, China*
[4]*First Affiliated Hospital of Dalian Medical University, Dalian 116000, China*

Correspondence should be addressed to Wei Huang; huangwei9898@163.com

Received 9 July 2020; Revised 5 August 2020; Accepted 21 September 2020; Published 6 October 2020

Academic Editor: Xiaojie Wang

Aeroengine, served by gas turbine, is a highly sophisticated system. It is a hard task to analyze the location and cause of gas-path faults by computational-fluid-dynamics software or thermodynamic functions. Thus, artificial intelligence technologies rather than traditional thermodynamics methods are widely used to tackle this problem. Among them, methods based on neural networks, such as CNN and BPNN, cannot only obtain high classification accuracy but also favorably adapt to aeroengine data of various specifications. CNN has superior ability to extract and learn the attributes hiding in properties, whereas BPNN can keep eyesight on fitting the real distribution of original sample data. Inspired by them, this paper proposes a multimodal method that integrates the classification ability of these two excellent models, so that complementary information can be identified to improve the accuracy of diagnosis results. Experiments on several UCR time series datasets and aeroengine fault datasets show that the proposed model has more promising and robust performance compared to the typical and the state-of-the-art methods.

1. Introduction

Aeroengine is known as "the pearl on the crown of industry" because of the irreplaceable roles it plays in industry and the highly sophisticated internal structure it has. The fault diagnosis technology of aeroengine is vitally important to guarantee its performance and efficiency and reduce the maintenance cost, which can accurately diagnose and locate the fault of aeroengine and provide powerful support for engine maintenance. Therefore, to a certain extent, the efficiency decline and performance instability of engine caused by component fault may become predictable. On the other hand, the complexity of aeroengine can be expressed by the complex nonlinear relationship among thermodynamic parameters of its components. The fault and degradation of aeroengine components can reflect in thermodynamic parameters such as temperature and pressure. However, it is difficult to reverse this possess, which means to deduce the causes and locations of faults by traditional computational fluid dynamics (CFD) software tools or by thermodynamics and fluid mechanics partial differential equation. Thus, most of the existing fault diagnosis technologies are realized by big data and artificial intelligence methods, which do not analyze why fault occurs but utilize the features or distribution of data to identify fault modes.

The existing aeroengine fault diagnosis methods can be roughly summarized as the following categories. First, techniques based on signal processing, such as the applications of Wavelet transform, Fourier transform, and Kalman filter. These methods extract features from the continuous signal waveform and present diagnosis results by analyzing them [1]. However, they are only applicable to continuously sampled data. Second, classification or clustering algorithms are based on similarity and distance. They treat each dimension of sample data as a coordinate of multidimensional vectors and find boundaries or curves in feature space to split samples into different groups. Data used by these methods should have strong numerical differentiation. An example is SVM,

which has been optimized into various versions [2, 3] from the basic form [4] or used in combination with other models [5, 6], in order to better cope with this problem. K-means algorithm [7] based on Euclidean distance to cluster states of the engine belongs to this category, as well. Third, diagnosis methods by combining expert experience and mathematical models. Expert experience is usually expressed as a decision tree or table which generates branches according to the threshold of sample attributes or features. In addition, expert experience is usually strongly subjective, so its correctness will directly affect diagnosis results. Fourth, diagnosis models based on neural networks, such as CNN, BPNN, DBN [8], PNN [9], and other deep learning models. The first three groups of methods mentioned above work merely by separating data or their features, without carefully fitting the functional or causal relationship between sample data and its categories, while methods based on neural networks can tackle it.

Therefore, in recent years, the deep learning models, such as multilayer autoencoders, CNN, LSTM, DBN, and BPNN, have attracted more attentions in the field of aeroengine fault detection and repair. Where CNN has the ability of learning features and is a powerful vehicle in many fields including problems related to sequence data. Thus, it was brought into the field of fault diagnosis, aiming to distinguish samples by its features. BPNN is a basic neural network that has the ability to express complex multivariate functional relationship. Hence, it is competent to most classification and regression tasks in various fields [10]. Due to its widespread applicability, it is now often used as the baseline in this field, or as the classifier in combination with other models.

In summary, CNN has the talent in extracting features from input data and then classifies data by them, while BPNN focuses on fitting original sample data and has a wide range of applications. However, most of the existing fault diagnosis mechanisms are designed only based on one neural network, which does not take advantage of the characteristics of both CNN and BPNN. Therefore, this paper proposes a multimodal deep neural network diagnosis method based on the feature perception ability of CNN and the fitting ability of BPNN. Specifically, the BPNN is employed to fit the sample data distribution, and CNN is utilized to explore the features of the samples, so as to obtain the multimodal decision information learned from different angles [11]. After that, these decisions work as the information source of the evidence bodies of D-S evidence theory. Finally, with the fused multimodal information and decision-making rules, diagnosis results can be given. Moreover, we describe a method to construct the basic probability assignment of evidence body from the classification result of neural network and simplify the limitation of decision rules under practical application circumstance. Experiments on several UCR standard time series datasets and the aeroengine fault datasets prove that by integrating the complementary information of multimodal neural networks which have distinctive abilities and learn sample data from different angles, a high accuracy diagnosis model can be achieved.

The rest of the paper is organized as follows. Section 2 reviews the related work on deep learning and D-S evidence theory methods for aeroengine fault diagnosis. Section 3 presents the framework of the proposed model and the steps to implement it in details. In Section 4, experiments are carried out to validate the effectiveness of the proposed model. Finally, conclusions are drawn.

2. Related Work

This section will present how CNN, BPNN, and D-S evidence theory are employed for aeroengine fault diagnosis. Besides, we find that previous works ignored the diversity of aeroengine fault data, which may cause their methods hard to generalize across datasets. Referring to the shortcomings existing in previous models and the problem about datasets, we further summarize the advantages of our proposed model.

With its excellent automatic feature extraction ability, CNN success in many fields [12], including aeroengine fault diagnosis. It is used primarily to process images, videos, and other two-dimensional structured data. Then, in the field of natural language processing, convolution kernel is reshaped to one-dimensional to find the connections between words and sentences. Similarly, horizontal one-dimensional CNN is also employed for aeroengine fault diagnosis. For example, Jiang et al. [13] focused on the feature extraction ability of CNN and used it to cope with fault classification. Wang et al. [14] processed signals simply at first and then employed CNN to extract features automatically from them, avoiding the inconsistency of performance caused by traditional manual extraction of features. Some derived models of CNN have also immigrated from the initial field of image processing. As ResNet [15], it was modified into a model with supermultilayer convolution layer to find high-level features of fault, which can obtain a promising diagnosis result. Furthermore, there are also some methods to extract the features of sample data by CNN and then use other classification methods to distinguish them. For example, in [16], the output of CNN's second-to-last layer is regarded as features of samples, and the features are transferred to SVM to be classified.

BPNN, as one of the most common and widely used artificial neural networks [17, 18], has been applied to the classification problem of aeroengine fault diagnosis long time ago [19, 20]. In past decades, researchers have tried to combine it with other methods or to optimize its structural parameters using some optimization search algorithms, in order to yield better diagnosis results. For example, in [21], to cope with the difficulty of insufficient fault samples by reducing the scale of neural network, rough set theory was used to pretreat original data, and then, BPNN was employed for fault diagnosis using the optimal decision attributes. Similarly, the method which integrated BPNN with SVM was proposed in [5]. To tackle the disadvantages of BPNN in gas turbine fault diagnosis, Yuan et al. [22] attempted to use the Particle Swarm Optimization algorithm and Lenvenberg Marquardt algorithm to improve the performance. Besides, BPNN is also used as a basic element of other complex models, such as methods based on nested BPNN [23] and integrated BPNN [24] for aeroengine fault diagnosis.

D-S evidence theory is one of the common means in the area of information fusion. It can integrate multiple evidence

bodies which can be the predictions of different people, data of different sensors, and results of different classifiers. Various basic models have been served as the information source of evidence bodies, for instance, RBF, SVM, and BPNN [25]; RBF and BPNN [26]; and CNN and SVM [27]. Information source of evidence bodies can be different results given by the same type of basic classifier, as well. For example, Song [28] divided the multidimensional attributes of samples into serval groups and used each group to train a BPNN, creating a fusion diagnosis model of multiple BPNN. Wu et al. [29] presented an information fusion fault diagnosis method based on D-S theory using SVM. However, these means are not perfect, as they carry out D-S theory merely by incorporating homogeneous models, without taking multimodal information into account.

For aeroengine fault diagnosis, there exists a common problem: aeroengine big data has no standard format. They are diverse in quality, dimension, continuity, and order of magnitudes of the attributes. Some fault diagnosis models are sensitive to datasets and narrow in application scope. They may only have high accuracy for certain dataset. In [30], author took the influence of attribute dimensions (13 and 8, respectively) into account and found that this factor did have a great impact on the capability of the model. In [25, 31], SVM and BPNN, both as the baselines, had a conflict on whose classification accuracy is higher, because it depends on datasets. Therefore, the aeroengine fault diagnosis model cannot only be accuracy-oriented. And neural networks have strong generalization ability and certain robustness, so they can ease the problem to some extent. But some previous methods which execute D-S theory by combining deep learning models with nondeep learning models will still have that defect, since the nondeep learning model is more sensitive to datasets. This is the second point leaving to be desired for the implementation of D-S theory in field of aeroengine fault diagnosis.

Those composite models that use D-S theory have above two shortcomings, while the single modal ones which merely use CNN or BPNN do not consider information of the two excellent models at the same time, obtaining a lower diagnosis accuracy [32]. Therefore, we propose a multimodal neural network diagnosis model that learns from multiple angles. It does not only perceive features but fits the distribution of original data. Meanwhile, compared with other composite models, the proposed is composed of pure neural network. Thus, it has better robustness.

3. Diagnose Method Based on Multimodal Deep Neural Networks

The framework of our proposed method is shown in Figure 1. First, samples are cleaned and normalized, which is a common means to reduce the impact of various data specification. Then, CNN and BPNN are trained to the best with samples, and the scores they output are served as information sources. Finally, the D-S evidence theory is employed to coordinate the multimodal information in form of evidence bodies and generate the final decisions.

3.1. Basic Models. BPNN is a multilayer fully connected neural network trained by the back propagation algorithm (BP), which is established by simulating the indescribable complicated working process in the brain. Each layer in the network is composed of multiple artificial neuron cells. The weight matrix and bias in cells carry out linear transformation on its input data. Next, the transformation result is used as the net input of activation function, which then makes nonlinear mapping. Through the stacking of nonlinear mapping layer by layer and the adjustment of parameters in cells by training algorithms, BP neural network can be employed to fit complex multivariate nonlinear functions [33]. Therefore, it can be used to explore the functional or causal relationship between the overall distribution of property values and fault types.

Convolutional neural network is a feedforward neural network that consists of convolution layer, pooling layer, and fully connected layer. Compared with BPNN, CNN has characteristics of partial connection and weight sharing, which make its complexity reduces a lot. Convolution is often viewed as an effective means for feature extraction. The convolution filter used for feature extraction is a matrix in nature, which slides over input data and performs matrix operation with the data it covers, and thus, the calculation result can be regarded as the matching extent between the covered data and the feature to be extracted. Then, the result is added by bias and put into the activation function to exert a nonlinear transformation. The convolution filter is self-adaptive under the adjustment of the back propagation algorithm, so CNN has the so-called ability to automatically extract features. In the domain of aeroengine fault diagnosis, CNN is usually used to discover the features residing in attributes and establish a mapping between the fault types and the features of the properties.

The frequently used activation function mentioned before is the Rectified Linear Unit. The cross-entropy function is generally selected as the loss function for classification problems, and it can be reduced by gradient back propagation algorithm with the update of parameters in cells, as follows:

$$J(y, \hat{y} \mid W, b) = -y^T \log(\hat{y}), \tag{1}$$

$$\begin{aligned} \frac{\partial J(y,\hat{y})}{\partial W^{(L)}} &= \frac{\partial J(y,\hat{y})}{\partial z^{(L)}} \cdot \frac{\partial z^{(L)}}{\partial W^{(L)}} \\ \frac{\partial J(y,\hat{y})}{\partial b^{(L)}} &= \frac{\partial J(y,\hat{y})}{\partial z^{(L)}} \cdot \frac{\partial z^{(L)}}{\partial b^{(L)}} \end{aligned}, \tag{2}$$

$$\begin{aligned} W_{n+1}^{(L)} &= W_n^{(L)} - \alpha \cdot \frac{\partial J(y,\hat{y})}{\partial W_n^{(L)}} \\ b_{n+1}^{(L)} &= b_n^{(L)} - \alpha \cdot \frac{\partial J(y,\hat{y})}{\partial b_n^{(L)}} \end{aligned}, \tag{3}$$

Herein, $J(\cdot)$ represents the loss function. W and b are parameters in cells, representing the weight matrix and bias, respectively. y and $\hat{y}$ represent the real and predicted categories of sample, expressed in one-hot vector form.

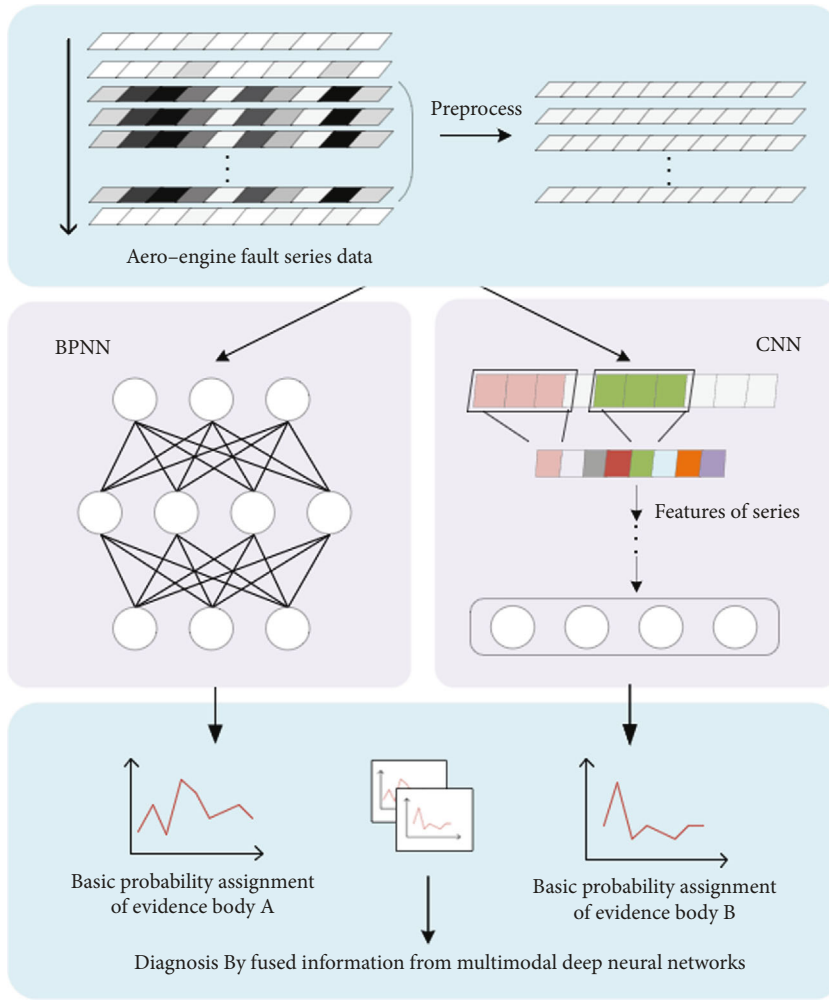


Figure 1: The architecture of multimodal deep neural networks diagnosis model.

The subscript n represents the parameter after the $n-th$ training, and L represents the $L-th$ layer. $Z = W^T x + b$ is the net input of activation function. α is the learning rate.

D-S evidence theory, another basic algorithm used in this paper, roughly includes three phases: constructing the basic probability assignment of each evidence body, fusion of evidence bodies, and giving final result according to the decision rules. In D-S evidence theory, the conclusion given by each evidence body about the hypothetic categories that samples may belong to is presented as a probability assignment. Since the decision information of each evidence body is not authoritative, the uncertainty of decision should be considered when constructing basic probability assignment from information source. In fusion phase, the basic probability assignments of different evidence bodies will be combined into a new fused one as output. Then, according to the decision rules, final classification results can be obtained. In theory, only when the difference between the maximum and the second largest values of the fused probability assignment is greater than a certain threshold, can the final classification decision be considered as an effective judgment, since the uncertainty information is involved in the probability calculation of every hypothesis. Otherwise, if they are close to each other, the given decision may be a misjudgment caused by uncertain information.

Many researchers have been engaged in the improvement of D-S theory, including how to scientifically construct the evidence bodies, improve the fusion formulas, and solve the problem of evidence conflict. Consequently, various specific formulas and rules to implement D-S theory are constructed. When applied, the specific implementary formulas used by different authors and in different scenes may vary greatly. In this paper, we design a new D-S theory-based multimodal fusion method.

3.2. The Proposed Method. Steps and details of the proposed model are as follows:

The first step is to preprocess the aeroengine sequence data, so that they can be directly input into the neural networks and can be better applied for feature perception and data fitting. Specifically, ignore the sequences with too many zero values at the beginning and end of datasets, and normalize the remaining samples by attributes according to formula (4):

$$X_{ij} = \frac{X_{ij} - \min\left\{X_{1j}, X_{2j}, \cdots, X_{Nj}\right\}}{\max\left\{X_{1j}, X_{2j}, \cdots, X_{Nj}\right\} - \min\left\{X_{1j}, X_{2j}, \cdots, X_{Nj}\right\}} \quad (i = 1, 2, \cdots, N. j = 1, 2, \cdots, d) \tag{4}$$

where X_{ij} represents the *j-th* attribute of the *i-th* sequence data in samples. N is the number of sequence data in samples. d represents the dimension of attributes. max $\{\cdot\}$ means the maximum value and min $\{\cdot\}$ means the minimum value.

Labels should be added one by one to samples with $1, 2, \cdots, m$ representing fault types $F_1, F_2, \cdots, F_m$. After that, merge and randomly shuffle all fault sequence data. On the one hand, for the problem of sequence classification, CNN and BPNN do not depend on the order relationship between samples. On the other hand, the shuffle can increase the randomness of samples, then train favorable models. At last, samples are divided into training set D, validation set V, and test set T.

The second step is to train CNN and BPNN. Thus, these two trained models are employed to diagnose the samples in validation set V and test dataset T, respectively. The scores they given before *SoftMax* will be used as the information source in the next step.

Generally, the number of fully connected layers in neural network should be less than or equal to three. Regarding the fault diagnosis of aeroengine, two or three layers should be set due to the complicated relationship between input data and types of faults. For time series data like aeroengine data, one-dimensional convolution should be carried out, and the size of convolution kernel should be smaller than the dimension of attributes. The training process uses the cross-entropy loss function and the back propagation algorithm given in formulas (1)–(3).

The third step is to convert multimodal information into the basic probability assignment of evidence bodies, with the selected coordination factor ρ, which will be discussed in the next subsection 3.3.

For each sequence data sample, its scores of belonging to different fault types before *SoftMax* given by BPNN and CNN should be converted into the initial probability assignments of evidence bodies. These can be calculated as

$$E_i^{M0}(l) = softmax(\rho \cdot r_i(l)) = \frac{\exp\left(\rho \cdot r_i(l)\right)}{\sum_{l=1}^{m} \exp(\rho \cdot r_i(l))}. \tag{5}$$

Herein, r represents the scores before *SoftMax* function, given by the neural network. ρ is the coordination factor of information source. E_i^{M0} represents the initial probability assignment of samples. The subscript i represents the *i-th* sequence data of samples. Superscript M represents an evidence body, denoted by $M = \{A : CNN, B : BPNN\}$. And $l = 1, 2, 3, \cdots, m$ refers to one hypothesis of the fault categories.

Thus, the initial probability assignment can be used to calculate the uncertainty measurement of samples, according to formulas (6) and (7). The sum of squares of the distance from the initial probability assignment of one evidence body to the average initial probability assignment of two evidence bodies is selected as the uncertainty measurement. This value measures the conflict between decisions for the same sample data given by two evidence bodies. The larger it is, the more inconsistent the two judgments are, which means the less determinate they are.

$$AVG\_E_i(l) = \frac{1}{2}\left[E_i^{A0}(l) + E_i^{B0}(l)\right], \tag{6}$$

$$D_i^M = \sum_{l=1}^{m}\left[E_i^{M0}(l) - AVG\_E_i(l)\right]^2. \tag{7}$$

In the formulas, AVG_E represents the average of the two evidence bodies. E^{A0} and E^{B0} represents the initial probability assignments of the two evidence bodies. D_i^M represents the sum of the squares of the distance from the result of the evidence body M to the average result. Then, the measurement can be converted into the uncertainty of the sample using formula (8). Thus, the basic probability assignment of evidence body can be calculated by formula (9) and (10), with initial probability assignment and uncertainty:

$$U_i^M = \sqrt{D_i^M}, \tag{8}$$

$$E_i^M(l) = E_i^{M0}(l) \times \left(1 - U_i^M\right), \tag{9}$$

$$E_i^M(\Theta) = U_i^M, \tag{10}$$

where $E_i^M(l)$ represents the belief probability of hypothesis l in basic probability assignment.$E_i^M(\Theta)$ represents the probability of belonging to the item of universal set Θ. U_i^M represents the uncertainty of sample under evidence body M.

The fourth step is to determine how to fuse the multimodal information which are expressed in form of evidence bodies and give the decision-making criteria.

The basic probability assignments of CNN and BPNN are used to calculate their counterpart after fusion, as shown in formulas ((11) and ((12):

$$E_i(l) = \frac{\sum_{k,h} E_i^A(k) \times E_i^B(h)}{1 - U_i} A_i(k) \cap B_i(h) = l, \tag{11}$$

$$U_i = \sum_{k,h} E_i^A(k) \times E_i^B(h) A_i(k) \cap B_i(h) = \varnothing \text{or} \Theta. \tag{12}$$

Herein, $k, h \in \{F_1, F_2 \cdots F_m\} \cup \{\Theta\}$ represent two hypotheses of assignment. $A_i(k), B_i(h)$ represent the hypotheses taken, respectively, from two evidence bodies for the same sequence. U_i represents the sample's uncertainty of assignment after fusion. $E_i(l)$ represents the belief that this sequence belongs to hypotheses l, according to the fused probability assignment.

Diagnose Method Based on Multimodal Deep Neural Networks.
Input: Aero engine gas-path fault datasets of m fault modes.
Output: Diagnosis result $Y$of sample sequence in test set T
1 Preprocess the input dataset: clean, normalize by formula (4), label, merge and shuffle it. Divide it into training set D ,validation set V and test set T.
2 **For** M **in** {A: CNN, B: BPNN}:
3 **Repeat:**
4 Adjust one hyper-parameter, including layers, learning rate, etc.
5 **For** i <maxiterations:
6 Calculate the value of cross-entropy loss.
7 Update parameters by **BP algorithm**: $W_i^{(l)} \leftarrow W_{i-1}^{(l)}, b_i^{(l)} \leftarrow b_{i-1}^{(l)}$.
8 **End For**
9 **Until:** Performance of M will no longer be improved.
10 Give the diagnosis scores r of $V$and T, by M before its *SoftMax* process.
11 **End For**
12 Choose one scores information to be adjust, and fix the other ρ=1.
13 **Repeat**
14 Adjust $\rho$of the selected scores information.
15 Execute steps given in **20-23** on validation set V.
16 Calculate $std(E_i)$ of its initial probability assignment by formula (14).
17 **Until:** Find the continuous range of $\rho$where diagnosis result for $V$are best.
18 Select a value for ρ with proper standard deviation in that range.
19 **For** M **in** {A: CNN, B: BPNN}:
20 With the selected ρ, compute basic probability assignments by formula (5) to (10):

$$E_i^{M0}(l) = softmax(\rho \cdot r_i(l))$$
$$D_i^M \leftarrow E_i^{A0}(l), E_i^{B0}(l)$$
$$E_i^M(l), E_i^M(\Theta) \leftarrow E_i^{M0}(l), D_i^M$$

21 **End For**
22 Fusion the multimodal evidence bodies by formula ((11) and ((12):
$E_i(l) \leftarrow E_i^A(k), E_i^B(h)$
23 According to the comprehensive information after fusion, give decision by rule:
$Y_i = argMax(E_i(l))$
24 **Return:**Y_i

Algorithm.

Therefore, the decision rule can be achieved by formula (13). For each sequence, the hypothesis with the highest probability is selected as the final result.

$$Y_i = \text{argMax}(E_i(l)), \tag{13}$$

where Y_i is the diagnosis result of the $i-th$ sequence in datasets.

In order to give each sample a definite category, the final probability assignment can no longer exist the universal set item Θ, which represents uncertain category. Besides, according to D-S theory, the difference of belief among hypotheses should be considered when formulating decision rules. Whereas, for the same purpose, no threshold limitation is applied in our decision-making criteria.

The details of the proposed method are shown in the Algorithm.

3.3. Factor Selection. The complementary multimodal information for integration should be at the same crucial level. Indeed, this cannot be guaranteed. If one neural network gives scores of samples all like "0.99-0.005-0.005-0," while the other one is "0.85-0.10-0.03-0.02" (take the assignment with four hypotheses as an example), it will inevitably prevent the second information source from playing its role. So, the adjustment is necessary, and the coordination factor ρ is thus brought in.

The adjustment needs to be done on just one of the two output information from CNN and BPNN, while the other is fixed as $\rho = 1$. The value of factor ρ can be determined according to the accuracy variation of the validation set V. And the relative magnitudes of standard deviations of two initial probability assignments $std(E_i)$ can serve as a reference index, as well.

Table 1: Details of the UCR standard datasets.

Dataset	Sequence amount	Attributes number	Classes number
Chlor.Conc	4307	166	3
Cinc_ECG	1420	1639	4
Dist.phal.O.C	876	80	2
ECGFivedays	884	136	2
Yoga	3300	426	2

Table 2: Comparison of experimental results on UCR standard datasets.

Dataset	Proposed	NN-based methods					Distance-based methods		
		ResNet	MCNN	CTN-T	ESN	TN	ST	LWDTW	BOSS
Chlor.Conc	*0.999*	0.84	0.797	0.83	0.920	0.731	0.700	0.644	0.66
Cinc_ECG	*1.00*	—	0.942	—	0.679	—	0.846	0.935	0.901
Dist.phal.O.C	*0.864*	0.80	—	0.80	—	0.812	—	0.739	0.815
ECGFivedays	*1.00*	0.97	1.00	1.00	1.00	0.926	0.999	0.835	0.983
yoga	*0.941*	0.87	0.888	0.92	0.820	0.84	0.846	0.847	0.901
average	*0.961*	0.88	0.907	0.91	0.855	0.827	0.838	0.800	0.852

For each value of the undetermined ρ, a diagnosis accuracy of V can be obtained by processing its scores information using formula (5)–(13). And $std(E_i)$ should be calculated at the same time with the following formula.

$$std(E_i) = \left(\frac{1}{m}\sum_{l}\left(E_i(l) - \frac{1}{m}\right)^2\right)^{1/2}, \quad (14)$$

When the coordination factor ρ within a certain continuous range, the diagnostic accuracy of V can reach the highest. The proper value of the undetermined ρ should generate in this range. Meanwhile, $std(E_i)$ of the two evidence bodies under the selected value should be similar. More directly, their ratio should approximately between 0.9 and 1.1.

4. Experiments

Since there is no available public standard dataset for aeroengine fault diagnosis, and most artificial intelligence methods of this problem have not opened their source codes, it is difficult to compare the performance of the proposed model with other methods. Fortunately, aeroengine fault data can be treated as time series data, and most methods used to classify time series data can be employed for aeroengine fault diagnosis. The effectiveness of the proposed model can be validated by comparison experiments on standard datasets of UCR (University of California, Riverside) time series.

Therefore, the experimental verification can be done in two parts. Firstly, the proposed method is used to classify standard UCR time series datasets and compared with other time series classification models. Secondly, its effectiveness is verified in the target field by using the real aeroengine fault datasets.

4.1. Experiments on Standard UCR Time Series Datasets. Experiments are firstly executed on several UCR standard datasets, including Chlor.Conc, Cinc_ECG, Dist.phal.O.C, ECGFivedays, and yoga. Details of them are shown in Table 1.

The comparisons among diagnosis accuracies of various models are shown in Table 2, and the specific experiment results of the proposed could be found in Table 3. CTN-T is a deep learning method based on single modal CNN and applying the transfer learning idea [34]. MCNN take into account the features at different time scales. It can be regarded as a model which ensemble multiple CNN at feature level [35]. ResNet is a typical neural network, and paper [36] conducted an experiment to test its ability to classify time series data. LW-DTW is k-nearest-neighbors approach based on a locally weighted dynamic time warping [37]. BOSS proposes a distance which is based on histograms of symbolic Fourier approximation words [38]. The results are from University of East Anglia website. Libsvm library (2017). https://www.csie.ntu.edu.tw/~cjlin/libsvm/. ST (Shapelet Transform) classifies samples by considering the similarity between a shapelet and a sequence, where shapelet is a time-series subsequence. Herein, LW-DTW and CTN-T are the latest methods for time series classification proposed by researchers in 2019 and have good performance. Since the data includes temporal information, comparisons to RNN-based method are also taken into consideration, though classification methods on RNN in recent years are quite few. TN (TimeNet) [39] is a generic off-the-shelf feature extractor for time series data based on RNN automatic encoder, which can be combined with SVM and other classifiers. ESN (Echo state network) is a form of RNN. The authors tried classical and improved ESN-based approach and their fusion version [40]. We took the best results of them for comparison. Besides, RNN-based models are rarely used alone, so in subsequent analyses, they do not perform as representations of deep learning models in this field.

Table 3: Comparison of the performance without and with ρ.

Dataset	BPNN	CNN	Multimodal results	
			Without	With
Chlor.Conc	0.991	0.993	0.999	0.999
Cinc_ECG	0.958	0.996	1.00	1.00
Dist.phal.O.C	0.835	0.858	0.858	0.864
ECGFivedays	0.983	1.00	1.00	1.00
yoga	0.931	0.926	0.939	0.941
Type A	0.9598	0.9489	0.9621	0.9627
Type B	0.9818	0.9771	0.9843	0.9864

Horizontal contrast shows that the performance of deep learning models is better than distance-based methods on the whole. The overall accuracy of ResNet, MCNN, and CTN-T on the above datasets is about 90%, higher than that of Shapelet, LW-DTW, and BOSS methods. Distance-based methods distinguished samples by its closeness, which does

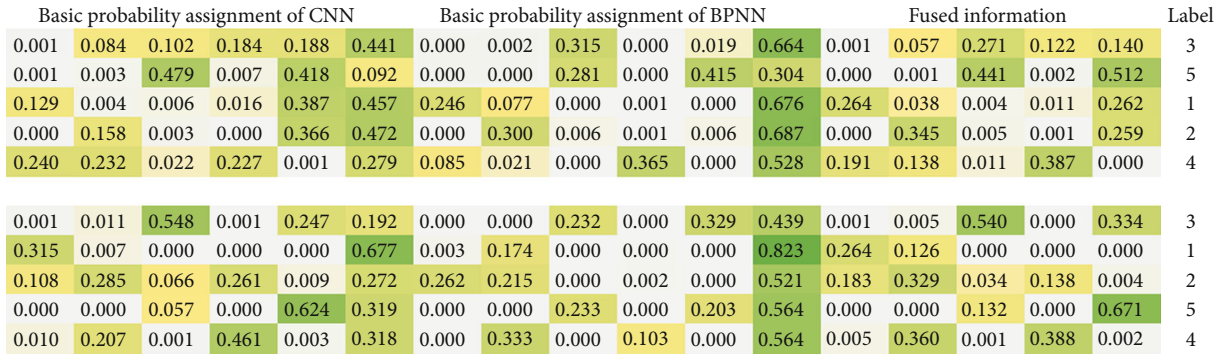


Figure 2: Basic probability assignment heat map of examples corrected by multimodal information.

not reflect the distribution of sample's attributes. While deep learning methods fitted the samples intensively; thus, they can better cope with the highly complex time series data. On the above dataset (some details could be found in Table 3), the basic deep neural network model like CNN and BPNN can achieve higher diagnosis accuracy than other complex versions. And after fusion, better performance can be obtained. Even in the case that the accuracy of single modal model like CNN reaches extreme high of 99%, for instance, on datasets Chlor.Conc and Cinc_ECG, it can still be improved in fused model. It was the comprehensive diagnosis information that contribute to the superior result. Which lead the multimodal model to have capabilities of BPNN and CNN at the same time. Figure 2 in the next experiments will give a more intuitive and detailed explanation for this. In addition, another support for this interpretation is that MCNN, the integration of homogeneous model, is even not as accurate as the single modal fully connected neural network on many datasets, in the comparison of the original paper [35].

4.2. Experiment on Real Aeroengine Fault Dataset. The aeroengine fault datasets used in this experiment are from AVIC Shenyang Engine Design Institute (606 institute). These datasets contain fault data of engine type A and type B, and the data has 23 dimensional attributes and 5 fault modes. Through experimental comparison, each model can be best trained when total 60,000 samples of various fault modes are taken. Samples were divided into training set, validation set, and test set according to the proportion of 0.7, 0.1, and 0.2. In addition, the collection surrounding of above datasets is a laboratory with relatively stable temperature and pressure. In order to simulate the influence brought by terrible environment at high altitude that aeroengines work, a quarter of sample data are randomly selected and added with Gaussian white noise.

Heat map of basic probability assignment in Figure 2 shows how sequences are corrected by multimodal information. Where the sixth term in basic probability assignment of BPNN and CNN represents its uncertainty. The first five sequences are corrected by the contribution of BPNN, and the last five are by that of CNN. Due to CNN and BPNN focus on different aspects of samples, they might give inconsistent diagnosis result for the same sequence. For the case where one model gives right result while the other judge wrongly, the fusion of multimodal information may lead to the correct final result. Therefore, on the whole, the multimodal model can correctly diagnose more samples.

Table 4: Experiment results on aeroengine fault datasets.

Dataset	CNN	BPNN	Multimodal NN
Aeroengine type A	0.9598	0.9489	0.9627
Aeroengine type B	0.9818	0.9771	0.9864

Table 4 shows the diagnosis accuracy of two kinds of single modal neural networks and that of multimodal one. On the two types aeroengine fault datasets, the accuracies of the multimodal neural network are improved by 0.29% and 0.46%, compared with the maximum values between the basic CNN and BPNN. Observation shows that the fusion of multimodal decision information by D-S evidence theory can effectively improve the diagnosis accuracy, and this model has a good performance on the problem of aeroengine fault diagnosis.

Meanwhile, we also paid attention to the performance improvement brought by the coordination factor ρ during the experiment. Except for the three datasets with extremely high diagnosis accuracy given by the basic model, it plays a good role on both other UCR datasets and aeroengine fault datasets. Particularly, for the dataset Dist.Phal.O.C, the coordination factor ρ makes the contribution of evidence theory improved from none to increasing accuracy by 0.6%. It enables the evidence theory to better integrate the multimodal information.

5. Conclusion

This paper proposes a multimodal aeroengine fault diagnosis model based on CNN and BPNN, aiming at obtaining high diagnosis accuracy and relatively strong robustness by comprehensively considering the complementary decision information from pure neural network. Firstly, sequence data will be preprocessed. Then, the two neural networks are trained with it. And then, the multimodal information of two networks are fused by D-S evidence theory. At last, final diagnosis results are given. This paper also gives a method to construct the basic probability assignment of evidence body by the output scores information of neural network, as well

as the fusion formulas and decision rules of D-S theory for practical application.

The two modal neural networks, BPNN and CNN, have the same level but distinctive ability. BPNN can establish mapping between a certain type of fault and the overall distribution of all attributes in sample, and CNN using one-dimensional convolution kernels can extract features from samples and explore relationship between features of attributes and the existence of faults. Moreover, pure neural network-based method can better adapt to input data of various specifications.

Experiment on UCR standard time series datasets proves that classification methods based on neural network are better than methods based on distance, and the proposed model has better performance than some typical models and latest methods on certain datasets. And though ultrahigh accuracy the single modal neural network has, it can still be enhanced by fusing multimodal diagnosis results. Likewise, the experiment conducted on the aeroengine fault data has also achieved good results, which illustrates that neural network does have a strong ability to cope with aeroengine fault diagnosis, and the fusion of multimodal information at decision level is indeed an effective way to further improve accuracy.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61906030) and the Science and the Fundamental Research Funds for the Central Universities (DUT20RC(4)009).

References

[1] Q. Pan, Y. Liu, R. Zhou, H. Wang, H. Chen, and T. He, "An automatic abrupt signal extraction method for fault diagnosis of aero-engines," *Journal of Mechanical Science and Technology*, vol. 33, no. 4, pp. 1633–1640, 2019.

[2] Q. Huang, G. Zhang, T. Zhang, and J. Wang, "A kind of approach for aero engine gas path fault diagnosis," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Location: Dallas, TX, USA, 2017.

[3] Z. B. Shi, Q. G. Song, M. Z. Ma, and Q. Li, "Fault diagnosis of steam turbine based on mpso-svm algorithm," *Journal of Chinese Society of Power Engineering*, vol. 32, no. 6, pp. 454–462, 2012.

[4] X. Shou, "Aero-engine fault diagnosis based on support vector machine," *Mechanical Science and Technology*, 2005.

[5] D. Seo, T. Roh, and D. Choi, "Defect diagnostics of gas turbine engine using hybrid SVM-ANN with module system in off-design condition," *Journal of Mechanical Science and Technology*, vol. 23, no. 3, pp. 677–685, 2009.

[6] F. Xia, H. Zhang, D. Peng, H. Li, and Y. Su, "Turbine Fault Diagnosis Based on Fuzzy Theory and SVM," in *Artificial Intelligence and Computational Intelligence*, vol. 5855, Springer, Berlin, Heidelberg, 2009.

[7] C. Ren, H. Dong, P. Hou, X. Dong, and Y. Tao, "A clustering-based method for health conditions evaluation of aero-engines," in *2019 Prognostics and System Health Management Conference (PHM-Paris)*, Paris, France, France, 2019.

[8] X. S. Lin, B. W. Li, and X. Y. Yang, "Engine components fault diagnosis using an improved method of deep belief networks," in *2016 7th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, London, UK, 2016.

[9] R. Jiang and W. Zhu, "A pnn fault diagnosis method for gas turbine," in *World Automation Congress 2012*, pp. 1–4, Puerto Vallarta, Mexico, Mexico, 2012.

[10] Z. Ning, R. Y. K. Kwok, K. Zhang et al., "Joint computing and caching in 5g-envisioned internet of vehicles: A deep reinforcement Learning-Based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[11] L. Zhao, Z. Chen, L. T. Yang, M. J. Deen, and Z. J. Wang, "Deep semantic mapping for heterogeneous multimedia transfer learning using co-occurrence data," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 1s, pp. 1–21, 2019.

[12] Z. Ning, Y. Li, P. Dong et al., "When deep reinforcement learning meets 5g-enabled vehicular networks: a distributed offloading framework for traffic big data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1352–1361, 2020.

[13] Z. Jiang, H. Fang, H. Shi, S. Ren, H. Yang, and F. Wang, "The prognostic method of engine gas path based-on convolutional neural network," *DEStech Transactions on Computer Science and Engineering*, no. iciti, 2019.

[14] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," in *2016 International Symposium on Flexible Automation (ISFA)*, pp. 65–70, Cleveland, OH, USA, 2016.

[15] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on resnet-50," *Neural Computing and Applications*, vol. 32, no. 10, pp. 6111–6124, 2020.

[16] S. Zhong, S. Fu, and L. Lin, "A novel gas turbine fault diagnosis method based on transfer learning with CNN," *Measurement*, vol. 137, pp. 435–453, 2019.

[17] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, 2020.

[18] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[19] C. Angelakis, E. N. Loukis, A. Pouliezos, and G. S. Stavrakakis, "A neural network-based method for gas turbine blading fault diagnosis," *International Journal of Modelling and Simulation*, vol. 21, no. 1, pp. 51–60, 2001.

[20] S. O. T. Ogaji and R. Singh, "Advanced engine diagnostics using artificial neural networks," *Applied Soft Computing*, vol. 3, no. 3, pp. 259–271, 2003.

[21] N. Zhao, S. Y. Li, S. Yi, Y. P. Cao, and Z. T. Wang, "Fault diagnosis based on rough set and bp neural network (rs-bp) for gas turbine engine," *Advanced Materials Research*, vol. 732-733, pp. 397–401, 2013.

[22] B. Yuan, F. Xia, Z. Wang, and H. Tie, "A comparative research based on three different algorithms for fault diagnosis in gas turbine," in *2017 4th International Conference on Systems and Informatics (ICSAI)*, Hangzhou, China, 2017.

[23] A. D. Fentaye, A. T. Baheta, and S. I. Gilani, "Gas turbine gas-path fault identification using nested artificial neural networks," *Aircraft Engineering and Aerospace Technology*, vol. 90, no. 6, pp. 992–999, 2018.

[24] S. Xiangyang, "Research on aero-engine fault diagnosis based on integrated neural network," *Mathematical Models in Engineering*, vol. 5, no. 2, pp. 41–47, 2019.

[25] Y. Xiaohong, G. Haifeng, Z. Jing, X. Jing, and Z. Dandan, "Aero engine gas path fault prediction based on multi-sensor information fusion," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, 2016.

[26] C. Xu, H. Zhang, D. Peng, Y. Yu, C. Xu, and H. Zhang, "Study of fault diagnosis of integrate of D-S evidence theory based on neural network for turbine," *Energy Procedia*, vol. 16, no. 2012, pp. 2027–2032, 2012.

[27] L. Xiaolin and Q. Weidong, "Aero-engine fault fusion diagnosis based on D-S evidence theory," in *Proceedings of the 37th Chinese Control Conference*, pp. 963–967, Wuhan,China, 2018.

[28] H. Song, *Aeroengine fault diagnosis based on infomation fusion technology,[Ph. D. Thesis]*, Central South University, 2013.

[29] W. J. Wu, D. G. Huang, and Z. Dong, "Research on a fault diagnosis method for aero-engine based on improved svm and information fusion," *Applied Mechanics and Materials*, vol. 66-68, pp. 811–816, 2011.

[30] G. Lan, N. Cheng, and Q. Li, "Comparison and fusion of various classification methods applied to aero-engine fault diagnosis," in *2017 29th Chinese Control And Decision Conference (CCDC)*, Chongqing, China, 2017.

[31] P. Jun and H. Jiangbo, "Aero-engine fault diagnosis based on ipso-elman neural network," *Journal of Aerospace Power*, vol. 32, no. 12, pp. 3031–3038, 2017.

[32] L. Zhao, T. Yang, J. Zhang, Z. Chen, Y. Yang, and Z. J. Wang, "Co-learning non-negative correlated and uncorrelated features for multi-view data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2020.

[33] Z. Ning, P. Dong, X. Wang et al., "Mobile Edge Computing Enabled 5g Health Monitoring for Internet of Medical Things: A Decentralized Game Theoretic Approach," *IEEE Journal on Selected Areas in Communications, to Appear*, 2020.

[34] K. Kashiparekh, J. Narwariya, P. Malhotra, L. Vig, and G. Shroff, "Convtimenet: A pre-trained deep convolutional neural network for time series classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, Hungary, 2019.

[35] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification Computer Vision and Pattern Recognition," 2016, https://arxiv.org/abs/1603.06995.

[36] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, 2017.

[37] J. Yuan, A. Douzal-Chouakria, S. Varasteh Yazdi, and Z. Wang, "A large margin time series nearest neighbour classification under locally weighted time warps," *Knowledge and Information Systems*, vol. 59, no. 1, pp. 117–135, 2019.

[38] P. Schafer, "The boss is concerned with time series classification in the presence of noise," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1505–1530, 2015.

[39] P. Malhotra, V. Tv, L. Vig, P. Agarwal, and G. Shroff, "Timenet: Pre-Trained Deep Recurrent Neural Network for Time Series Classification: Learning," 2017, https://arxiv.org/abs/1706.08838.

[40] W. Aswolinskiy, R. F. Reinhart, and J. Steil, "Time series classification in reservoir- and model-space," *Neural Processing Letters*, vol. 48, no. 2, pp. 789–809, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8879054, 10 pages
https://doi.org/10.1155/2020/8879054

Research Article

An Embedded-Based Weighted Feature Selection Algorithm for Classifying Web Document

G. Siva Shankar,[1] **P. Ashokkumar**,[1] **R. Vinayakumar**,[2] **Uttam Ghosh**,[3] **Wathiq Mansoor**,[4] **and Waleed S. Alnumay**[5]

[1]*Department of Computer Science, Sri Ramachandra Institute of Higher Education and Research, Chennai 600116, India*
[2]*Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA*
[3]*Vanderbilt University, USA*
[4]*Computer Engineering, University of Dubai, Dubai, UAE*
[5]*King Saud University, Riyadh, Saudi Arabia*

Correspondence should be addressed to P. Ashokkumar; ashok05002@gmail.com and Wathiq Mansoor; wmansoor@ud.ac.ae

Received 8 June 2020; Revised 30 July 2020; Accepted 17 August 2020; Published 15 September 2020

Academic Editor: Xiaojie Wang

With the exponential increase in a number of web pages daily, it makes it very difficult for a search engine to list relevant web pages. In this paper, we propose a machine learning-based classification model that can learn the best features in each web page and helps in search engine listing. The existing methods for listing have lots of drawbacks like interfacing the normal operations of the website and crawling lots of useless information. Our proposed algorithm provides an optimal classification for websites which has a large number of web pages such as Wikipedia by just considering core information like link text, side information, and header text. We implemented our algorithm with standard benchmark datasets, and the results show that our algorithm outperforms the existing algorithms.

1. Introduction

With the rapid growth of the number of web pages every day, it makes a web crawler difficult to read and organize the web pages. This problem makes the web classification process to get more important day by day. Web classification algorithms have many uses across various domains which include spam detection, web searching, document organizing, and cybersecurity [1]. In this paper, we are targeting web searching and aim at optimizing the search engine for providing quick and efficient search results to the users.

The web classification process can be done as follows: firstly, fix the number of classes and the properties of each class; secondly, given a set of training documents, the goal is to find the probability of all the classes that each document can fall in; lastly, the classification chooses the class which has the highest probability. There are lots of machine learning algorithms for the web classification process such as Support Vector Machine (SVM), Naive Bayes, and k-Nearest Neighbours (kNN) [2]. However, most of the machine learning models fail to produce desirable accuracy because there exist lots of features in a single web page. Hence, a good feature selection algorithm is combined with the machine learning model to increase the accuracy of the classification process. The goal of a feature selection algorithm is to get rid of most of the irrelevant features in the web page so that the input size that is fed to the machine learning model is reduced. As the input size is minimum, the machine learning model gets easier to learn the corelation between various features and performs better in terms of accuracy.

The main problem for the web crawler is to decide on what features (or words) to pick to simplify the web classification process and to achieve good accuracy. Most of the feature selection model falls under three categories, namely, Wrapper, Filter, and Embedded. Wrapper-based methods iteratively calculate the subset of features until an optimal subset is found that has the maximum performance. It starts with a zero-sized subset and iteratively adds/removes the features

and finds the accuracy. It recommends the best feature subset that yields the maximum accuracy. Sometimes, the wrapper methods work backward also. Filter-based methods try to rank each feature based on a few metrics such as Pearson correlation and Analysis of Variance. Then, it recommends top N features. Embedded methods are the hybrid of the other two methods. The feature selection that is going to be used in this paper is going to be the Embedded-based one.

Almost all of the existing works rely on term frequencies. If a term is present across all documents of the same class and is absent across all documents of other classes, then that particular term has a high weight in representing the class of the document because it is unique. The aim of this study is to not fully rely on term frequencies alone, instead considering other parts of the document which has the ability to represent the document class more accurately than the term frequencies.

Embedded-based feature selection has lots of difficulties for picking the required feature among the web pages for two main reasons; firstly, each web page has a different template; secondly, the number of web documents increases exponentially day by day. To solve the two problems, the proposed method extracts few information on the web page such as headings, link texts, and side information, which includes meta tags. A higher rank is been given to this extracted information, and the rest of the texts are given a lower rank based on Pearson correlation. Later, these features are fed into the machine learning model and got satisfying results while compared to existing methods.

1.1. Contribution. The main contributions of this paper are as follows:

(i) To increase the speed and accuracy of the web classification algorithm by reducing the dimensions of the document matrix

(ii) Side information such as meta tags, heading tags, table captions, and image descriptions is considered for classification purposes

(iii) The relationship between two web documents is calculated by the link between them. Using these relationships, a new way of classification is proposed

The rest of the paper is as follows, Section 2 describes the literature survey done by researchers on the field of link classification. Section 3 explains the working of our proposed methodology, and Section 4 compares our work with standard existing algorithms. Section 5 contains the discussions. The conclusion and future work are then proceeded by Section 6.

2. Related Works

In this subsection, we present a brief literature review on the recent research based on link-based web classification algorithms. The main goal of these algorithms is to use a clustering algorithm that relies on distance calculation steps to minimize the average distance among the web pages belonging to the same class and maximize the average distance among the web pages belonging to different classes. [3] explains the use of the tokenization of a web page to extract features; the class for a web page is then assigned based on the calculating distance between the features. Some researchers use machine learning algorithms such as support vector machines for classifying the web pages which extracts a large number of web links [4].

Few works like [5] calculate the fitness value on each iteration for classifying the documents. The fitness function gets more accurate at each successive iteration, and the iteration stops when there is no future increase in the fitness function. There are many works in the area of swarm optimization which is a nature-inspired algorithm where each member from a group of birds searches for food in different places and finally converges when food is found. The same algorithm is used to classify the web documents by searching for links across the documents in the group of web pages, and finally, all the partial solutions are merged to form a super optimal solution [6, 7].

The majority of the works in this area are based on the bag of words model [8]. The links in the document are converted into vectors, and then few algorithms like Term Frequency-Inverse Document Frequency (TF-IDF) are applied to extract the more frequent words present across the web pages [9]. These frequent words are used as features to classify the web pages [10, 11]. TF-IDF can only be efficient as long as each word in the document matrix is independent of each other, in case two or more words are synonym of each other, then it becomes difficult to represent the exact relationship; the research work [12] proposes a method which calculates the correlation between each pair of words to detect the synonym.

[13] proposes a kNN-based classification algorithm that uses the TF-IDF metric for classifying the Lao text; the uses of their research has wide use in Natural Language Processing (NLP). The input size used in their research has 7 attributes. The feature selection method they used is principal component analysis. The principal component analysis is an algorithm that can map an N feature vector space to a K feature vector space, where $k¡N$. Only 3 features are considered after the PCA feature selection method, and the accuracy of their proposed work is 71.4%.

[14] optimizes the support vector machine to be used in multiclass classification. SVM is good when there is only two target class, that is binary classification, but when there are more than 3 classes, then the computational cost of training is increased. In their research, they proposed a hierarchical classification model in which the SVM is optimized for a multiclass classification process. They compared their work with decision tree classifiers and kNN classifiers and produced better performance. Various optimization [15, 16] was done in the SVM classifier to produce better results for multiclass classification.

In recent years, the amount of data generated is huge and very complex [17], for example, lots of time series data are generated [18]. To tackle this issue, deep learning technology [18] is used for feature extraction. The deep learning approach performs better than the other existing machine learning algorithms [19].

There is one more problem that text feature selection can face, that is redundancy. Two or more different features having the same meaning are called redundant (sometimes called a synonym problem). [12] focuses on solving this

Table 1: The usage of machine learning algorithms for document classification—a summary.

Reference	Method	Summary
[30]	DragPushing	(i) Proposes kNN optimization which automatically balances the data points evenly across all the classes to avoid model misfits. (ii) They have set the value of k to 7 for their experiment. (iii) Three datasets are used Reuter-21578, industry sector, and TDT-5.
[31]	Prototype selection	(i) Much faster than [30]. (ii) Prototype selection recommends the most portable prototypes for training purposes. (iii) [31] eliminates most of the data points in training to increase speed.
[32]	ForesTexter	(i) The Gini index can easily predict the skewness in the majority class and creates many subtrees to balance the data points. (ii) Can work faster than [30, 31]. (iii) [32] combines both feature subspace selection and splitting criterion to create multiple subtrees to balance the data.
[33]	Resampling	(i) Handles the imbalance problem better than [32] by performing resampling. (ii) Instance weighting enables one to assign few weights for the imbalanced class so that the end performance (in terms of accuracy) is balanced. (iii) They validated the proposed method with SVM classifier.
[34].	Topic modelling	(i) Instead of balancing the data points across all the classes, this method uses the topic model to construct new data points in each class to create a complete dataset. (ii) This method considers more data points than [30, 32, 33] because of topic modeling which can construct new data points.
[35]	Bag of concepts	(i) Aims to reduce the dimensions of the document matrix representation. (ii) Instead of recommending data points from the data set (such as [31]), the bag of concepts groups one or more data points into topics. (iii) Bag of concepts solves many problems in the traditional bag of words models such as high dimensionality and sparsity issues.
[36]	Ontology-based deep learning model	(i) Enhances the problem of [35] by not considering the relationships among the documents. (ii) The features and their relationships are extracted based on deep learning. (iii) The ontology enhancement proposed in [36] helps to reduce the high dimensions. (iv) This method consumes more time in training the samples.
Proposed	Weighted feature selection	(i) Resolves the imbalance problem by assigning weights to the most important features. (ii) Three classifiers are used, namely, kNN, SVM, and Naïve Bayes. (iii) [35] fails to detect the relationships among the documents. In this paper, the proposed system detects the relationship and considers it for classification.

redundancy by considering semantic relationships along with corelation. Table 1 compares few existing works with the proposed feature selection model. Recently, many classification tasks in documents, such as [20], are done by deep learning. [21] recommends the use of three hidden layers with 1024 neurons yields in good accuracy; they also found that if the proper preprocessing (such as synonym elimination) is been applied, then the F1-Score is increased by 5 points. Different research papers use different methods for doing preprocessing; in the research work [22], each word is mapped to its concept vector which is a set of semantic words. In this paper, the proposed method uses the concept of link words and High-End Features to achieve dimension reduction. Furthermore, few researches [23, 24] focus more on finding out the relationship among various parts of the document towards the document class and recommended which parts to consider to increase the accuracy of the classification process.

3. Embedded-Based Feature Selection Method

The proposed method makes use of both Wrapper (selection of headers, link text, and side information) and Filter (correlation). The proposed method first assigns a few feature weights to the selected features and then assigns a few class weights to each document. Then, the original feature selection is implemented for selecting top N features for the web classification. The feature and class weights are calculated based on two metrics

(1) High end features (HEF)

(2) Weightage based on Links

After weights are assigned, the filter-based correlation feature selection is implemented to rank the features. Figure 1 shows the overall model of the proposed algorithm; the input documents are fed into the feature selection algorithm. There are two phases, the first phase assigns few weights to selective features, and the second phase computes the correlation for all the features. After the two phases are completed, the ranking step recommends top N features to the classifier.

3.1. High End Features (HEF). Since each web page may have different templates, it is difficult to choose which context has a high correlation between the target class. A HEF is a text that

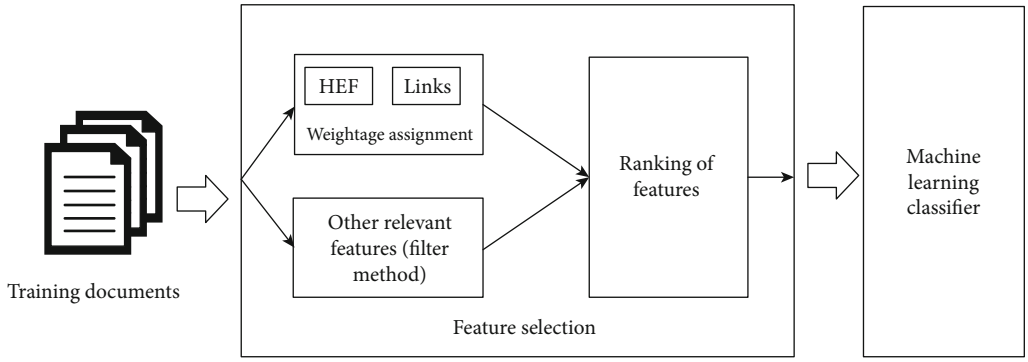


Figure 1: The model of the proposed algorithm.

determines a high correlation between the document and the target class. In this step, the proposed algorithm considers the texts present in headers, meta tags, and link text as HEF because they provide more information towards the determination of the target class while compared to other texts present in other areas. These HEF are given additional weightage for calculation of ranks; equation (1) describes the weightage process. The weightage W vector = $\{W1, W2, \cdots, Wn\}$, W is an n-dimensional vector (n denotes the number of unique words in the webpage) consisting of weights for each word.

$$\text{Weightage } W = \begin{cases} 1, \text{if word is not in HEF,} \\ 1.5, \text{if word is in HEF.} \end{cases} \quad (1)$$

3.2. Link Texts. The second metric that is considered is the link between the documents. Web documents are easy to classify while compared to normal text documents because of the links. A link between the two documents indicates a high correlation between the two documents. In this step, the proposed method selects only a few documents and run the machine learning model to predict the class. Then, the algorithm assigns the identified class of the selected documents to the rest of the documents based on the degree in which they are connected by links. The selected documents are called as leaders. The leaders are calculated by Link Inbound Outbound based classification algorithm (LIOBC). The algorithm is given at Algorithm 1.

3.3. LIOBC: Link Inbound Outbound Based Classification Algorithm. The algorithm first constructs a graph-based model from the collection of web pages, where each web page is represented as a node in the graph, and a link from two nodes is formed only when there is a link between them. A bag of the model is represented using the links available in a graph and then the Algorithm 1 (graph creation algorithm) connects the nodes (web pages) based on the hyperlinks. Thus, a web page having $n1$ hyperlinks will have $n1$ outbound and a web page that has $n2$ referring from other web documents will have $n2$ inbound.

There are lots of preprocessing steps that need to be done before executing the LIOBC algorithm such as detecting fake URLs [25], removing useless information like advertisements, images [26], and reducing the dimensionality of the contents [27, 28].

```
1:  procedure GraphGeneration
2:      N ⟵ List of web pages
3:  begin:
4:      for each document i in N do
5:          for each link l in i do
6:              J = Target of l
7:              create link between J and i
```

Algorithm 1: Graph generation.

Once the graph is generated, then it becomes easy to group related documents into separate distinct classes. A single document can belong to more than one class based on its similarity. The output of Algorithm 1 will create all links between web pages as shown in Figure 2(a). Two webpages can have more than one in links or out links or both. After creating links between web pages, the next step is to identify the leaders. A leader is a node that has several inbound than a specific fixed threshold TH1. Each leader is then given as an input to the machine learning model that assigns classes to leaders.

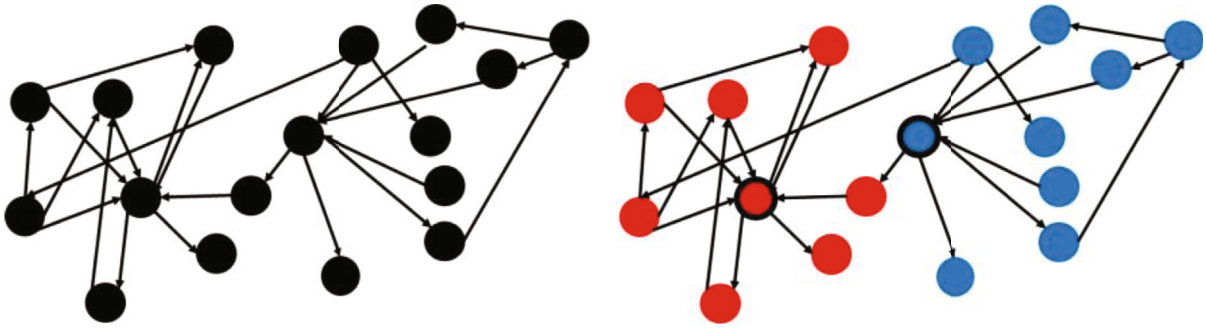
Leader (definition): a webpage that has the number of links higher than TH1. This signifies the high coupling. This coupling information can be used to increase the accuracy of the classifier.

After the classes of the leaders are found, it is very easy for assigning classes for the nonleader nodes. The assignment is as follows:

(i) A nonleader node which is having several outbound to only one leader greater than a prefixed threshold TH2, then the class of leader is fully assigned to the class of nonleader

(ii) A nonleader node which is having some outbound to more than one leader greater than a prefixed threshold TH2, then the classes of all leaders are equally assigned to the nonleader

(iii) If a nonleader node is not having enough outbound (number of outbound is less than TH2), then the machine learning model is used to run on nonleader independently considering it as a leader



(a) (b)

Figure 2: The class assignment of a document from leaders. (a) shows the output of Algorithm 1, and (b) shows the class assignment from leaders (leaders are bordered with dark black).

```
1:  procedure FeatureSelection
2:    BOW ⟵ Bag of Word representation of the corpus
3:    L ⟵ Set of Leaders from Algorithm 1
4:    G ⟵ The graph model from Algorithm 1
5:    begin:
6:    W={}
7:    for each Word w in BOW do
8:      if w thenord ∈ HEF such as link text, headings, meta informations, image descriptions etc...
9:        W[w]=BOW[w]∗1.5
10:     else
11:       W=[w]=BOW[w]
12:   DocClass={} - Represent the Document-Class vector, all documents are initialized 1 to all classes.
13:   for each leader l in L do
14:     class,prob=get class and probability of l from the classifer
15:     update DocClass, value=l, class and prob, weight =1.5
16:     for each neighbor n of l in G do
17:     update DocClass, value=n, class and prob, weight =1.5
18:   for each document d in the corpus do
19:   class,prob=get class and probability of d from the classifer
20:   update DocClass, value=l, class and prob, weight =1
21:   Assign the class which is having the highest prob∗weight value to d.
```

Algorithm 2: Embedded-based weighted feature selection algorithm.

The values of TH1 and TH2 can be dynamically fixed to get better accuracy. At the end of this step, each document will have a partial association (weightage) with a class by its leaders. Equation (2) represents this partial weightage association.

$$\text{Document Class Vector (DCV)} = \begin{cases} 1, \text{ for nonassociated class,} \\ 1.5, \text{ for associated class} \\ \text{(from leaders).} \end{cases} \tag{2}$$

3.4. The Ranking Method. Once the feature weights and class weights are assigned, each feature on the web page is ranked based on the Pearson Correlation. The final weight for each feature is calculated using equation (3). Then, top N features are recommended to the machine learning model. If the features are having the same final weight, then the feature with a high Pearson Correlation is been recommended. If the Pearson Correlation is also the same, then the features are recommended stochastically.

Table 2: Dataset description.

Dataset name	Number of documents
ClueWeb12	25 K
DBpedia	25 K

$$\text{Final Weight} = \text{Pearson Correlation} * W. \tag{3}$$

3.5. Web Page Classification. The machine learning models such as Naive Bayes, SVM, and kNN are implemented based on the top N final weight features, and each documents final class is calculated by equation (4). The class which is having

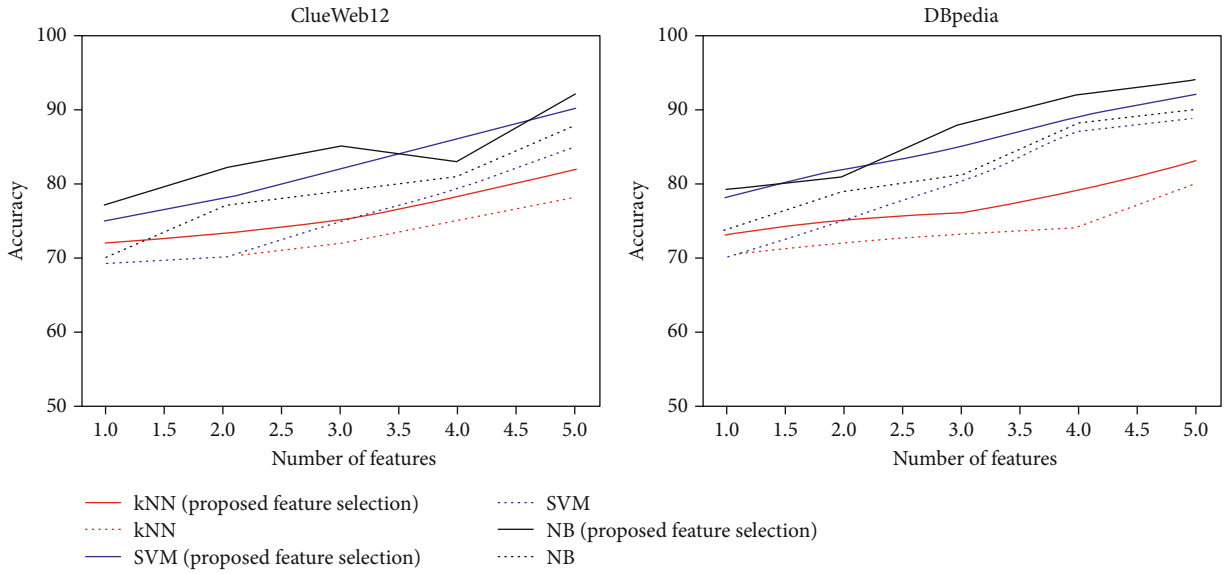


Figure 3: The accuracy of all the three classifiers when the number of classes = 5.

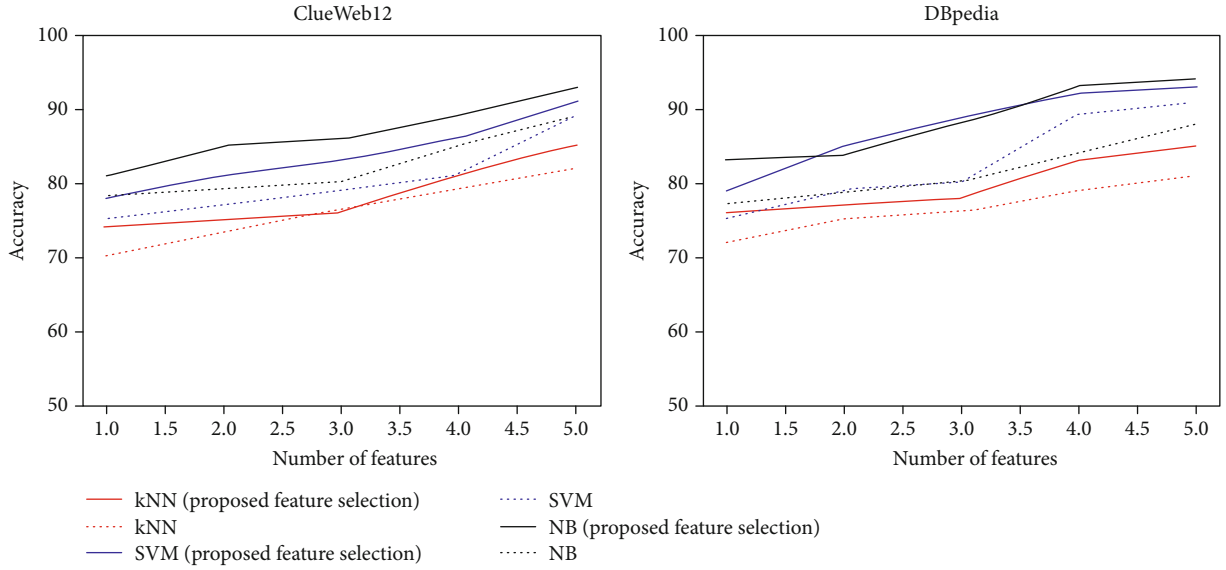


Figure 4: The accuracy of all the three classifiers when the number of classes = 7.

the highest probability is assigned to the web page. Algorithm 2 shows the overall working of the proposed model.

$$\text{Class Weight} = \text{Predicted} * \text{DCV}. \tag{4}$$

4. Experiment Result and Analysis

We have implemented our proposed algorithm on two standard datasets which are shown in Table 2. We took random 25 K documents from the two standard datasets and ran the experiment with three classifiers kNN, SVM, and Naive Bayes (NB). We have used Python 3.7 for implementing our experiment.

4.1. Preprocessing Stage. The datasets are preprocessed as follows: first, all the characters are converted into the lower case; second, the digits, punctuation marks, and stop words are filtered out; third, the stemming operation is performed and finally all the words are transformed into normalized words by adjusting singular, plural.

4.2. Performance Analysis. The efficiency of the classification model can be calculated using the accuracy metric. The calculation of accuracy is as per equation (5).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{5}$$

TP, TN, FP, and FN are first calculated for each feature in the dataset and then finally cumulated. The definitions for TP, TN, FP, and FN are as follows (for each feature)

(i) True Positive (TP) - determines how many documents classified as the positive class which contains the feature

(ii) True Negative (TN) - determines how many documents classified as the negative class which does not contain the feature

(iii) False Positive (FP) - determines how many documents classified as the negative class which contains the feature

(iv) False Negative (FN) - determines how many documents classified as the positive class which does not contain the feature

The better the accuracy the more efficient the classification model is. The comparison in Figure 3 (number of class = 5) and Figure 4 (number of class = 7) shows that our algorithm has better stats because the existing algorithm (without the proposed feature selection algorithm) uses the full documents which often leads to high noise. Weighted words are better comparison parameters than ordinary text because the weighted words tell more about the content of the document than normal text do. We have tested all algorithms by having several classes 5 and 7. The class topic name is listed in Tables 3 and 4, respectively. Figure 5 shows how many link texts are recommended in the proposed method. If the proposed feature selection technique is not used, then the classifier is given the whole text corpus features as an input to perform the classification process.

Classification accuracy can be measured using another parameter called purity, which is defined as a fraction of documents classified to the correct class. The higher value of purity will yield better classification results. Purity is found as per equation (6).

$$\text{Purity} = \frac{\sum_{i=1}^{N} D_i}{\sum_{i=1}^{N} N_i}. \quad (6)$$

Purity measure is calculated for all the three algorithms as per equation (6), where N denotes the total number of classes, D_i denotes the documents classified to the class i, N_i denotes the total number of documents belonging to the class i. Graphs in Figures 6 and 7 show the result of purity measure when the number of classes is 5 and 7, respectively.

5. Discussion

Web documents are having an advantage over normal text documents regarding classification that is it provides lots of additional information regarding the correlation towards the topic. This additional information includes links (which represent a strong relationship of correlation between the two documents), headings (the header tags represent the heading or topic of the content), and other side information which includes meta tags, title tags, and description tags.

While most of the research work carried out on the field of feature selection focus on ranking the features based on the number of times it appears on the document, the proposed method makes use of the abovementioned additional information and gives extra weightage to them. Thus, prioritizing this additional information over normal text helps to reduce the noise and improve the accuracy of the classifier, for example, let us take a web document [29] as shown in Figure 8. The additional information like link text is circled, while the normal text is rectangular bordered. From the image, it is clear that the additional information words like

Table 3: Topic heads (For #classes = 5).

#	Topic head name
1	People and animals
2	Tourism
3	Health
4	Entertainments
5	Organizations

Table 4: Topic heads (For #classes = 7).

#	Topic head name
1	Culture
2	Animals
3	Tourism
4	Health
5	Games
6	Films
7	Organizations

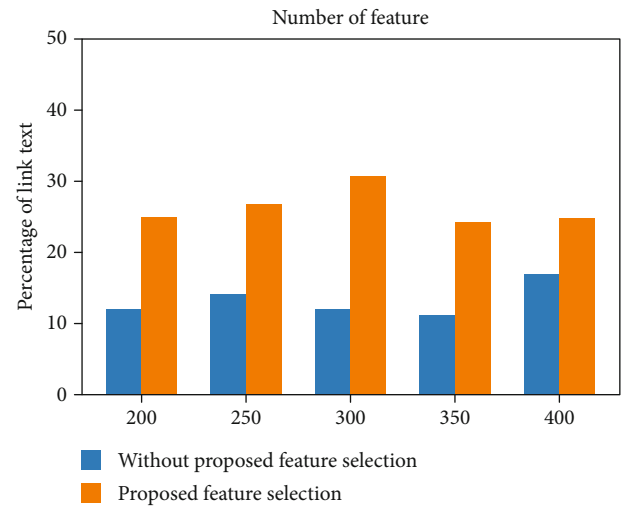


Figure 5: The number of link text recommended in top N features.

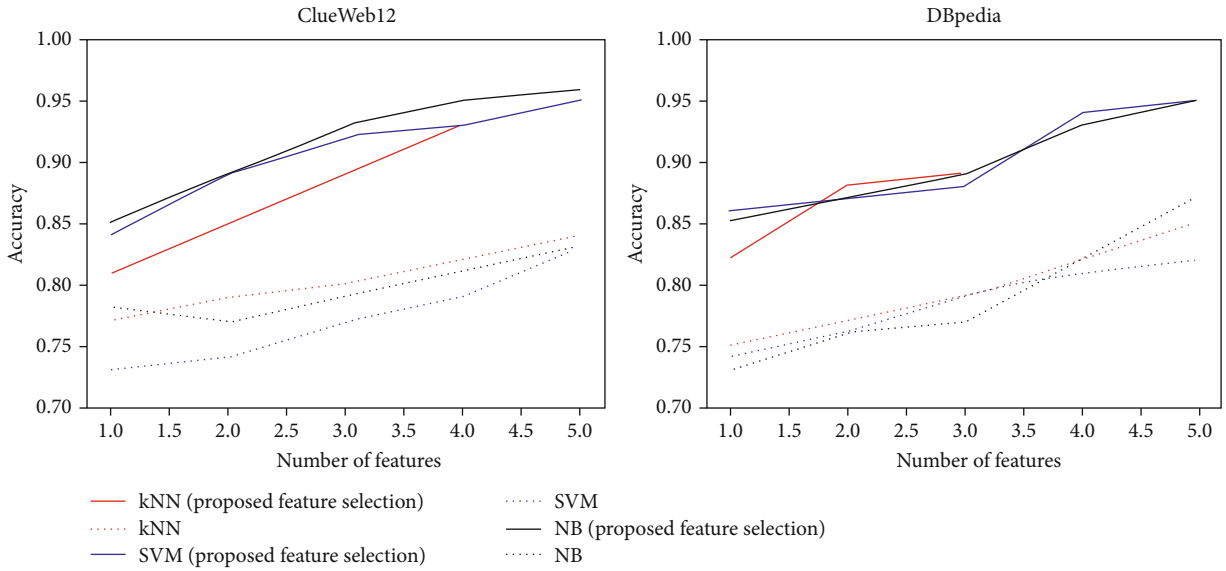


Figure 6: The purity of all the three classifiers when the number of classes = 5.

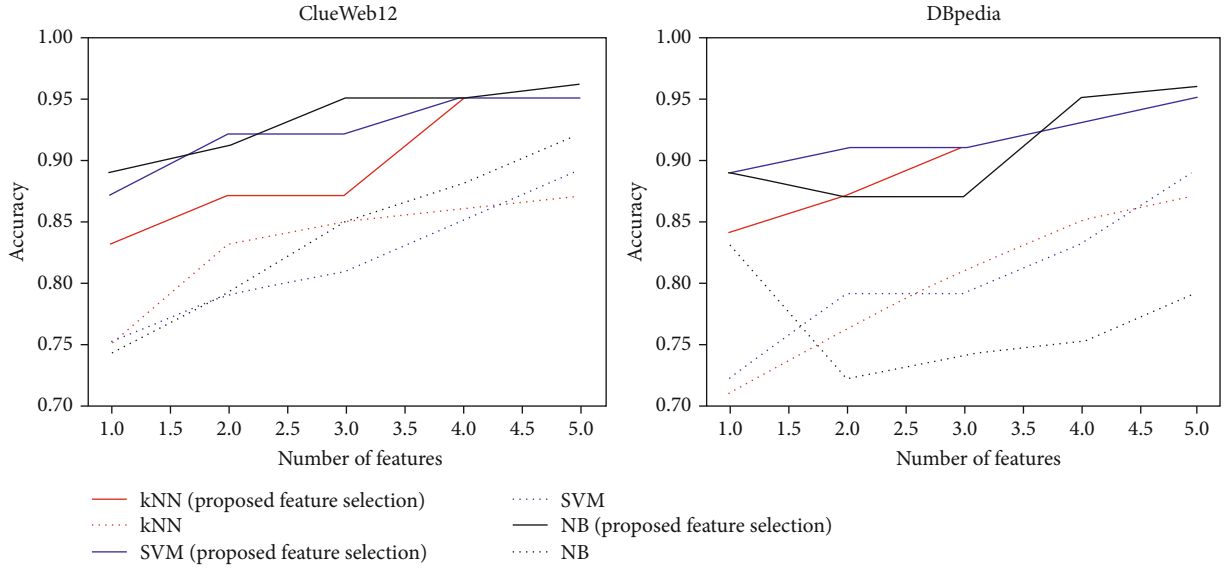


Figure 7: The purity of all the three classifiers when the number of classes = 7.

the immune system, small intestine, are related to health topics, while the other words in the documents such as group, major, have no relation with the real topic (health). Table 5 lists out a few recommended features; it can be easily noted that the features recommended by the proposed system have more correlation towards the class than the features that were recommended based on the high-frequency count. This is why additional information is given more weightage.

6. Conclusion

In this paper, we have proposed an embedded-based feature selection algorithm for recommending top N features to the machine learning model for predicting the correct class for a web document. Webpages in a corpus have links connecting each other; these links provide an additional advantage over the normal text document for classification. The links represent the coupling information, if a webpage has lots of links to a set of webpages, then it has high coupling—that means the target class also have a strong connection. The proposed feature selection algorithm in the paper makes use of this information for classification purposes. Along with the links, the proposed method considers the side information for improving classification accuracy. We have tested our classification model with two real-time benchmark datasets, and the results show that our work has a promising positive effect on classifying web documents. In future work, we will consider knowledge ontology and traffic connections as an additional parameter into account for classification.

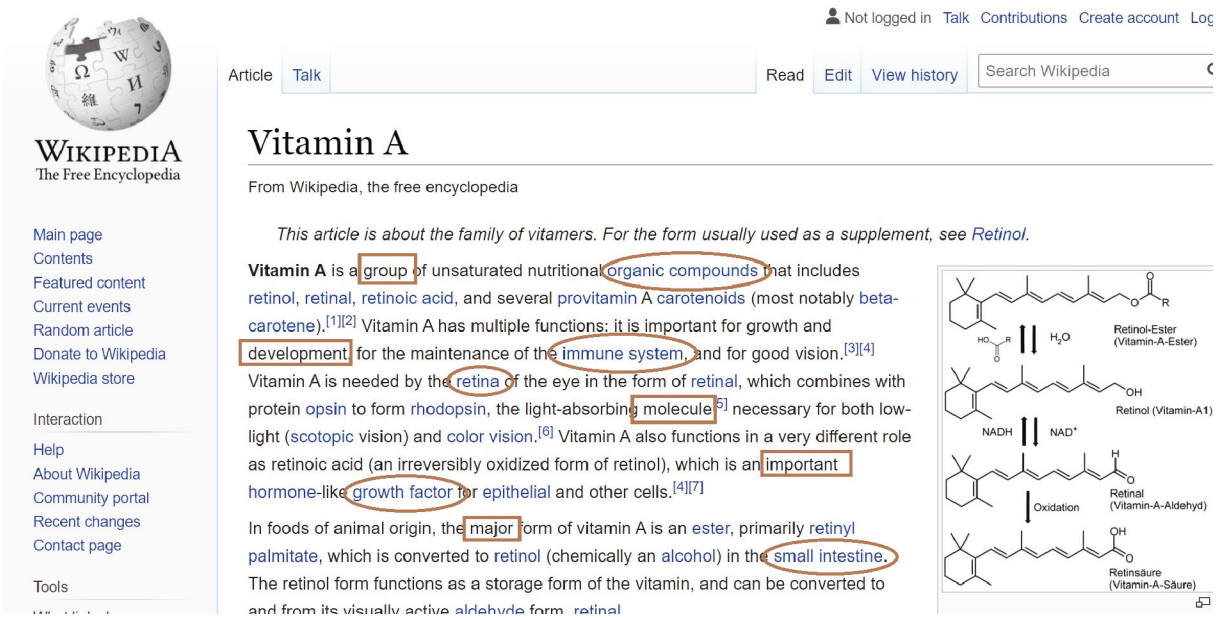


Figure 8: An example that demonstrates the importance of link text over normal text. The link text is circle bordered; normal text is rectangle bordered. It can be noted that circle bordered text talks more about the topic rather than rectangle bordered text.

Table 5: Some of the text features recommended in Figure 8 by existing (based on high-frequency count) and proposed (based on HEF and link text) algorithms.

#	High-frequency words	HEF and link text
1	Database	Medical
2	Reference	Supplementation
3	Functions	Metabolic
4	Reviews	Gene
5	Effects	Immune

Data Availability

Data available on request. The data are available by contacting Ashokkumar P (ashok05002@gmail.com).

Conflicts of Interest

The authors declare that there is no conflict of interest.

Authors' Contributions

P. Ashokkumar and G. Siva Shankar conceived the project, designed experiments, analyzed data and wrote the manuscript. R. Vinayakumar, Uttam Ghosh, Wathiq Mansoor, Waleed S. Alnumay analyzed data and edited manuscript.

Acknowledgments

This research work is supported by the project number (RSP-2020/250), King Saud University, Riyadh, Saudi Arabia. We thank all the reviewers and the special issue editor for proving valuable feedbacks which helped us to improve the quality of our research.

References

[1] K. L. Goh and A. K. Singh, "Comprehensive literature review on machine learning structures for Web Spam classification," *Procedia Computer Science*, vol. 70, pp. 434–441, 2015.

[2] B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *Journal of Advances in Information Technology*, vol. 1, no. 1, 2010.

[3] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec," *Information Sciences*, vol. 477, pp. 15–29, 2019.

[4] W. Bai, J. Ren, and T. Li, "Modified genetic optimization-based locally weighted learning identification modeling of ship maneuvering with full scale trial," *Future Generation Computer Systems*, vol. 93, pp. 1036–1045, 2019.

[5] L.-L. Li, J. Sun, M.-L. Tseng, and Z.-G. Li, "Extreme learning machine optimized by whale optimization algorithm using insulated gate bipolar transistor module aging degree evaluation," *Expert Systems with Applications*, vol. 127, pp. 58–67, 2019.

[6] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.

[7] X. Xu, H. Rong, E. Pereira, and M. Trovati, "Predatory search-based chaos turbo particle swarm optimisation (PS-CTPSO): a new particle swarm optimisation algorithm for web service combination problems," *Future Generation Computer Systems*, vol. 89, pp. 375–386, 2018.

[8] L. M. Francis and N. Sreenath, "Robust scene text recognition: using manifold regularized twin-support vector machine," *Journal of King Saud University - Computer and Information Sciences*, 2019.

[9] A. Alaei, P. P. Roy, and U. Pal, "Logo and seal based administrative document image retrieval: a survey," *Computer Science Review*, vol. 22, pp. 47–63, 2016.

[10] B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label arabic text categorization: a benchmark and baseline comparison of multi-label learning algorithms," *Information Processing & Management*, vol. 56, no. 1, pp. 212–227, 2019.

[11] T. Dogan and A. K. Uysal, "Improved inverse gravity moment term weighting for text classification," *Expert Systems with Applications*, vol. 130, pp. 45–59, 2019.

[12] S. Yang, R. Wei, J. Guo, and H. Tan, "Chinese semantic document classification based on strategies of semantic similarity computation and correlation analysis," *Journal of Web Semantics*, vol. 63, article 100578, 2020.

[13] Z. Chen, L. J. Zhou, X. D. Li, J. N. Zhang, and W. J. Huo, "The lao text classification method based on knn," *Procedia Computer Science*, vol. 166, pp. 523–528, 2020.

[14] P. Hao, J. Chiang, and Y. Tu, "Hierarchically svm classification based on support vector clustering method and its application to document categorization," *Expert Systems with Applications*, vol. 33, no. 3, pp. 627–635, 2007.

[15] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal sink node placement in large scale wireless sensor networks based on Harris' hawk optimization algorithm," *IEEE Access*, vol. 8, pp. 19381–19397, 2020.

[16] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, and M. Hassaballah, "A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery," *Computers & Chemical Engineering*, vol. 133, article 106656, 2020.

[17] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[18] Z. Ning, R. Y. K. Kwok, K. Zhang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[19] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, 2020.

[20] M.-J. Tsai, Y.-H. Tao, and I. Yuadi, "Deep learning for printed document source identification," *Signal Processing: Image Communication*, vol. 70, pp. 184–198, 2019.

[21] Z. Kastrati, A. S. Imran, and S. Y. Yayilgan, "The impact of deep learning on document classification using semantically rich representations," *Information Processing & Management*, vol. 56, no. 5, pp. 1618–1632, 2019.

[22] Z. Wu, H. Zhu, G. Li et al., "An efficient wikipedia semantic matching approach to text document classification," *Information Sciences*, vol. 393, pp. 15–28, 2017.

[23] M. Mittal, P. Siriaraya, C. Lee, Y. Kawai, T. Yoshikawa, and S. Shimojo, "Accurate spatial mapping of social media data with physical locations," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 9–12, Los Angeles, CA, USA, December 2019.

[24] P. Siriaraya, Y. Zhang, Y. Wang et al., "Witnessing crime through tweets: a crime investigation tool based on social media," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 568–571, Chicago, United States, 2019.

[25] S.-H. Hong, S.-K. Lee, and J.-H. Yu, "Automated management of green building material information using web crawling and ontology," *Automation in Construction*, vol. 102, pp. 230–244, 2019.

[26] L. K. Shih and D. R. Karger, "Using urls and table layout for web classification tasks," in *Proceedings of the 13th conference on World Wide Web - WWW '04*, pp. 193–202, New York, NY, USA, 2004.

[27] X. Zhu, X. Yang, C. Ying, and G. Wang, "A new classification algorithm recommendation method based on link prediction," *Knowledge-Based Systems*, vol. 159, pp. 171–185, 2018.

[28] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF∗IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011.

[29] "Vitamin a - wikipedia," 2020, https://en.wikipedia.org/wiki/Vitamin_A.

[30] S. Tan, "An effective refinement strategy for knn text classifier," *Expert Systems with Applications*, vol. 30, no. 2, pp. 290–298, 2006.

[31] J. Calvo-Zaragoza, J. J. Valero-Mas, and J. R. Rico-Juan, "Improving knn multi-label classification in prototype selection scenarios using class proposals," *Pattern Recognition*, vol. 48, no. 5, pp. 1608–1622, 2015.

[32] Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S.-S. Ho, "Forestexter: an efficient random forest algorithm for imbalanced text categorization," *Knowledge-Based Systems*, vol. 67, pp. 105–116, 2014.

[33] A. Sun, E.-P. Lim, and Y. Liu, "On strategies for imbalanced text classification using svm: a comparative study," *Decision Support Systems*, vol. 48, no. 1, pp. 191–201, 2009.

[34] S. Liu, K. Lee, and I. Lee, "Document-level multi-topic sentiment classification of email data with bilstm and data augmentation," *Knowledge-Based Systems*, vol. 197, p. 105918, 2020.

[35] P. Li, K. Mao, Y. Xu, Q. Li, and J. Zhang, "Bag-of-concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base," *Knowledge-Based Systems*, vol. 193, p. 105436, 2020.

[36] N. Phan, D. Dou, H. Wang, D. Kil, and B. Piniewski, "Ontology-based deep learning for human behavior prediction with explanations in health social networks," *Information Sciences*, vol. 384, pp. 298–313, 2017.