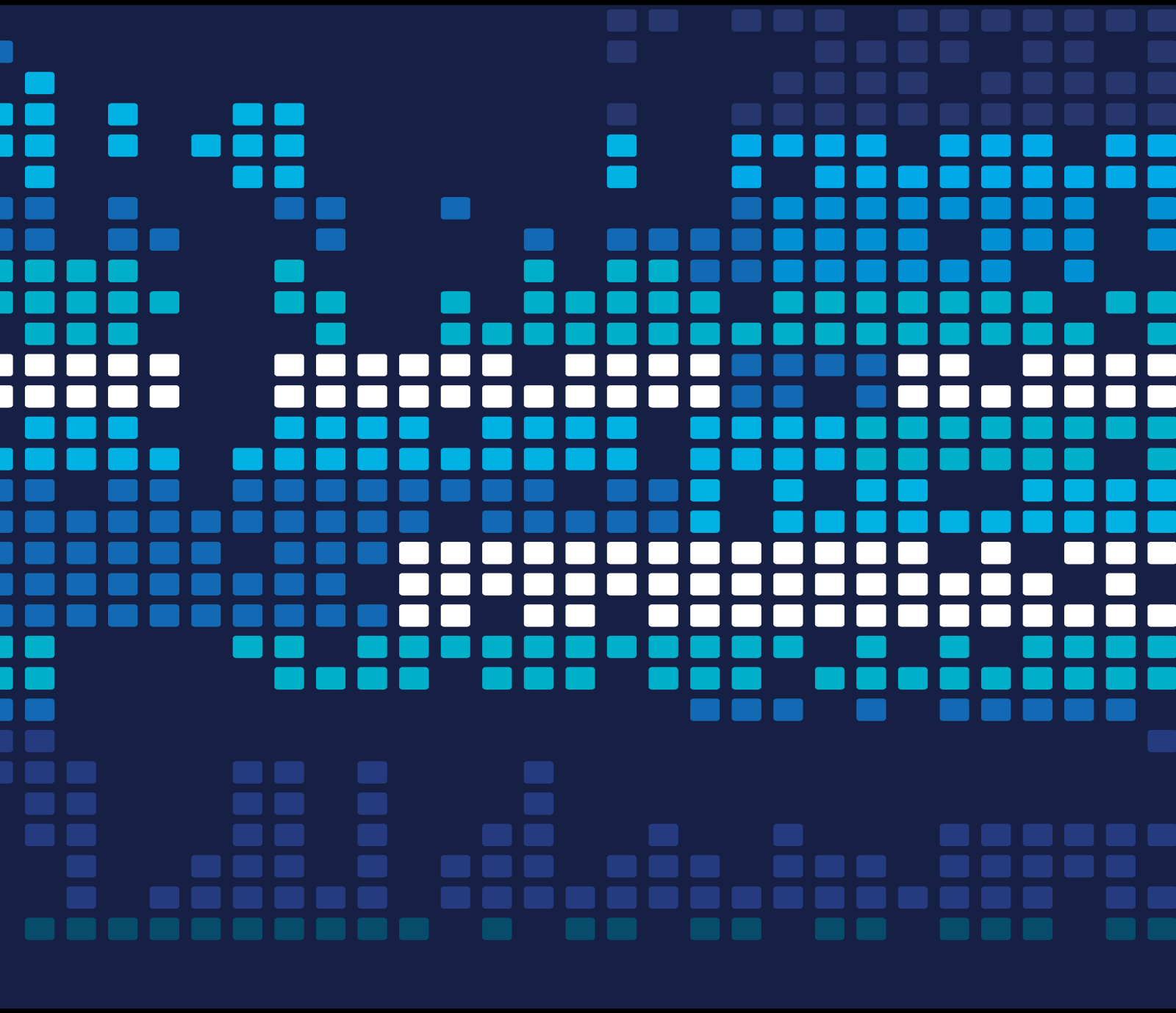
Scientific Programming

Scientific Programming for Multimodal Big Data

Lead Guest Editor: Liang Zhao
Guest Editors: Boxiang Dong, Qingchen Zhang, and Liang Zou



Scientific Programming for Multimodal Big Data

Scientific Programming for Multimodal Big Data

Lead Guest Editor: Liang Zhao
Guest Editors: Boxiang Dong, Qingchen Zhang, and Liang Zou

Contents

Hindawi
Scientific Programming
Volume 2021, Article ID 3225933, 14 pages
https://doi.org/10.1155/2021/3225933

Review Article

Exploration of Cross-Modal Text Generation Methods in Smart Justice

Yangqianhui Zhang

The University of British Columbia, School of Biomedical Engineering, Vancouver, Canada

Correspondence should be addressed to Yangqianhui Zhang; arielzhang2018@alumni.ubc.ca

Received 7 May 2021; Revised 9 August 2021; Accepted 22 September 2021; Published 21 October 2021

Academic Editor: Liang Zou

With the development of modern science and technology, information technology has brought great changes to many fields. Smart justice has become one of the increasing areas that people are paying more attention to. For example, large and small cases occur every day, and the legal library is continuously updated. Therefore, a large number of documents and evidence collection archives will bring tremendous pressure on the judiciary. The text generation technology can automatically present the results extracted from these redundant legal data and express the results of the analysis in natural language. It facilitates the business for huge amounts of legal data effectively, which relieves the work pressure of the judicial department. However, the text generation algorithms have not been promoted in justice. Therefore, this paper focuses on what benefits text generation can produce in law and how to apply text generation technology in legal field. The survey provides a comprehensive overview on text generation firstly, through summarizing the existing methods, that is, text to text, data to text, and visual to text. Then, we examine the process of the practical application of text generation in law. Furthermore, this paper puts forward the challenges and possible solutions to the judicial text generation, which provides pointers on future work.

1. Introduction

For a country, law maintains social stability. For each individual, law is a powerful weapon to defend people's rights and interests. As a result, the work of the legal sector is often arduous and onerous. According to statistics, the legal database has collected nearly one million pieces of provisions. It is conceivable that judges cannot memorize all laws and regulations, thus affecting the fairness and efficiency of judgments. In addition, in recent years, Chinese citizens have visited, consulted, and handled affairs on the website of the Ministry of Justice hundreds of millions of times, which indicates that the legal department needs to devote a lot of time, manpower, and material resources to solve people's problems. Text data processing is particularly important in many judicial services. Automatic generation of legal texts can alleviate the shortage of legal professionals. Through the automatic generation of legal texts, the paperwork of legal service personnel can be reduced, thus improving the efficiency of generating legal documents and avoiding the waste of judicial resources. With the gradual improvement of the society ruled by law, the requirements of judicial activities in China are getting higher and higher, so the generation of legal texts is of great significance to the judicial field.

Automatic text generation is a technique in which a computer generates natural language from some form of data content. Natural language generation technology rose since the 70s [1]. The template generation (template-based generation) is the first use of text automatic generation technology. After that, the schema generation technology (schema-based generation) and phrases planning technology (phrase/plan expansion) which are based on the theory of RST (Rhetorical Structure) and many other technologies gradually appeared.

There are quite a few frontier research works on legal text generation in NLP (natural language processing) and artificial intelligence fields. In recent years, there have been

some achievements and applications with international influence in this field. Text automatic generation is the main research direction in the field of natural language processing, and deep learning algorithms play an important role in the field of natural language processing. In recent years, more and more researchers have combined the technology with artificial intelligence, such as Microsoft's chatbot "Xiaobing," Headline's news robot "Zhang Xiaoming," and Tencent's "dream writer." At present, automatic text generation technology has been successively applied in entertainment, meteorology, medicine, news, and other fields [2–4].

However, the technology is not yet fully available in the judicial system, but the importance of text generation in law should not be underestimated. At present, judicial artificial intelligence can simply realize legal retrieval, document search, and so on. Besides, some intelligent legal software has been put into commercial use, in which text generation technology has made many contributions. For example, in the Competition on Legal Information Extraction in 2018, Tran et al. [5] used text generation technology to represent documents with abstracts and achieved the best performance. Then, they used the 2018 model as a pretrained phrase scoring model and lexical matching technology in the 2019 competition. The model combining text generation techniques performed well again in the legal case retrieval task. In commercial software products, Kira can be used to extract the terms of the contract; RAVN systems can efficiently summarize your legal documents; Lex Machina analyzes the historical data of the lawsuit for lawyers and generates a report; Lisa and Automio robots can generate agreements and legal documents based on questions and answers from users. These beneficial features are inseparable from text generation technology.

In addition, automatic text generation still has rosy prospect in the judicial system. If the automatic text generation technology is widely used in the law system, it will greatly improve the efficiency of the workflow of law, which is a promising opportunity. Examples include the following: (1) When people need legal advice or case inquiry, due to limited human and material resources, the human window may not be able to provide timely services. In the process of inquiry, there may be some questions that are too embarrassing to mention, which may lead to the ineffective and inaccurate progress of the case. In the process of solving problems, the staff may not be able to find the appropriate provisions in hundreds of thousands of legal provisions in a short time [6]. Intelligent dialog system based on text generation algorithms can solve the above problems. (2) At present, the public security department has presented "data police," which can make prediction and give warning according to police data [7]. Regular work reports are indispensable. Therefore, some content selection can be made on these data, and relevant reports can be generated automatically by text generation algorithm. (3) To meet the requirements of modern information management, text generation algorithm can convert traditional files in the form of picture and video into document format for storage. (4) In traditional sentencing, judges need to read a lot of documents, which requires a lot of time and energy. If the text generation technology is applied to extract and summarize the contents of files and indictments, it can not only save time but also realize transparent and fair handling of cases. The possible application of automatic text generation in law is not limited to this, but it can be seen that legal text generation is very promising.

This article aims to explore the necessity and possibility of automatic text generation in the judicial system. First, this article will classify and summarize the existing classic text generation algorithms from three different forms of input content: text input, data input, and visual input in Section 2. Then, we explain in detail how to apply these text generation algorithms to justice with the existing works and provide 6 authoritative and available legal datasets that can be used for text generation or other artificial intelligence tasks in Section 3. Section 4 analyzes the possible problems and feasible countermeasures in the application of text generation algorithm to justice, which gives new research directions for both text generation and intelligent law. At last, Section 5 concludes the paper.

2. Automatic Text Generation

According to different input, automatic text generation can be divided into three categories: text-to-text generation, data-to-text generation, and image-to-text generation [8]. Each technology here is extremely challenging, but with the rapid development of natural language generation technology and artificial intelligence, each technology has more detailed classification and cutting-edge application methods. Text-to-text generation is divided into text summarization and dialog system. Text summarization is divided into extraction and abstraction forms to express the central idea of the article. Dialog system is intended to generate the natural language of response and generates models in three modes: template-based, knowledge-based, and network-based. Data-to-text generation is mainly divided into two solutions: content selection and surface realization, and rule-based and data-based methods are adopted. For visual-to-text generation, image and video input are integrated into visual form input. It can be realized by template-based and network-based methods. In this section, we will discuss text generation techniques: text-to-text generation, data-to-text generation, and visual-to-text generation. Figure 1 gives an overview of the automatic text generation methods.

2.1. Text to Text. Text-to-text generation is a technology to convert the given text content into a new text. The process of this technology mainly includes text summarization, sentence compression, sentence fusion, and text retelling, which can be applied in the fields of information summarization, news writing, system dialog, and machine translation. This section mainly introduces text summary technology and intelligent dialog technology which can be employed to smart justice.

2.1.1. Text Summarization. Automatic text summarization utilizes computers to extract simple coherent text content from the original text, which can fully and accurately express

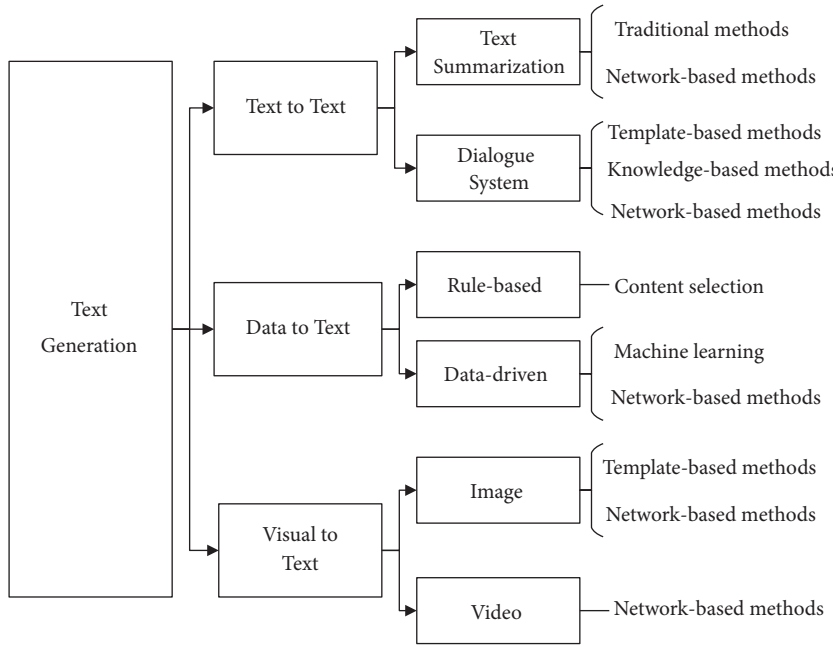


Figure 1: Text generation methods.

the central idea of the whole text. Summarization can be divided into extractive form and abstractive form: extractive form is composed of important sentences in the original text, while abstractive form is composed of new sentences. Traditional automatic text summarization is in extractive form.

In the original text summary methods, sentences were rated, sorted, and selected by word frequency, sentence position (first and last sentence), and keywords. Luhn proposed [9] to rate sentences according to the word frequency. The sentences with more frequent words have higher scores, and the sentences with higher final scores constitute the abstract of the text. This seemingly simple method sometimes has better effects than some complex methods [10]. In [11], Edmundson calculated the score of each sentence by integrating factors such as clue words, title, sentences at the beginning and end of paragraphs, and keyword frequency and selected sentences with high scores to form the abstract.

At the end of the twentieth century, machine learning emerged in text automatic summarization, making the process of summarization more intelligent. Inspired by Edmundson's idea, Kupiec added naive Bayesian classification model [12] to determine whether the extracted sentences meet the requirements of abstract. In 1999, Lin et al. applied the decision tree to the process of grading sentences and extracted the sentences with the highest scores to form an abstract. After that, Osborne [13] proposed an automatic text summarization method with a better extraction effect than the naive Bayesian model, which was based on the log-linear model and considered the relationship between different features.

In the twenty-first century, the emergence of neural networks made a breakthrough in text summarization technology. Kageback et al. [14] proved that the network-based text summarization method was significantly superior to other traditional methods. The automatic text summarization based on neural networks could generate the summarization of extractive form mentioned above or abstractive form [15]. The models can be divided into extraction model and abstraction model [16]. Among them, CNNs (convolutional neural networks) and RNNs (circular neural networks) were commonly used for neural-based abstracts, which were the basic models of many new technologies.

The extraction models focus on how to express sentences and how to choose the most suitable sentences. For example, CNNLM [17] employed convolutional neural network to represent sentences. Through training with noise contrast estimation, it can distinguish the real next word from the noisy word and select sentences based on the principle of optimizing submodule targets. This model can well process redundant information in candidate words. In [18], the method NN-SE utilized CNN and RNN to represent a sentence, which was input into the LSTM encoder. Thus, the LSTM decoder with sigmoid was used in grading, sorting, and choice of a sentence. Regarding the encoder and decoder, the stochastic gradient descent method was employed to minimize the negative logarithm likelihood. The contribution of this model is that the generation of abstracts no longer requires the manual language annotation process.

In [19], the SummaRuNNer was proposed to use a two-layer bidirectional RNN to represent sentences and documents, each of which was a bidirectional GRU. The model SummaRuNNer is shown in Figure 2. The blue part is the word-level representation, and the red part is the sentence-level representation. For each sentence representation, there is a 0, 1 label output indicating whether or not each sentence belongs to the summary. The second layer merges the sentence representation of the first layer into a document representation, in which the sentences are sorted using the

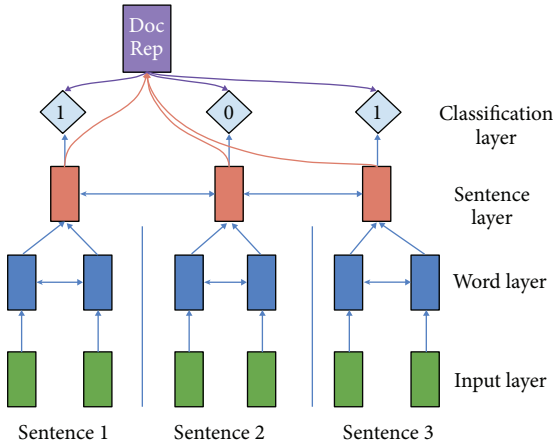


Figure 2: SummaRuNNer model.

sigmoid function. The training of this model is similar to the NN-SE model. Its advantage lies in the visualization of prediction, which makes the model intuitive and easy to understand. Moreover, its performance is comparable to that of some advanced depth models.

In the abstract summary model, the main consideration is how to represent the whole document in the encoder and how to generate sequence words through the decoder. For example, RAS-LSTM and RAS-Elman [20] used the encoder based on CNN and attention mechanism and used Elman RNN or LSTM model for decoding. The novel convolutional attention encoder of this model can ensure that the generation process of the decoder always focuses on the appropriate word input. Nallapati et al. [21] proposed a feature-rich hierarchical attention encoder based on two-way GRU to represent documents, in which one-way GRU, decoder based on LVT (the large vocabulary trick), and pointer switch mechanism were utilized. The innovation of this model is to model basic structures such as keywords, rare words, and word-to-sentence hierarchy, which will help improve the performance of the model. In [22], Pointer-Generator Networks adopted single-layer bidirectional LSTM as encoder and single-layer unidirectional LSTM as basic decoder and added pointer switch mechanism. On this basis, an overlay mechanism to punish repeated attention was also proposed. This model effectively solves the problem that the traditional sequence-to-sequence neural network model is prone to duplicate inaccurate content [23–25].

2.1.2. Intelligent Dialog. Automatic text generation in the dialog system refers to the natural language of the organization to generate responses based on the user's statement. Intelligent dialog system currently has three modes [26]: template-based, knowledge-based, and deep learning-based sequence-to-sequence generation model.

(1) Template-Based Models. This technique designs dialog templates for specific scenarios, and the text generation process is a template filling process [27]. The template-based model can accurately answer the questions in a certain field, but it has poor portability. It is suitable for the scenario of human assistant.

Apple's Siri uses template-based natural language generation. Siri employs the system's vocabulary to map surface words to related concepts, relationships, and properties, creating a dialog template that allows it to interact easily with users.

(2) Knowledge-Based Models. Based on an indexed dialog database, the user's statements are first analyzed using natural language processing (NLP) technology, and then fuzzy matching is performed in the statement database to select the response statements with the highest matching degree. This model is often used in entertainment chat and question-and-answer systems, and its knowledge base is easy to expand. However, when the amount of data is too large, the context is often not connected.

IBM's computerized question answering system, Watson, uses knowledge-based retrieval technology during the text generation stage [28]. After collecting large-scale evidence, Watson further analyzes and evaluates the answers. The system uses Deep QA architecture, which follows: (1) including more than one principle of assertion for the answers of fact, (2) searching for different resources for different understandings of the problem [29], and (3) achieving more than one candidate answer. After evaluation, scoring of each answer, the best answer is finally selected. Moreover, the complementarity of unstructured information and structured information is employed to improve the correctness of evidence analysis [30].

The architecture of Deep QA is extensible, in which Q&A tasks can be improved through the expansion of the knowledge base. However, the knowledge base is growing too fast to be updated in real time.

(3) Deep Learning-Based Models. Dialog generation based on deep learning does not rely on any template or knowledge base. This model is based on the end-to-end technology of deep learning, which acquires the ability of organization by learning natural language directly through a large amount of corpus. The resulting text is more flexible and intelligent.

Google [31] proposed a sequence-to-sequence framework to train their conversation engines. The model used end-to-end training patterns and backpropagation learning. The output of the conversation was based on the predicted sentences or sentences in the conversation. The completely data-driven approach can save a lot of manual overhead, but the model is capable of only simple conversations and lacks consistency before and after conversations.

Sordoni et al. [32] added context relation on the basis of the previous model and replaced the RNN model with multilayer forward neural network, so that the model could input context information and dialog information into encoder and maintain the dynamic consistency of input and output information. This context-aware approach also presents problems, such as adding distant content unnecessarily to the current generation process.

Kumar et al. [33] proposed a model of dynamic memory network, which used an episodic memory module to store context information and corpus based on HNN. Dialog is a

process of iterative attention; thus, the final text generation will be a hierarchical recursive reorder, resulting in a high-quality dialog generation.

It can be seen that neural network performs well in both text summarization and conversational system technology. However, when generating real sentences, there is a high probability of failure for two main reasons: (1) When using autoencoders to map sentences to their hidden representations, the representations of these sentences often occupy a small area of the hidden space. Therefore, most areas in the hidden space are not necessarily mapped to real sentences [34]. (2) Because of the nature of RNN itself, the error rate of sentence generation may increase greatly with the length of the sentence itself, which makes the quality of long sentences difficult to be guaranteed. In order to solve the above problems, in recent years, researchers pay more attention to how to generate more realistic sentences; they usually adopt the following methods: (1) using GANs (Generative Adversarial Networks) [35] frame to make the text more like human writing; (2) using reinforcement learning; (3) combining semantic or grammatical information to make the resulting sentences more correct [36].

The application of adversarial training can effectively improve the above problems by alternately updating discriminator and generator. Zhang et al. [34] proposed a method of adversarial training texts, which utilized LSTM as a generator and CNN as a discriminator. The generator constantly generates near-real sentences, and the discriminator aims to accurately distinguish the sentences generated by the generator from the real sentences. After adversarial training, the sentence was guaranteed to maintain high quality from a holistic perspective. In addition, Li et al. [37] have applied adversarial training to the neural dialog system, making the dialog generated by the intelligent dialog system almost indistinguishable from human language.

In 2019, Gao et al. [38] applied a GAN model to add text-related comment information. They chose the Seq2Seq model based on the attention mechanism and pointer mechanism as the generator and CNN as the discriminator. They used the content of the comments to get the main ideas and redundant information in the text.

Zhang et al. [39] used a more powerful generator, Transformer. Transformer is a completely attention-based model proposed by Google in 2017. It has achieved excellent performance in machine translation. Similarly, they chose CNN as the discriminator. The efficient parallelization of the Transformer framework has made their work a good result.

The GAN model is often combined with reinforcement learning. The GAN model alone cannot be applied to natural language generation, because the generated data of text is discrete, and the improvement of generator is effective for continuous data such as image based on discriminator information. However, for text data, the improved results are likely to correspond to invalid text information. Reinforcement learning has an inherent advantage in discrete data. It can use customized reward or punishment mechanisms to drive the final result more flexibly. Therefore, GAN-based text generation models usually use the policy gradient method in reinforcement learning during the training of the generator and discriminator. The above models of Gao et al. [38] and Zhang et al. [39] were designed as such. Of course, reinforcement learning itself makes a great contribution to natural language generation. Chen and Bansal [40] first used a deep learning model to extract important sentences and then used reinforcement learning to abstract the extracted text. Their model uses the idea of parallel decoding, which makes the decoding process very efficient.

In the dialog system, the process of dialog is like a decision-making process, so it can be fitted by the strategy learning process of reinforcement learning. Li et al. [41] used adversarial inverse reinforcement learning technology and provided a unique reward mechanism for the discriminator of the adversarial model, so the generator can obtain more accurate reward signals from it. Experiments have shown that their dialog system can produce high-performance responses.

In addition to the above two advanced methods, it is also an effective and feasible way to return to the semantic and grammatical structure of the text. Kouris et al. [42] proposed a new model combining deep learning and semantic data transformation in 2019. The principle of conversion is as follows: generalize the low-frequency words in the text into high-frequency words in the learning process, and then materialize the words in postprocessing. Based on this principle, the prediction of the model becomes more accurate.

Song et al. [43] expressed text as Abstract Meaning Representation (AMR) [44], which describes the grammatical structure of a sentence. They used a novel graph-to-text encoder. The traditional graph-to-text method is to traverse the nodes in the graph in a depth-first search or a breadth-first search. This method has disadvantages, which will cause the words that are close to each other to be far apart after traversal. Therefore, they use RNN to directly encode the graph into text, which solves the above problem well, and this parallel encoding method saves a lot of time.

The accuracy of text-to-text generation model still needs to be improved. The training of a text model usually requires a large corpus, and the training time of the model is very long (it may take several days). Therefore, an efficient text generation model is needed. Recently, many scholars have tried joint training on multiple documents. Fabbri et al. [45] applied a single-document model to multidocument text and found that combining methods such as Maximum Marginal Relevance (MMR) [46] is feasible. However, they just simply connect multiple documents, and the relationship between different documents is not considered. Effective multi-document-based text generation is also an important research direction in the future.

2.2. Data to Text. The generation of data to text is based on various data and tables, from which the internal structure and correlation are analyzed to form a smooth text. Ehud Reiter of the University of Aberdeen put forward the general framework of the data-to-text generation system [47], as shown in Figure 3. Firstly, the numerical data is input from the signal analysis module, and the basic patterns in the data

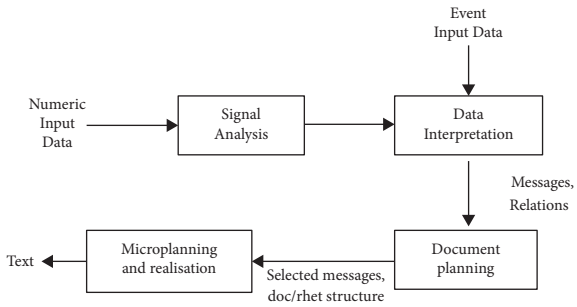


Figure 3: Data-to-text generation system.

are detected by various data analysis methods, which are output as discrete data patterns. The input of the data interpretation module is the basic events. By analyzing the basic patterns and input events, more complex and abstract messages are inferred, and their relationship is inferred. Finally, the high-level messages and the relationship between messages are output. Then, enter messages and relationships in the Document Planning module, analyze and decide which messages and relationships need to be mentioned in the text, at the same time determine the structure of the text, and finally output the messages and document structures that need to be mentioned. The last step is to input the selected message and structure in the Microplanning and Realization module and output the final text through natural language generation.

At present, this technology is mainly used in the fields of meteorological report, finance, sports, and medicine. There are two main problems that need to be solved in data-to-text generation: (1) how to choose effective data subset from the data obtained, which can be called content selection; (2) how to describe these data subsets in human language, which can be called surface implementation. The actual methods of data-to-text generation can be divided into rule-based methods and data-driven methods [48]. The relevant development context and research methods are introduced as follows.

2.2.1. Rule-Based Methods. Rule-based data text generation methods make the content selection and natural language representation of data according to expert knowledge in a certain field, which is suitable for specific fields, such as medicine and meteorology [49].

In the medical field, Hallett et al. [50] proposed a medical research method based on medical history information in 2006. The innovation of this method is to encode the information of clinical history into data and use the medical history data to generate text reports to support further clinical research. The approach also incorporates visual navigation tools to address the shortcomings of text generation. It aims to study cancer-related problems, but the method could be applied to other areas of medicine as well. In 2009, Gatt et al. proposed [51] "BabyTalk" system to generate the natural language summary of neonatal intensive care data. This system adopted the algorithm proposed in [50] to combine data with visualization and other technologies, to make the decision-making results more accurate.

Banaee et al. [52] developed a text generation method based on physiological sensor data in 2013. This method extracts information from the original data, performs data denoising and other processing, and uses expert knowledge to delete the value of the workpiece, to ensure that the system can generate text according to reliable signal input. In the natural language generation stage, the system uses correlation functions to order the importance of sentences and finally outputs robust text.

In the field of meteorology, Ramos et al. [53] proposed a meteorological service system "GALiWeather" in 2014, which took the weather data as the initial input and abstracted the data values into time-related language labels, namely, an intermediate code, through a computational method. Finally, the intermediate code was used as secondary input to generate natural language descriptions using an NLG system containing expert rules. They designed two NLG systems: One dealt with simple variables (cloud cover, wind, and temperature), in which language templates were defined. The other dealt with precipitation variables to prevent repetition, redundancy in the generated sentences. This method can guarantee high performance in content and form of text generation and can generate text description close to expert generation. However, the system is currently only applicable to the field of meteorology, with poor universality.

In 2016, Gkatzia et al. [54] developed two natural language generation systems, one based on "WMO (world meteorological organization)" and the other based on "NATURAL." Both systems provided text descriptions of precipitation and temperature, improving the accuracy of prediction. WMO is a rules-based system that can make predictions such as a 30 percent probability of rain, taking into account an interval of sunny days. The system can then generate the following text description: "it may be sunny, it may be rainy–less likely than impossible." The NATURAL system can imitate the tone and description of a weather forecaster. The rules used in this system come from the way in which observations (such as the BBC weather reporter) make predictions. For the same example above, the system obtains the following text description: "mainly dry and sunny."

The above methods can demonstrate that the rule-based data text generation needs the power of experts, and it can perform well in professional fields, but the applicability of the model is not wide. Moreover, rule-based methods often require a language template, which makes the generated text form too monotonous. Fortunately, the data-driven approach can improve both of these problems.

2.2.2. Data-Driven Methods. Data-driven text generation refers to the direct use of data for training, without the intervention of expert knowledge [49]. At present, data-driven methods have dominated natural language generation.

Liang et al. proposed a probabilistic generation model in 2009 [55], which can uniformly deal with the correspondence from segmentation text to description, fact identification, and data-to-text matching and solve the increasing ambiguity and noise in data. Inspired by this, Angeli et al. designed a new log-linear classifier in 2010 [56]. The whole text generation process is decomposed into several local decisions, which proved to have high performance in different fields such as sports and weather.

In 2014, Sowdaboina et al. proposed to utilize machine learning (ML) technologies to solve the problem of data content selection for the first time [57]. The model uses a mixture of natural language generation techniques and template-based methods to help the NLG system select text suitable for the application of templates, thus combining their respective strengths to produce high-quality text. The use of machine learning makes the rules of text generation closer to the human mind.

In the same year, Gkatzia et al. [58] introduced the feedback mechanism based on the content selection model in [57]. They compared and discussed the methods of multilabel classification and reinforcement learning (RL). The results showed that ML technologies can make the prediction results more accurate, while reinforcement learning is more exploratory.

In recent years, deep learning has achieved remarkable results in text summarization technology, and it also performs well in data-driven text generation. Mei et al. [59] proposed an end-to-end neural network model in 2016, which does not require the intervention of experts or rules. The model uses an encoder-allocator-decoder architecture and employs LSTM network unit as nonlinear encoder and decoder. In the model, the bidirectional LSTM-RNN encoder takes input from a set of event records and obtains the representation after modeling the dependencies that exist between the records in the database. The aligner of the model performs content selection using an extension of the alignment mechanism. This model can achieve satisfactory results even in fields where data is scarce.

In 2016, Lebret et al. [60] introduced a feedforward neural language model based on conditional neural language models, which can regulate text generation by tabular conditional language model and generate the sentences of people's biographies according to the fact tables in the dataset of people's biographies in Wikipedia. It copies and transfers words from fixed vocabularies and sample tables into output statements, which is a way to process large vocabulary data. The model has a good grasp of the tenses of the text, but some words need to be correctly predicted under a global condition. Overall, the model is able to generate fluent one-sentence descriptions of each character. However, generating longer descriptions is the problem that they have to tackle.

In 2019, Liu et al. [61] layered reinforcement learning frameworks to accommodate multimodal tasks. The model consists of multilevel strategy mechanism and multilevel reward mechanism. The first part aims to improve the accuracy of word level and sentence level, since the multilevel policy network can adaptively integrate word-level and sentence-level policies to generate each word. The second part guides the reward mechanism by combining image and language information. In order to better connect policies and rewards, they also designed novel optimization guidance items [61], as shown in Figure 4.

The difficulties of data-driven methods are mainly as follows: (1) There are high requirements for reliability and accuracy of data sources, which will directly affect the accuracy of the generated text. When dealing with large-scale data, the performance of the model decreases dramatically. (2) Efficiency is low when facing large-scale data. Wiseman et al. [48] employed a series of advanced neural methods and a simple template generation system to tackle document generation tasks. Experiments have shown that recent neural network models perform well in generating short textual descriptions of small amounts of data, but when faced with large-scale data, even with the ability to generate smooth text, text descriptions and human-generated documents still have a large difference. Puduppully et al. [62] found that if the content planning of data is carried out in advance, it can make a good combination of a large amount of data and deep learning model. They identified two questions before modeling to specify what to say and in what order. Experiments proved that they made a correct attempt, and the generated text had better conciseness and grammar. However, this method only improves the overall quality of the text, and more research is needed to accurately express the details. At present, there are not many generation models for large-scale data, so how to overcome the challenge brought by massive data is still a serious problem. To further advance data-driven text generation, both the bottlenecks must be addressed.

2.3. Visual to Text. With the popularization of all kinds of electronic products and the development of multimedia technology, a large quantity of pictures and video information is generated every day. If the multimedia information is accurately converted into descriptive text, the efficiency of classification and management can be greatly improved. The text generation work of image and video is rough as follows.

2.3.1. Image to Text. Image-to-text generation is a natural language description process after analyzing the visual content of image. There are two main ways to generate text from images: (1) The text can be generated through predefined generic language templates, during which the key attributes of images and other effective information are added. (2) Deep learning researchers generate descriptive sentences by using sequential generation models. These two generation methods are described below.

(1) Template-Based Generation. The template-based methods first use computer vision technology to identify the objects in the image, preset the template to be filled, and populate object relations and attribute labels into the template to generate the descriptive language of the image.

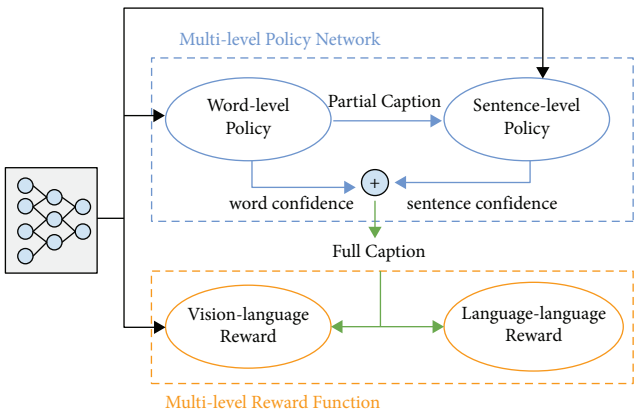


Figure 4: Multilevel policy and reward.

Farhadi et al. [63] first proposed the idea of cross-modal transformation from picture to text and studied the method based on language template. The model assumes that there are three spaces: image space, sentence space, and meaning space between them. The model uses triples (object, action, scene) for meaning representation. For sentence space, they use Curran & Clark parser [64] to generate the dependencies of each sentence and extract the subject-verb-object and other structures of the sentence, and then add them to the template of the sentence. By learning the mapping of image space and sentence space to the meaning space, measuring the similarity between them, and establishing the connection with the meaning space, the two-way conversion of image and text can be realized [65].

Kuznetsova et al. [66] proposed a new tree-based template method, which generated tree-structured phrase fragments by learning existing training sets, and then selectively combined these fragments to generate text descriptions. This model has a stronger generalization and generation ability than previous methods.

Yang et al. [67] utilized the hidden Markov model in the template-based method to fill in the template of sentence generation with the most likely predicted subject-object, verb, preposition, and other contents and finally output the natural language description of the image after decoding. The sentences generated by this model are more readable and relevant, but sometimes they are less predictive of nouns and verbs.

Language template-based methods tend to produce monotonous sentence patterns and content. In order to solve this limitation, deep learning-based coding-decoding methods are a better choice.

(2) Deep Learning-Based Generation. The implementation process of deep learning-based coding-decoding methods is naturally divided into two parts: The coding process is designed to extract visual features, generally using deep neural network, CNN. In the decoding process, extracted features are used as input, and natural text describing image is generated by using RNN or LSTM model. Coding-decoding methods are the applications of deep learning in image text generation, which often combine some different fusion methods, attention mechanism, or reinforcement learning [67] to generate more diverse sentences.

Mao et al. [68] first proposed an image text generation model m-RNN based on neural network. In addition to the CNN-based visual feature extraction part and language modeling part, this model also has a multimodal part, which connects the language model and CNN through a layer representation in m-RNN. m-RNN model can not only complete the image-to-text generation, but also solve the problem of sentence and image retrieval.

Fang et al. [69] proposed a new image generation text model, which consisted of three main parts: (1) visual detector, used to identify high-frequency words in image titles; (2) language model, adopting the CNN structure, which is used for the statistics of the related information of words and the generation of natural language; (3) a multimodal similarity model, which is for reordering words. The model is directly studied in the title text of the image, which combines the image content to obtain words of various parts of speech, ensuring that the generated text contains these words. Its global semantic result is the best in the official benchmark test.

Xu et al. [70] added attention mechanism to the model. It used convolutional neural network as an encoder to extract feature vectors of images and used long and short time memory networks in the decoder. The generated position of each word was determined according to the context vector, past hidden state, and position of previously generated words. The mechanism of the attention model allows the algorithm to selectively focus on certain areas of the image, thus visually selecting important parts.

Zhou et al. [71] proposed a special attention-based approach, which focused on words in the text, as opposed to the traditional approach focusing on part of the image. The model uses td-gLSTM (time-dependent gLSTM) method to generate attention guidance signal, which guides LSTM to generate descriptive natural language.

In recent years, reinforcement learning has become a hot topic in machine learning. In 2017, Zhang et al. [72] applied reinforcement learning to the process of image text generation. The model uses actor-critic method to train, thus it can improve the matching between training results and prediction results through the mechanism of reward and punishment. In the same year, Ren et al. [73] developed a new decision-making framework, which used the "strategy network" and "value network" in reinforcement learning to generate texts collaboratively.

However, using only the reward and punishment mechanism in reinforcement learning and the strategy network to generate text images is still unsatisfactory. Multitask learning vision and language pose a challenge to generation. In 2019, Liu et al. [61] layered reinforcement learning frameworks to accommodate multimodal tasks. The model consists of a multilevel strategy mechanism and a multilevel reward mechanism. The first part aims to improve the accuracy of both the word level and the sentence level, and the second part guides the reward mechanism by combining images and language information. In order to better bridge strategies and rewards, they also designed a

novel optimization guidance item. Aiming at solving the problem of multimodal learning, Nguyen et al. [74] also added detailed natural language descriptions of objects based on title information and combined the mixed end-to-end CNN-LSTM model to effectively solve the two problems of natural language generation and object retrieval of object titles.

Although image-to-text generation methods are constantly being innovated, there are still many problems to be improved, such as the immature image feature extraction technology, the semantic gap between image and text, and the cross-language description of images [75].

2.3.2. Video to Text. Early video-to-text generation works depended on the manual operation of the video feature extraction and modeling tasks [76, 77]. After that, more and more research was proposed. In 2015, Xu et al. [78] designed a new discriminative CNN to learn video representation for event detection. However, this model ignores the time structure of video. In order to solve the problem, Ballas et al. [79] proposed GRU-RCN algorithm, which considered video time and space feature information. It can obtain more refined video motion information in order to reduce the bad influence brought by high-dimensional video reproduction.

Pan et al. [80] also proposed a hierarchical recursive neural encoder (HRNE) to generate text for video, aiming at the integration of time information in video. The hierarchical structure enables video information to be better expressed, and the higher part of the model can make full use of the time structure and can be transformed at different granularity of time. In addition, the HRNE model has promising flexibility and nonlinearity, but the generalization ability of the model needs to be improved.

The above models are only used to generate a few sentences of short video. Yu et al. [81] first attempted to use deep learning method to generate multiple sets of statements or paragraphs for long video in 2016. They proposed a framework based on RNN structure to generate video paragraph text. The framework consists of a sentence generator and a paragraph generator. The paragraph generator models the relationships of simple sentences generated by the sentence generator. This algorithm has achieved favorable results in two large datasets, YouTubeClips and TACoS-MultiLevel, but the model is unable to process very small objects in video. Besides, the error superposition may occur due to the unilateral nature of the sentences generated by the model. All these problems need to be solved.

For video's multimodal features, many current models simply connect the features of video with different modes. Xu et al. [82] focused on the characteristics of video's multimodal features and proposed a multimodal attention span memory neural network (MA-LSTM) model. LSTM encoders and decoders are used in the model. Because video has multimodal characteristics, three LSTM models are built to encode video frames, video motion, and audio, and then they are fused to form multimodal flows, which are then output from the decoder. A multilevel attention mechanism is added to enhance the flexibility and effectiveness of modal integration. Compared with the advanced video-to-text generation algorithms GUR-RCN and HRNE, this algorithm has more obvious advantages and is a more successful network model.

3. Application of Text Generation in Smart Justice

This section will discuss the practical application of text generation in justice with existing generation models. Prior to this, we will introduce 6 authoritative legal datasets that can be used for text generation, such as judgment prediction and clerical generation. Of course, they can also be used for other intelligent judicial tasks.

3.1. Legal Case Reports Dataset. (https://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports)The dataset was provided by the Federal Court of Australia (FCA). It includes all the legal cases of the Federal Court from 2006 to 2009. For each document, the dataset contains its catchphrases, citations sentences, citation catchphrases, and citation classes. These data can be used for automatic text summarization and citation analysis.

3.2. Department of Justice Open Data. (https://www.justice.gov/open/open-data) US Department of Justice published a list of legal data publicly online on November 30, 2013, so this dataset is a high-quality open dataset. It includes specific databases such as violent crime cases, FBI crime reports, and statistical reports.

3.3. The Supreme Court Database. (http://scdb.wustl.edu/) The database comes from the US Supreme Court and has absolute authority. The data records cases of court judgments from 1791 to 2018. Each case contains the legal provisions referenced by the case and many details at the time of the decision.

3.4. Caselaw Access Project (CAP). (https://case.law/) The database contains 360 years of various judgment cases in the United States, which have been digitally obtained from the collections of the Harvard Law Library. The cases have been organized into a unified form. A total of 1,693,904 different cases have been collected.

3.5. Bureau of Justice. (https://www.bjs.gov/index.cfm?ty=dca) The data source is provided by the Bureau of Justice Statistics and contains data on some US law enforcement agencies, prisons, parole, and probation. This data is essential to improve the efficiency of legal offices and effectively help fight crime.

3.6. CAIL2018. (https://github.com/thunlp/CAIL2018) CAIL2018 [83], the first large-scale legal dataset for judgment prediction in China, is derived from the website of adjudication documents. The dataset includes 2676,075 legal cases, all published by the Supreme People's Court. Each

case includes a description of the facts of the case and the outcome of the judgment, which is embodied in the relevant legal provisions, the predicted charges, and the sentence. This dataset is very large and very well annotated.

It can be found that most of these legal datasets are composed of text, so text-to-text generation technology plays a vital role in the generation of legal texts, which is also the current research content of most researchers. However, the potential contribution of data-to-text and visual-to-text technologies to legal work cannot be ignored.

3.7. Application of Text-to-Text Generation. Automatic text-to-text generation technology can be applied to intelligent extraction and intelligent dialog in smart justice. The application of text summary technology to the reading and summary of case documents can relieve the pressure of judges and reduce the errors caused by human operations. For example, in order to better solve the issue of appealing for disability benefits for veterans, Zhong et al. [84] hope to extract important sentences from cases as summarization. The abstracts can help the Board of Veterans' Appeals (BVA) to make more accurate decisions on cases. They used a corpus of about 35,000 BVA cases on disability compensation for posttraumatic stress disorder (PTSD). The authors used the idea of train-attribute-mask pipeline, sentence type classifier, and MMR technology successively to select summary sentences with a priority prediction function and finally embedded the generated sentences into a template. The selection of an advanced abstract neural network model is the key step for intelligent extraction. During the generation of abstract, additional modeling or attention can be paid to such important information as time and place.

The retrieval model based on judicial knowledge base can be used in the intelligent dialog system of judicial domain. Firstly, a complete judicial law knowledge database is built, which can be expanded or deleted according to the modification of laws and regulations, and the appropriate algorithms are selected to evaluate the matching statements. Finally, the response statements are generated. Governatori et al. [85] extended an existing dialog framework into the legal field. They used the framework to model the process of dialog in legislative deliberations. For more flexible questions, deep learning-based dialog system can be used to answer.

3.8. Application of Data-to-Text Generation. Data-to-text automatic generation technology can be applied to intelligent report generation in smart justice. Usually, the legal system creates files for each criminal, records the occurrence of some cases, etc. Thus, we can establish a database of this content and make corresponding structural selection. For example, using one kind of criminal event or a certain period of time of the case records, the natural language description can be generated automatically based on data-driven text generation algorithms.

GAN is improved by Kang et al. [86]. The encoder-decoder model based on LSTMs is used as the generator, and the binary classification module based on CNN is used as the discriminator. Through the real legal documents of divorce cases and through the data-driven method, a total of 25,000 case report datasets were preprocessed by word segmentation. Finally, through comparison, it is concluded that the text index of case description generated by this model has a good effect.

3.9. Application of Visual-to-Text Generation. Automatic visual-to-text generation technology can be applied to intelligent storage in smart justice. With the gradual informatization of legal systems, document storage format is no longer the traditional JPG, PNG, MPEG, MP4, and other forms of pictures and video; they need to be expressed into text. Image files can use the infrastructure of encoder and decoder in deep learning to generate natural language descriptions. Kang et al. [86] constructed a deep learning network model, ED-GAN, which is suitable for automatic generation of legal texts and applied the model to the generation of legal case description. At the same time, the discriminator model based on CNN can improve the accuracy of the generated text and form a competitive confrontation with the real text. The method can generate the case description text for a long time through the network-based method. The experimental results show that ED-GAN model has a good effect in generating case description text. If necessary, other technologies should be combined to enhance the learning ability of images. In judicial work, a video, which is monitored for a long time and has a lot of redundant information, is generally processed. Therefore, in addition to video-to-text generation model with good multimodal characteristics, attention mechanism is often needed.

4. Challenges of Text Generation in Smart Justice

In this section, we further discuss the text generation techniques according to the characteristics of judicial text and judicial work and locate the problems and challenges in their applications in judicial work.

(1) The text generation algorithms cannot yet be used to solve complex problems in smart justice. For example, in the existing intelligent consulting service, the intelligent dialog systems are realized by text-to-text generation. However, the existing dialog systems are not perfect enough. When dealing with complex problems, manual services are still needed. This indicates that the current natural language generation models are not fully capable of thinking like a human brain. We look forward to the day when computers can be answered like humans, which is not just simple and mechanical. However, this requires further development of artificial intelligence in text generation for smart justice.

(2) The performances of the existing techniques have not met the standards required by law. From the characteristics of judicial text, it is different from other

texts. In essence, the law is the highest standard of conduct used to regulate and constrain the whole society, which is formulated or recognized by the state and guaranteed by the state's coercive force. It has supreme authority and prescriptiveness. Concreteness, accuracy, simplicity, preciseness, and specification are the standards of wording in legal texts [87]. In the previous section, some neural network-based text generation algorithms were summarized. However, due to the inherent nature of the model, the high quality of sentences cannot be guaranteed when generating long sentences. Even if adversarial training was used, it can only improve the overall quality of the sentence.

To tackle this, the corpus should be accurately classified or extracted for keywords before sentence training, and the idea of keyword coverage should be used for modeling. Thus, promising results may be achieved. However, judicial text generation should improve its efficiency as much as possible in the links of input, training, and output. Therefore, the study of high-quality text generation technique is another promising area in smart justice.

(3) The generation of judicial text needs to standardize the wording and format. Specifically, legal terms have a single meaning [61], and each term represents a specific legal concept, which cannot be arbitrarily replaced when used. For example, "alimony" refers to "alimony for divorce," which cannot be replaced with "payment," even though in reality the terms are similar. Besides, legal terms also have opposite meanings [87]; namely, many terms come in pairs with contradictory meanings, such as plaintiff and defendant, actor and victim. Therefore, it is necessary to accurately grasp the subject and object in the text, and there must be no situation where the host and the guest are upside down [88].

In the generation of judicial texts, their characteristics should be fully considered, and the terms in corpus should be used accurately. A semantic-driven approach can be used to study judicial documents. This model should consider the complex structure and semantic knowledge of judicial texts to enhance the application effect in law. Besides, before using the text generation model, a domain knowledge model of judicial documents should be constructed. The more accurate the knowledge model is built, the more effective the results will be. Therefore, the construction of the domain knowledge model is pretty important in smart justice.

(4) The size of the data generated by judicial texts is huge. There are nearly hundreds of thousands of laws and regulations that need to be entered into the system. At present, tens of millions of judicial documents have been published. Different from meteorological and news fields, the storage of judicial data needs to be more complete and lasting, which proposes certain requirements for its storage technology. Moreover, the current text generation techniques are not good at large-scale data, especially in data-driven text generation.

Therefore, the text generation models should be combined with some advanced caching technologies to solve the storage problem of a large amount of text data. For example, the extension mechanism based on replication and reconstruction can effectively improve the neural network system of a large amount of data, but the overall efficiency is still limited. Thus, more research is needed to make a significant breakthrough.

(5) Models need interpretability. Many models of artificial intelligence are like black boxes, which may produce correct but abstract results. If the results of a model are not well explained, they may not be convincing, especially in the serious and infallible field of law. Keppens et al. [89] used Bayesian networks in legal decision making. Encouragingly, their model can well explain the production results, coupled with the probabilistic rationality of the Bayesian network. This will be an acceptable one in the legal field model.

Thus, if the text generation technology is combined with knowledge such as mathematical statistics or given a reasonable explanation for each process in machine learning, advanced models will be better promoted in law.

Text generation technology needs to integrate the research results of natural language processing, machine learning, cognitive science, and other fields, and it has very high research value and prospects. However, smart justice has great challenges in text generation because of its complex problems, strict wording standards, and huge data specifications. Therefore, in this case, this paper proposes a cross-modal legal text generation direction as a future research opportunity. Combined with text, data, and visual analysis, more accurate text can be generated to meet the filing requirements of judicial documents.

5. Conclusion

In this paper, we put forward the importance of text generation technology in the intellectualization of judicial system and then summarize the current text generation techniques according to text input, data input, and visual information input. After that, we propose how to apply these techniques to the actual judicial text generation. Particularly, the intelligent dialog system and text summary technology can be employed to intelligent consultation and intelligent extraction in smart justice. Moreover, data-driven text generation can be used to automatically generate judicial reports. The generation of image, video, and text can meet the requirements of judicial document filing. Finally, we discuss the text generation techniques according to the characteristics of judicial text and judicial work and locate the problems and challenges in their application to judicial work.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] P. Br&Dotzillon, "Context in problem solving: a survey," *The Knowledge Engineering Review*, vol. 14, no. 14, pp. 47–80, 1999.

[2] B. Lavoie and O. Rainbow, "A fast and portable realizer for text generation systems," in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC, USA, March 1997.

[3] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, vol. 1, pp. 285–295, Hong Kong, China, May 2001.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[5] V. Tran, M. L. Nguyen, and K. Satoh, "Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 275–282, Montreal, QC, Canada, June 2019.

[6] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2010.

[7] M. R. Keyvanpour, M. Javideh, and M. R. Ebrahimi, "Detecting and investigating crime by means of data mining: a general crime matching framework," *Procedia Computer Science*, vol. 3, pp. 872–880, 2011.

[8] E. Loper and S. Bird, "Nltk: The Natural Language Toolkit," https://arxiv.org/abs/cs/0205028.

[9] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[10] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," *Advances in Automatic Text Summarization*, pp. 111–121, MASS, Amherst, MA, USA, 1999.

[11] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[12] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," *Advances in Automatic Summarization*, pp. 55–60, MASS, Amherst, MA, USA, 1999.

[13] M. Osborne, "Using maximum entropy for sentence extraction," in *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pp. 1–8, Association for Computational Linguistics, PA, USA, July 2002.

[14] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive summarization using continuous vector space models," in *Proceedings of the 2nd Workshop on Continuous Vector Space Models and Their Compositionality (CVSC)*, pp. 31–39, Gothenburg, Sweden, April 2014.

[15] A. Fiori, *Trends and Applications of Text Summarization Techniques*, IGI Global, Hershey, PA, USA, 2020.

[16] Y. Dong, "A survey on neural network-based summarization methods," https://arxiv.org/abs/1804.04589.

[17] W. Yin and Y. Pei, "Optimizing sentence modeling and selection for document summarization," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, July 2015.

[18] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," https://arxiv.org/abs/1603.07252.

[19] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: a recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[20] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 93–98, CA, USA, June 2016.

[21] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, and G. Cagler, "Abstractive text summarization using sequence-to-sequence rnns and beyond," https://arxiv.org/abs/1602.06023.

[22] A. See, P. J. Liu, and C. D. Manning, "Get to the point: summarization with pointer-generator networks," https://arxiv.org/abs/1704.04368.

[23] Q. Zhang, C. Bai, L. T. Yang, Z. Chen, P. Li, and H. Yu, "A unified smart Chinese medicine framework for healthcare and medical services," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, 2021.

[24] Q. Zhang, C. Bai, Z. Chen et al., "Deep learning models for diagnosing spleen and stomach diseases in smart chinese medicine with cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 4, Article ID e5252, 2019.

[25] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "Incremental deep computation model for wireless big data feature learning," *IEEE Transactions on Big Data*, vol. 6, no. 2, 2020.

[26] R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "On the evaluation of dialogue systems with next utterance classification," https://arxiv.org/abs/1605.05414.

[27] S. S. Mohamad, N. Salim, and M. N. Jambli, "Service chatbots: a systematic review," *Expert Systems with Applications*, vol. 184, Article ID 115461, 2021.

[28] D. Ferrucci, E. Brown, J. C. Carroll et al., "Building watson: an overview of the deepqa project," *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[29] A. Kalyanpur, S. Patwardhan, B. Boguraev, A. Lally, and J. C. Carroll, "Fact-based question decomposition in deepqa," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 13–21, 2012.

[30] A. Kalyanpur, B. K. Boguraev, S. Patwardhan et al., "Structured data and inference in deepqa," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 10:1–10:14, 2012.

[31] O. Vinyals and Q. Le, "A neural conversational model," https://arxiv.org/abs/1506.05869.

[32] A. Sordoni, M. Galley, M. Auli et al., "A neural network approach to context-sensitive generation of conversational responses," https://arxiv.org/abs/1506.0671.

[33] A. Kumar, O. Irsoy, P. Ondruska et al., "Ask me anything: dynamic memory networks for natural language processing," in *Proceedings of the International Conference on Machine Learning*, pp. 1378–1387, NY, USA, June 2016.

[34] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *Proceedings of the NIPS workshop on Adversarial Training*, vol. 21, Barcelona, Spain, December 2016.

[35] I. Goodfellow, J. A. Pouget, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, pp. 2672–2680, MIT Press, Cambridge, MA, USA, 2014.

[36] M. Jang, "Sentence transition matrix: an efficient approach that preserves sentence semantics," *Computer Speech & Language*, vol. 71, Article ID 101266, 2021.

[37] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," https://arxiv.org/abs/1701.06547.

[38] S. Gao, X. Chen, P. Li et al., "Abstractive text summarization by incorporating reader comments," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6399–6406, Honolulu, HI, USA, February 2019.

[39] C. Zhang, C. Xiong, and L. Wang, "A research on generative adversarial networks applied to text generation," in *Proceedings of the 2019 14th International Conference on Computer Science & Education (ICCSE)*, pp. 913–917, IEEE, Toronto, ON, Canada, August 2019.

[40] Y. C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," https://arxiv.org/abs/1805.11080.

[41] Z. Li, J. Kiseleva, and M. D. Rijke, "Dialogue generation: from imitation learning to inverse reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6722–6729, Honolulu, HI, USA, February 2019.

[42] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Abstractive text summarization based on deep learning and semantic content generalization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5082–5092, Florence, Italy, July 2019.

[43] L. Song, Y. Zhang, Z. Wang, and D. Gildea, "A graph-to-sequence model for amr-to-text generation," https://arxiv.org/abs/1805.02473.

[44] L. Banarescu, C. Bonial, S. Cai et al., "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 178–186, Sofia, Bulgaria, August 2013.

[45] A. R. Fabbri, I. Li, T. She, S. Li, and D. R. Radev, "Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model," https://arxiv.org/abs/1906.01749.

[46] J. G. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," *SIGIR*, vol. 98, pp. 335-336, 1998.

[47] E. Reiter and R. Dale, *Building Natural Language Generation Systems*, Cambridge University Press, Cambridge, 2000.

[48] S. Wiseman, S. M. Shieber, and A. M. Rush, "Challenges in data-to-document generation," https://arxiv.org/abs/1707.08052.

[49] F. O. Asahiah, "Comparison of rule-based and data-driven approaches for syllabification of simple syllable languages and the effect of orthography," *Computer Speech & Language*, vol. 70, Article ID 101233, 2021.

[50] C. Hallett, R. Power, and D. Scott, *Summarisation and Visualisation of E-Health Data Repositories*, UK E-Science All-Hands Meeting, Nottingham, UK.

[51] A. Gatt, F. Portet, E. Reiter et al., "From data to text in the neonatal intensive care unit: using nlg technology for decision support and information management," *Ai Communications*, vol. 22, no. 3, pp. 153–186, 2009.

[52] H. Banaee, M. U. Ahmed, and A. Loutfi, "Towards nlg for physiological data monitoring with body area networks," in *Proceedings of the 14th European Workshop on Natural Language Generation*, pp. 193–197, Sofia, Bulgaria, August 2013.

[53] A. S. Ramos, A. J. Bugarin, S. Barro, and J. Taboada, "Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 44–57, 2014.

[54] D. Gkatzia, O. Lemon, and V. Rieser, "Natural language generation enhances human decision-making with uncertain information," https://arxiv.org/abs/1606.03254.

[55] P. Liang, M. I. Jordan, and D. Klein, "Learning semantic correspondences with less supervision," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 91–99, Association for Computational Linguistics, Suntec, Singapore, August 2009.

[56] G. Angeli, P. Liang, and D. Klein, "A simple domain-independent probabilistic approach to generation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 502–512, Association for Computational Linguistics, MA, USA, October 2010.

[57] P. K. V. Sowdaboina, S. Chakraborti, and S. Sripada, "Learning to summarize time series data," in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 515–528, Springer, Kathmandu, Nepal, April 2014.

[58] D. Gkatzia, H. Hastie, and O. Lemon, "Comparing multi-label classification with reinforcement learning for summarisation of time-series data," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1231–1240, MD, USA, June 2014.

[59] H. Mei, M. Bansal, and M. R. Walter, "What to talk about and how? selective generation using lstms with coarse-to-fine alignment," https://arxiv.org/abs/1509.00838.

[60] R. Lebret, D. Grangier, and M. Auli, "Neural text generation from structured data with application to the biography domain,".

[61] A. Liu, N. Xu, H. Zhang, W. Nie, Y. Su, and Y. Zhang, "Multi-level policy and reward reinforcement learning for image captioning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pp. 821–827, Stockholm, Sweden, July 2018.

[62] R. Puduppully, L. Dong, and M. Lapata, "Data-to-text generation with content selection and planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6908–6915, Honolulu, HI, USA, February 2019.

[63] A. Farhadi, M. Hejrati, M. A. Sadeghi et al., "Every picture tells a story: generating sentences from images," in *Proceedings of the European Conference on Computer Vision*, pp. 15–29, Springer, Crete, Greece, September 2010.

[64] J. R. Curran, S. Clark, and J. Bos, "Linguistically motivated large-scale nlp with c&c and boxer," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 33–36, Association for Computational Linguistics, Prague Czech Republic, June 2007.

[65] J. Liu, M. Yang, C. Li, and R. Xu, "Improving cross-modal image-text retrieval with teacher-student learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3242–3253, 2021.

[66] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "Treetalk: composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351–362, 2014.

[67] Y. Yang, C. L. Teo, H. Daumé III., and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 444–454, Association for Computational Linguistics, Edinburgh United Kingdom, July 2011.

[68] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn," https://arxiv.org/abs/1412.6632.

[69] H. Fang, S. Gupta, F. Iandola et al., "From captions to visual concepts and back," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1473–1482, Boston, MA, USA, June 2015.

[70] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning*, pp. 2048–2057, Atlanta GA USA, June 2015.

[71] C. Zhou, J. Bai, J. Song et al., "Atrank: An attention-based user behavior modeling framework for recommendation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, LA, USA, February 2018.

[72] L. Zhang, F. Sung, F. Liu et al., "Actor-critic sequence training for image captioning," https://arxiv.org/abs/1706.09601.

[73] Z. Ren, X. Wang, N. Zhang, X. Lv, and L. J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 290–298, HI, USA, July 2017.

[74] A. Nguyen, Q. D. Tran, T.-T. Do, I. Reid, D. G. Caldwell, and N. G. Tsagarakis, "Object captioning and retrieval with natural language," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Seoul, Korea (South), October 2019.

[75] K. Barnard, P. Duygulu, and D. Forsyth, "Clustering art," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, vol. Vol. 2, IEEE, HI, USA, December 2001.

[76] H. Wang, A. Kläser, C. Schmid, and L. Cheng, "Action recognition by dense trajectories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, Colorado Springs, CO, USA, June 2011.

[77] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, Sydney, Australia, December 2013.

[78] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative cnn video representation for event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1798–1807, MA, USA, June 2015.

[79] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving Deeper Into Convolutional Networks For Learning Video Representations," https://arxiv.org/abs/1511.06432.

[80] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1029–1038, Las Vegas, Nevada, USA, June 2016.

[81] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4584–4593, Las Vegas, Nevada, USA, June 2016.

[82] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: a large video description dataset for bridging video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5288–5296, Las Vegas, Nevada, USA, June 2016.

[83] C. Xiao, H. Zhong, Z. Guo et al., "Cail2018: a large-scale legal dataset for judgment prediction," https://arxiv.org/abs/1807.02478.

[84] L. Zhong, Z. Zhong, Z. Zhao, S. Wang, K. D. Ashley, and M. Grabmair, "Automatic summarization of legal decisions using iterative masking of predictive sentences," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 163–172, Montreal, QC, Canada, June 2019.

[85] G. Governatori, A. Rotolo, R. Riveret, and S. Villata, "Modelling dialogues for optimal legislation," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 229–233, Montreal, QC, Canada, June 2019.

[86] Y. Kang, D. Peng, Z. Chen, and C. Liu, "Ed-gan: Judicial document generating model based on improved generative adversarial networks," *Journal of Chinese Computer Systems*, vol. 40, no. 5, pp. 1020–1025, 2019.

[87] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pp. 225–230, ACM, CA, USA, June 2007.

[88] N. Martínez Melis and A. Hurtado Albir, "Assessment in translation studies: research needs, Meta," *journal des traducteurs/Meta: Translators' Journal*, vol. 46, no. 2, pp. 272–287, 2001.

[89] J. Keppens, "Explainable bayesian network query results via natural language generation systems," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 42–51, Montreal, QC, Canada, June 2019.

Hindawi
Scientific Programming
Volume 2021, Article ID 9921831, 17 pages
https://doi.org/10.1155/2021/9921831

Research Article

Saliency Detection in Weak Light Images via Optimal Feature Selection-Guided Seed Propagation

Nan Mu, Hongyu Wang, Yu Zhang, Hongyu Han, and Jun Yang

School of Computer Science, Sichuan Normal University, Chengdu 610101, China

Correspondence should be addressed to Nan Mu; nanmu@sicnu.edu.cn

Received 24 March 2021; Revised 28 July 2021; Accepted 23 August 2021; Published 13 September 2021

Academic Editor: Liang Zou

Salient object detection has a wide range of applications in computer vision tasks. Although tremendous progress has been made in recent decades, the weak light image still poses formidable challenges to current saliency models due to its low illumination and low signal-to-noise ratio properties. Traditional hand-crafted features inevitably encounter great difficulties in handling images with weak light backgrounds, while most of the high-level features are unfavorable to highlight visually salient objects in weak light images. In allusion to these problems, an optimal feature selection-guided saliency seed propagation model is proposed for salient object detection in weak light images. The main idea of this paper is to hierarchically refine the saliency map by learning the optimal saliency seeds in weak light images recursively. Particularly, multiscale superpixel segmentation and entropy-based optimal feature selection are first introduced to suppress the background interference. The initial saliency map is then obtained by the calculation of global contrast and spatial relationship. Moreover, local fitness and global fitness are used to optimize the prediction saliency map. Extensive experiments on six datasets show that our saliency model outperforms 20 state-of-the-art models in terms of popular evaluation criteria.

1. Introduction

Aiming to mimic *human visual system* (HVS), which has the ability to effortlessly sort out the most attractive things from the scene in front of eyes, the goal of salient object detection is to calculate the most important objects in an image. For the moment, salient object detection can substantially facilitate a series of applications, such as image segmentation [1, 2], object recognition [3], image retrieval [4], image compression [5], and photo cropping [6].

By computing pixel or region uniqueness in either low-level cue or high-level cue, existing salient object detection models can be broadly divided into two types. (1) Bottom-up models are usually unsupervised and based on local contrast or global contrast. These methods tend to suffer from false detections in the context of cluttered background and less effective visual features. (2) Top-down models mainly leverage supervised learning to guide object detection. However, the complexity of the algorithm and the diversity of objectives limit the generality of these methods.

Although a large number of bottom-up and top-down salient object detection models have been proposed, most of them are only designed for normal light scenes. These saliency models are confronted with significant challenges in weak light images due to low signal-to-noise ratio and lack of well-defined features to capture saliency information in low lighting scenarios. The most likely reasons may attribute to two aspects: (1) current hand-crafted visual features can hardly evaluate the objectness in weak light images; (2) most of the high-level features normally present enormous challenges in detecting accurate object boundary information, which can be easily blurred due to multiple levels of convolution layers and pooling layers in common convolutional neural network models.

To address these challenges, this paper proposes an optimal feature selection-based saliency seed propagation model for salient object detection in weak light images (the code of this paper can be downloaded from https://drive.google.com/open?id=1w0qBapNVygh8TOOp7AijFWOxsYxdRcWa). Several hand-crafted visual features are selected to hierarchically refine the saliency map obtained from the high-level cues recursively. The flowchart of our model is presented in Figure 1. The optimal low-level features are first selected to give a robust expression for weak light images,

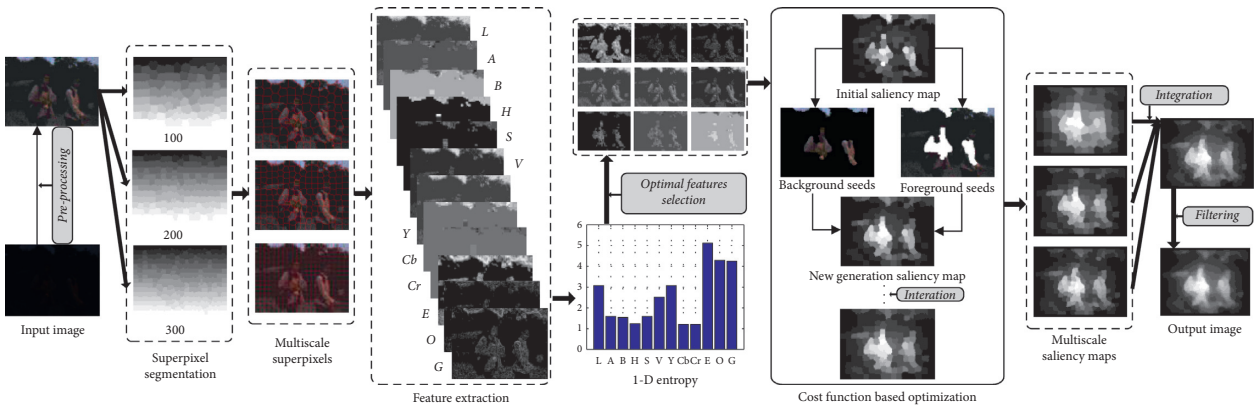


Figure 1: Overview of the proposed salient object detection model in the weak light image.

which aims to capture more objectness information and contributes to the prediction of salient objects under weak light conditions. Next, two cost functions are introduced to iteratively optimize the foreground seeds and background seeds of the initial saliency map, which can continuously compensate for salient information and remove nonsalient information to generate more precise object details. To estimate the overall performance, the proposed model is compared with 20 state-of-the-art salient object detection models on six datasets.

The paper is an extended version of our previously accepted conference paper [7], which provides a more detailed explanation and richer experimental demonstration. To sum up, this research has four main contributions: (1) a bottom-up visual saliency model, which requires no training, is explored toward weak light images, (2) an effective feature selection strategy is put forward to provide a robust representation of saliency information, (3) two cost functions are built to refine the initial saliency seeds recursively, and (4) a *nighttime image* (NI) dataset (the nighttime image (NI) dataset can be downloaded from https://drive.google.com/open?id=0BwVQK2zsuAQwX2hXbnc3ZVMzejQ) is constructed to verify the performance of our model.

The rest of this paper is organized as follows. Section 2 reviews the related works of saliency detection. Section 3 introduces the proposed saliency model. Section 4 presents the experimental results of the state-of-the-art models and the proposed model on six datasets. The conclusion of this paper is given in Section 5.

2. Related Works

Numerous salient object detection models have been proposed recently (see [8] for review); the main task of them is to highlight the most important visual regions for further processing. Depending on whether the task-independent or task-dependent is considered, they can be categorized as the bottom-up models and the top-down models, respectively.

Bottom-up saliency models are stimuli-driven and rely on low-level features. One typical model was presented by Itti et al. [9], which is mainly based on the center-surround difference of multiple features. Following this pioneering work, various bottom-up saliency models were proposed. Goferman et al. [10] computed the saliency value of image patches by implementing the local and global contrasts. Cheng et al. [11] executed the saliency computation by calculating the histogram and region contrasts. Xu et al. [12] introduced the contrast and spatial distribution strategies to evaluate the image saliency. Kim et al. [13] estimated the local saliency and global saliency based on regression and high-dimensional color transform. Hu et al. [14] performed salient object detection by utilizing the compactness hypothesis of color feature and texture feature. Huang and Zhang [15] presented a minimum directional contrast based salient object detection method. Wang et al. [16] exploited the pyramid attention and salient edges to guide the salient object detection. Sun et al. [17] detected the salient objects by employing a cascaded bottom-up feature aggregation module to capture the detailed information of low-level features. Jiang et al. [18] proposed a task-independent saliency model based on the bidirectional absorbing Markov chains. Molin et al. [19] exploited a neuromorphic dynamic bottom-up saliency detection method, which is feed-forward and requires no training. Typically, these bottom-up saliency models tend to face many difficult problems in handling images of a busy background and struggle to predict the true salient objects, which are in a low-contrast weak light environment.

Top-down saliency models are task-driven and rely on high-level perceptual learning. Xu et al. [20] used the *support vector machine* (SVM) model to produce the superpixel-level saliency map. Qu et al. [21] proposed a deep learning-based salient object detection model by combining the superpixel-based Laplacian propagation and the trained *convolutional neural network* (CNN) model. Mu et al. [22] designed a region covariance-based CNN method to learn the saliency value of image patches. Wang et al. [23] employed the top-down process for coarse-to-fine saliency estimation. Mu et al. [24] explored global convolutional and boundary refinement in a top-down manner to guide the learning of salient objects. Qiu et al. [25] introduced an *automatic top-down fusion* (ATDF) saliency model, which utilizes the global information to guide the learning of underlying knowledge. Zhang et al. [26] developed a top-down multilevel fusion method for RGB-D salient object detection. Wang et al. [27] progressively optimized the salient objects by exploiting the fixation map in a top-down mode. Xu et al. [28] utilized a *progressive architecture with a knowledge*

review network (PA-KRN) for salient object detection, which compensates for the important information in a top-down way. Dong et al. [29] presented a *bidirectional collaboration network* (BCNet) for salient object detection, which integrates feature fusion and feature aggregation in an edge-guided top-down progressive pathway. These top-down saliency models generally have high computational complexity and are relatively ineffective in determining accurate boundary and localization of salient objects under weak light conditions.

Since saliency detection in a weak light environment is a challenging problem, there were few studies on the salient object detection of weak light images [30, 31]. Mu et al. [30] proposed an *ant colony optimization* (ACO) based saliency model for predicting the salient objects on weak light images. Xu et al. [31] explored an image enhancement method for salient object detection in weak light images. These saliency models, however, are not robust enough to capture the salient objects in real-time. Different from these previous methods, the proposed model creates a totally unsupervised algorithm by integrating the bottom-up measures and the single-objective optimization cues. Specifically, (1) the proposed saliency model explores low-level features to represent the object properties and selects the most effective ones based on the entropy information; (2) the superpixel-level saliency is directly estimated by the feature dissimilarity and spatial similarity; (3) the prior saliency map, which contains the foreground seeds and the background seeds, is generated by implementing the bottom-up measures; (4) the single-objective optimization cues are formulated by designing the fitness-based cost functions to iteratively optimize the salient and nonsalient seeds; and (5) experimental results indicate that the proposed model can generate a high-performance saliency map in real time.

3. The Proposed Saliency Model

The proposed optimal feature selection-guided saliency seed propagation model is presented in detail in this section. The input image is first segmented into superpixels at three scales. Then, 12 features are extracted from the preprocessing image, and only nine optimal ones are chosen for the next calculation. Next, the initial saliency map is computed by combining the global contrast and center prior. And then, the new saliency map can be obtained by the foreground and background seeds from the previous one. Two cost functions which are based on global fitness and local fitness are defined to control the end of the iteration. At last, the optimal saliency map is obtained, and the results of three scales are integrated to get the final saliency map.

3.1. Multiscale Superpixel Segmentation. To make full use of the midlevel information and preserve the object structure context of the input image, the *simple linear iterative clustering* (SLIC) algorithm [32] is used to divide the input image into N superpixels (denoted as $\{s_i\}$, $i = 1, \ldots, N$). This operation can boost the efficiency of the method by regarding the superpixel as a processing unit. For saliency detection, the background region is more likely to have semblable superpixels at different scales, while the salient regions may have similar superpixels at some scales. That is, the fusion of the acquired salient superpixels at different scales can more accurately represent the real salient regions. However, as the number of superpixels increases, the time required for superpixel segmentation also increases. For accuracy and efficiency, our model generates the superpixels at three different scales, where the superpixel number N is set to 100, 200, and 300, respectively. The final saliency map is the integration of the obtained multiscale saliency maps.

3.2. Effective Feature Extraction. Given an input image, 12 low-level visual features are extracted, containing nine color features in three color spaces, the texture feature based on local entropy information, the orientation feature fused by the information in four directions, the gradient feature obtained from the horizontal and vertical vectors. Since the effectiveness of these various features varies according to the contrasts of different input images, nine optimal ones are selected from the 12 features, and the adaptive selection strategy is mainly based on the global information entropy of these features. The feature extraction process is introduced in detail as follows.

3.2.1. Color Features. The input image is first normalized to eliminate the interference of shadow and light (see preprocessing in Figure 1). This preprocessing is a general procedure in our model, including processing both normal light images and weak light images. Then, the input image is transformed from RGB color space to LAB, HSV, and YCbCr color spaces to capture nine color features. The L, A, and B components of LAB color space can describe all colors visible to the human eye, which are closer to human visual perception in weak light images. The H, S, and V components of HSV color space can be very intuitive to represent the hue, depth, and bright degree, which have good robustness in low lightness and weak light images. The Y, Cb, and Cr components of YCbCr color space can better perceive the intensity changes and the chromatic differences, which are more conducive to highlight the salient object information in weak light images.

3.2.2. Texture Feature. The 2-dimensional entropy of the original image is mainly used to represent the texture feature. Let I, $(0 \le I \le 255)$ denote the gray value of an image pixel, and let J, $(0 \le J \le 255)$ denote the average gray value of its neighborhood pixels; the spatial synthesis characteristic of gray distribution can be expressed as follows:

$$p_{IJ} = \frac{f(I, J)}{R^2}, \tag{1}$$

where $f(I, J)$ is the frequency of the characteristic tuple (I, J) and R^2 is the size of the neighborhood region. The discrete 2-dimensional entropy of the input image is defined as follows:

$$E = \sum_{I=0}^{255} p_{IJ} \log p_{IJ}. \tag{2}$$

Since the entropy information has strong resistance against noise interference and geometric deformation, the texture feature changes of salient objects in the weak light image can be well estimated by the variations in entropy.

3.2.3. Orientation Feature. The orientation feature is computed by executing the Gabor filter of different directions (denoted as $g_\theta(x, y)$) on the grayscale image (denoted as $\text{gray}(x, y)$) via

$$O = \sum_{\theta \in \{0°, 45°, 90°, 135°\}} gray(x, y) * g_\theta(x, y). \tag{3}$$

The rotational invariance and the global property of the orientation feature make it have less impact from weak light scenes.

3.2.4. Gradient Feature. The gradient feature is calculated by averaging the vertical gradient and horizontal gradient via

$$\begin{aligned} G = &|\text{gray}(x+1, y) - \text{gray}(x, y)| \\ &+ |\text{gray}(x, y+1) - \text{gray}(x, y)|. \end{aligned} \tag{4}$$

Thus, the magnitude information of local grayscale changes can be represented by the gradient feature, which can overcome the interference of a low signal-to-noise ratio in the weak light image.

3.2.5. Optimal Feature Selection. Feature selection plays an important role in predicting the real salient objects in weak light images. Gopalakrishnan et al. [33] proposed an unsupervised feature selection method, which removes the irrelevant features by maximizing the mixing rate of Markov processes of different features. However, naive inclusion of irrelevant features for a particular image can easily lead to performance degradation. Liang et al. [34] explored feature selection methods in supervised saliency learning, the features utilized in the model are highly redundant. Naqvi et al. [35] selected useful features by measuring the feature quality. However, they use a large number of features trying to explain all possible saliency-related factors, which increases the time cost and ignores some truly effective features. Since the goal of our model is to identify a small set of optimal features, with which the salient object detection in the weak light image can be both efficient and effective, traditional adaptive feature selection techniques are not suitable for us. The proposed model mainly extracts 12 features to participate in the salient object calculation. Due to the fact that the effectiveness of each feature is different when the image contrast changes, which can be seen in Figure 2, nine optimal features (denoted as $\{F_k\}$, $k = 1, \ldots, 9$) that can better describe the attributes of the corresponding weak light image are then selected from the extracted 12 different visual features $\{L, A, B, H, S, V, Y, Cb, Cr, E, O, G\}$ by calculating the 1-dimensional entropy information of these feature maps as follows:

$$\text{entropy} = \sum_{I=0}^{255} p_I \log p_I, \tag{5}$$

where p_I denotes the proportion of image pixels and I denotes the grayscale values of these pixels.

As a statistical feature form, the mean information content contained in the aggregation properties of image grayscale distribution can be well represented by image entropy information. The greater the entropy of the feature map $\{F_k\}$ is, the more efficient this feature will be. Thus, the selected nine optimal features could better account for the visual saliency of the corresponding weak light image.

3.3. Initial Saliency Map Generation. The global contrast measure and the spatial relationship strategy of the feature map are calculated to estimate the saliency value of each superpixel as follows:

$$\begin{aligned} \text{Sal}(s_i) &= \left(\sum_{j=1, j \neq i}^{N} \frac{\sqrt{(F_k(s_i) - F_k(s_i))^2}}{1 + \text{pos}(s_i, s_j)} \right) \times c(s_i), \\ c(s_i) &= \exp\left(-\frac{(x_i - x')^2}{2v_x^2} - \frac{(y_i - y')^2}{2v_y^2} \right), \end{aligned} \tag{6}$$

where $\text{pos}(s_i, s_j)$ is the Euclidean distance between superpixels s_i and s_j. $c(s_i)$ denotes the spatial distance between the coordinate (x_i, y_i) and image center (x', y'). v_x and v_y are variables, which are decided by the vertical and horizontal information of the input image.

3.4. Saliency Map Optimization. To achieve clean and uniform salient objects, optimization strategies are considered to improve detection accuracy. Zhu et al. [36] presented a principled optimization structure to fuse multiple low-level saliency cues, the whole framework mainly relies on the background cues, and it does not work well in weak light images, of which the background information is cluttered. Lu et al. [37] devoted to learning optimal saliency seeds set by utilizing a large margin formulation of discriminant saliency criterion. However, the gradient descent they used is not robust in weak light images and is not efficient for high accuracy salient object detection. In the proposed model, we built two cost functions to refine the generated saliency seeds recursively, which is an effective and straightforward manner to obtain more accurate salient objects in weak light images. The initial saliency map (denoted as Smap_k, $k = 0$) is first segmented into the salient region and nonsalient region by utilizing Otsu's thresholding [38]. The salient region and nonsalient region can be seen as the foreground seeds (denoted as FS) and the background seeds (denoted as BS) of the input image, respectively. The larger the difference between the superpixel and the foreground region is, the lower the saliency value of this superpixel is. Conversely, the greater the difference between the superpixel and the background region is, the higher the saliency value of this superpixel will be. Thus, the saliency value of s_i can be updated based on foreground seeds FS and background seeds BS as follows:

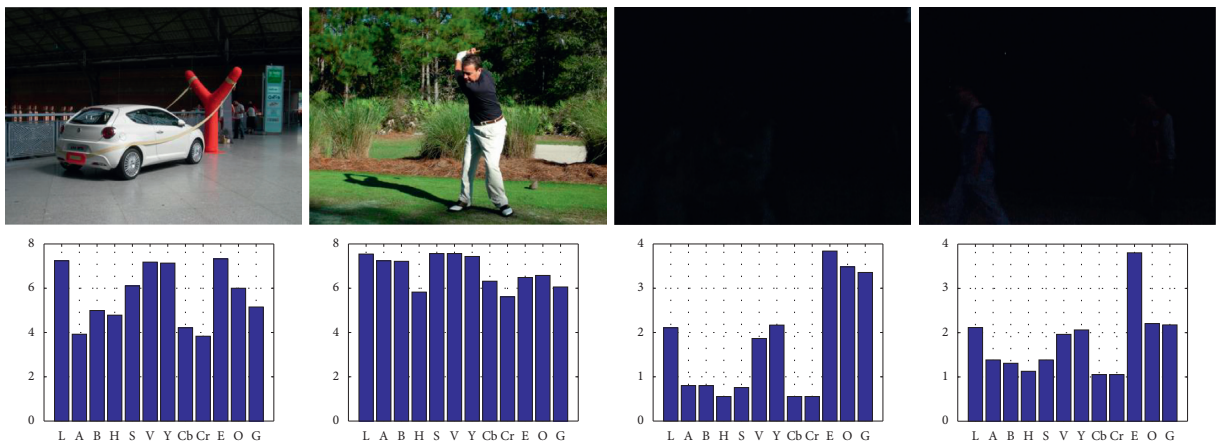


FIGURE 2: The top row contains the test images with different contrasts. The bottom row is the corresponding entropy bar graphs of 12 different features.

$$\mathrm{Sal}_{\mathrm{FS}}(s_i) = \sum_{s_j \in \mathrm{FS}, j \neq i} \frac{1}{\left(\sqrt{\left(F_k(s_i) - F_k(s_j)\right)^2} + \mathrm{pos}(s_i, s_j)\right)}, \tag{7}$$

$$\mathrm{Sal}_{\mathrm{BS}}(s_i) = \sum_{s_j \in \mathrm{BS}, j \neq i} \frac{\sqrt{\left(F_k(s_i) - F_k(s_j)\right)^2}}{\left(1 + \mathrm{pos}(s_i, s_j)\right)}, \tag{8}$$

$$\mathrm{Sal}(s_i) = \left(1 - \exp\left(-\frac{\mathrm{Sal}_{\mathrm{FS}}(s_i) + \mathrm{Sal}_{\mathrm{BS}}(s_i)}{2}\right)\right) \times c(s_i). \tag{9}$$

Then, a new saliency map (denoted as Smap_k, $k = 1$) of the first iteration optimization is obtained. The Otsu's method is reused to generate new FS and BS; the saliency map of the next generation (denoted as Smap_{k+1}) can be computed according to (7-9). Finally, two cost functions are implemented to decide whether the iteration procedures meet the end condition or not:

$$\text{minimize} \begin{cases} f_1(k) = \left(\mathrm{Smap}_k - \mathrm{Smap}_{k-1}\right)^2 \\ f_2(k) = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\left(\mathrm{Sal}(s_i) - \mathrm{Sal}(s_j)\right)^2}{1 + \mathrm{pos}(s_i, s_j)} \end{cases}, \quad \text{where } k \geq 1,\ s_i, s_j \in \mathrm{Smap}_k,\ 1 \leq i,\ j \leq N. \tag{10}$$

The function $f_1(k)$ mainly represents the global fitness, which denotes that the smaller the change between the saliency map of the new generation Smap_k and the previous generation Smap_{k-1} is, the more optimization of the objective can be. The function $f_2(k)$ mainly represents the local fitness, which denotes that the smaller the difference between the superpixel $\mathrm{Sal}(s_i)$ and its neighboring superpixels $\mathrm{Sal}(s_j)$ is, the better the saliency information of each decision variable can be. By minimizing the two functions $f_1(k)$ and $f_2(k)$, the optimal superpixel-level saliency map can be obtained.

4. Experiment Results

Comprehensive experiments are carried out on six datasets to estimate the performance of our model against 20 state-of-the-art salient object detection models.

4.1. Experimental Setup

4.1.1. Testing Datasets. The six test datasets contain five public datasets and the proposed weak light image dataset as follows: (1) the MSRA dataset [39] includes 10000 images which have relatively high contrast and only simple background; (2) the SOD dataset [40] includes various images of multiple objects and complex background; (3) the CSSD dataset [41] includes complex natural scenes; (4) the DUT-OMRON dataset [42] includes complex and challenging images; (5) the PASCAL-S dataset [43] includes images of cluttered background; and (6) our NI dataset includes 200 weak light images, which are captured at night with a stand camera. The resolution of these images is 640×480, and the human-annotated *ground-truths* (GTs) are also given.

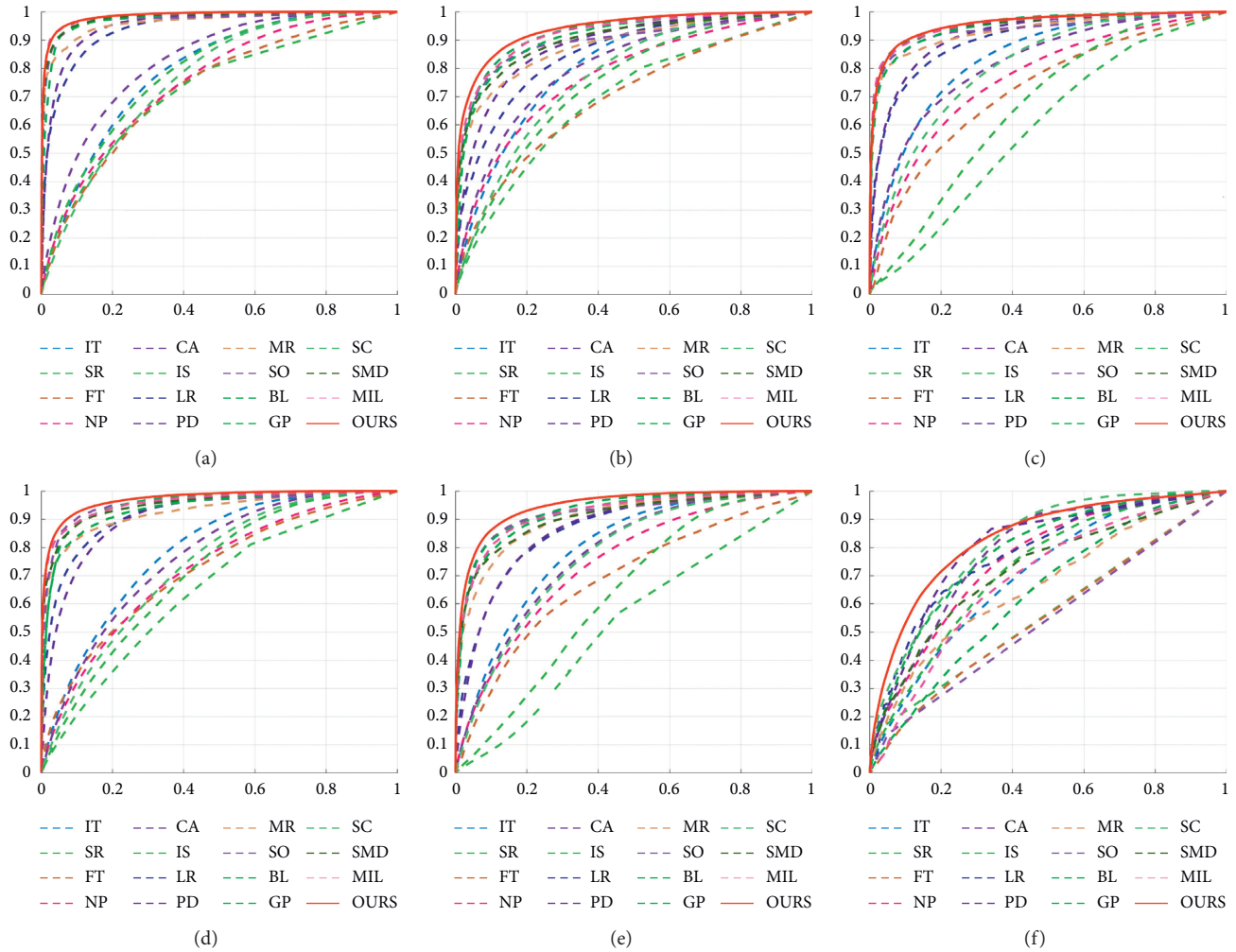


Figure 3: The TPRs-FPRs curves performance comparisons of different saliency models on the six datasets. (a) MSRA dataset. (b) SOD dataset. (c) CSSD dataset. (d) DUT-OMRON dataset. (e) PASCAL-S dataset. (f) NI dataset.

4.1.2. Comparison Models. The first 15 state-of-the-art saliency models include: *Itti*'s (IT) model [9], *spectral residual* (SR) model [44], *frequency-tuned* (FT) model [45], *nonparametric* (NP) model [46], *context-aware* (CA) model [10], *image signature* (IS) model [47], *low rank matrix recovery* (LR) model [48], *patch distinct* (PD) model [49], *graph-based manifold ranking* (MR) model [42], *saliency optimization* (SO) model [36], *bootstrap learning* (BL) model [50], *generic promotion* (GP) model [51], *spatiochromatic context* (SC) model [52], *structured matrix decomposition* (SMD) model [53], and *multiple-instance learning* (MIL) model [54]. All these experiments are performed by MATLAB software on an Intel i5-5250 CPU (1.6GHz) PC with 8 GB RAM.

4.1.3. Evaluation Criteria. To estimate the overall performance of various saliency models, seven criteria are used, including the *true positive rates* and *false positive rates* (TPRs-FPRs) curve, the *precision-recall* (PR) curve, the *area under the curve* (AUC) score, the *mean absolute error* (MAE) score, the *weighted F-measure* (WF) score, the *overlapping ratio* (OR) score, and the average execution time per image (in seconds).

The TPR is defined as the ratio of salient pixels that are correctly detected to all the true salient pixels, and FPR corresponds to the ratio of falsely detected salient pixels to all the true nonsalient pixels. The precision is computed as the ratio of correctly detected salient pixels to all the detected salient pixels, and the recall is the same as TPR, which measures the comprehensiveness of the detected salient pixels. By varying the threshold over the obtained saliency map, different TPRs, FPRs, precisions, and recalls can be calculated by comparing the generated different binary images with GT via

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}, \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned} \tag{11}$$

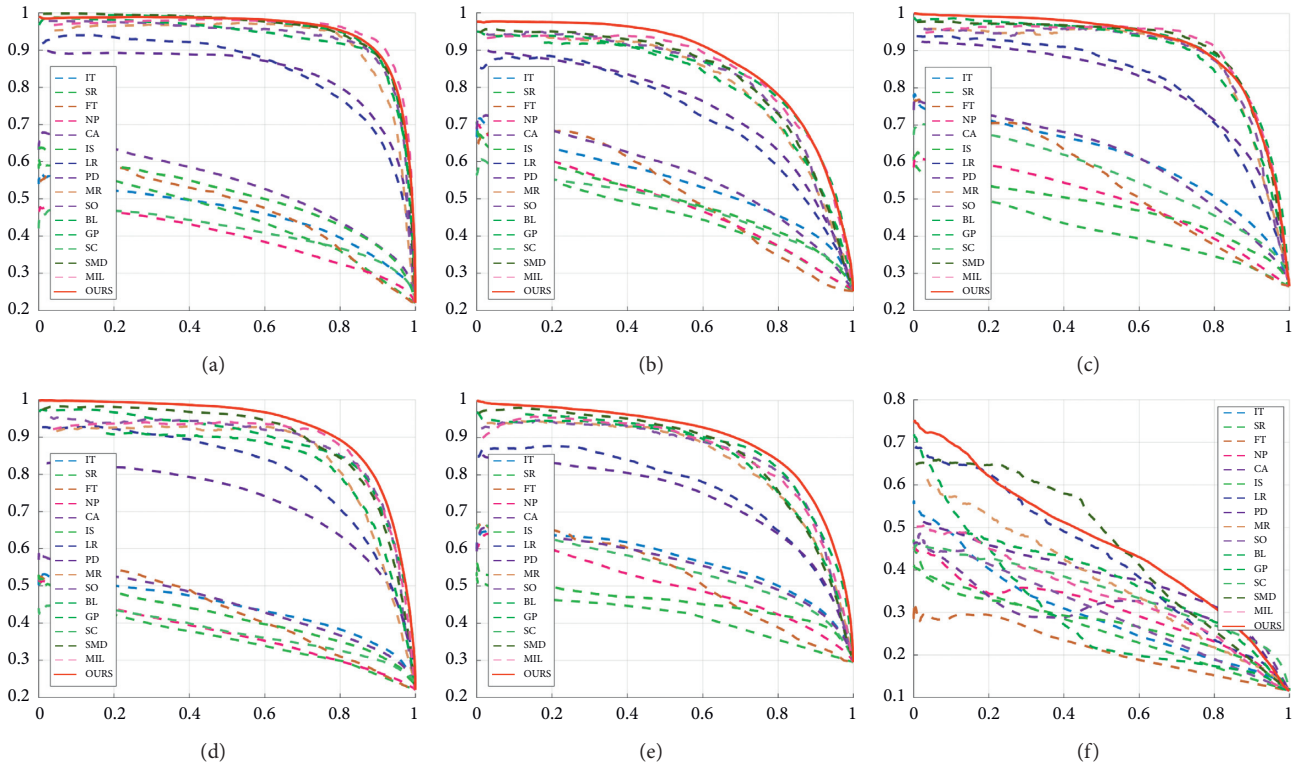


Figure 4: The PR curves performance comparisons of different saliency models on the six datasets. (a) MSRA dataset. (b) SOD dataset. (c) CSSD dataset. (d) DUT-OMRON dataset. (e) PASCAL-S dataset. (f) NI dataset.

where the *true positive* (TP) is the collection of pixels that correctly identify the salient object; the *false positive* (FP) is the collection of pixels that falsely identify the salient object; the *true negative* (TN) is the collection of pixels that correctly identify the nonsalient pixels; and the *false negative* (FN) is the collection of pixels which falsely identify the nonsalient pixels.

The TPRs-FPRs curve and the PR curve can be generated by plotting the corresponding ratios. The AUC score is calculated by measuring the proportion of the area under the TPRs-FPRs curve, which can give an intuitive indication of how well the obtained saliency map represents the real salient objects. The MAE score is calculated as the average absolute difference between the generated saliency map (denoted as Salmap) and the ground-truth (denoted as GT) via

$$\mathrm{MAE} = \mathrm{mean}\left(\left|\mathrm{Salmap}\left(x, y\right) - \mathrm{GT}\left(x, y\right)\right|\right). \tag{12}$$

The smaller the MAE value is, the higher the similarity between Salmap and GTis. The F-measure score is computed as the weighted harmonic mean of the precision and the recall via

$$F_\beta = \frac{\left(1 + \beta^2\right)\mathrm{precision} \cdot \mathrm{recall}}{\beta^2 \cdot \mathrm{precision} + \mathrm{recall}}, \tag{13}$$

where $\beta^2 = 0.3$ is the parameter to weigh the precision and recall. The WF score is calculated by adding a weighting function to the detection errors [55].

The OR score is measured by computing the overlapping ratio of salient pixels between the binary saliency map (denoted as Bmap) and GT via

$$\mathrm{OR} = \frac{\left|\mathrm{Bmap}\left(x, y\right) \cap \mathrm{GT}\left(x, y\right)\right|}{\left|\mathrm{Salmap}\left(x, y\right) \cup \mathrm{GT}\left(x, y\right)\right|}. \tag{14}$$

4.2. Experimental Results. The quantitative performances of our salient object detection method against the other 15 saliency models on the six datasets are presented in Figure 3 and 4 and Tables 1–6. The best three experimental results of Table 1–6 are highlighted in the red, blue, and green fonts, respectively. In particular, the up-arrow ↑ denotes the larger the value is, the better the performance of the saliency model is. At the same time, the down-arrow ↓ indicates the opposite meaning. As shown in the quantitative results, our salient object detection model performs the first or second performance on the five public datasets in most cases and obtains the best performance on the NI dataset in a relatively low time-consuming.

On MSRA, DUT-OMRON, and PASCAL-S datasets ((a), (d) and (e) of Figures 3 and 4 and Tables 1–3), our model achieves the best performance on the TPRs-FPRs curve, PR curve, and AUC score, while the saliency model SO obtains the best MAE score and WF score, and the saliency model MIL obtains the best OR score. The main reason is that the SO model used boundary connectivity and global optimization to increase its robustness, and the MIL model introduced a multiple-instance learning approaches to

Table 1: Quantitative performance of various salient object detection models using five criteria on the MSRA dataset.

Criteria	IT	SR	FT	NP	CA	IS	LR	PD	MR	SO	BL	GP	SC	SMD	MIL	OURS
AUC↑	0.7261	0.5556	0.6682	0.6942	0.7580	0.7375	0.8462	0.8557	0.8544	0.8639	**0.8729**	0.8714	0.6922	0.8712	**0.8751**	*0.8793*
MAE↓	0.3475	0.2204	0.2659	0.4211	0.2700	0.3042	0.2240	0.1869	0.1067	*0.0934*	0.1548	0.1099	0.2841	**0.0956**	0.0963	**0.0945**
WF↑	0.2681	0.1032	0.2886	0.2668	0.3316	0.2860	0.4327	0.4772	0.6935	*0.7314*	0.5765	0.6853	0.2451	**0.7250**	0.7125	**0.7274**
OR↑	0.1551	0.2490	0.3002	0.0703	0.3324	0.2569	0.5515	0.6043	0.7688	**0.8044**	0.7541	0.7861	0.2109	0.8021	*0.8182*	**0.8068**
TIME↓	1.902	1.797	0.172	1.480	78.61	0.231	29.95	4.462	0.684	0.316	12.95	1.428	31.57	2.130	101.5	7.437

The italic values and bold values indicate the best performance.

Table 2: Quantitative performance of various salient object detection models using five criteria on the DUT-OMRON dataset.

Criteria	IT	SR	FT	NP	CA	IS	LR	PD	MR	SO	BL	GP	SC	SMD	MIL	OURS
AUC↑	0.7263	0.5325	0.6611	0.6669	0.7007	0.6605	0.8300	0.8165	0.8203	0.8441	**0.8523**	0.8291	0.6700	0.8441	**0.8489**	*0.8657*
MAE↓	0.3486	0.2301	0.2675	0.4315	0.2898	0.3461	0.2322	0.2149	0.1343	*0.1141*	0.1772	0.1441	0.2882	**0.1209**	0.1255	**0.1194**
WF↑	0.2663	0.0869	0.2758	0.2572	0.2892	0.2546	0.4107	0.4128	0.6085	*0.6645*	0.5297	0.6040	0.2291	**0.6443**	0.6273	**0.6560**
OR↑	0.1434	0.1810	0.2825	0.0543	0.2566	0.1662	0.4953	0.4650	0.6669	**0.7048**	0.6580	0.6401	0.1905	0.6975	*0.7085*	**0.6992**
TIME↓	1.99	1.962	0.140	1.385	54.16	0.238	30.36	4.305	0.687	0.437	18.53	1.411	35.28	1.861	114.4	8.749

The italic values and bold values indicate the best performance.

Table 3: Quantitative performance of various salient object detection models using five criteria on the PASCAL-S dataset.

Criteria	IT	SR	FT	NP	CA	IS	LR	PD	MR	SO	BL	GP	SC	SMD	MIL	OURS
AUC↑	0.7220	0.4942	0.6363	0.6736	0.6985	0.6011	0.7715	0.7525	0.7672	0.7820	**0.8078**	**0.7949**	0.6880	0.7817	0.7888	*0.8177*
MAE↓	0.3529	0.3074	0.3096	0.4083	0.3187	0.3925	0.2754	0.2580	0.1963	*0.1787*	0.2096	0.1920	0.3162	**0.1810**	0.1890	**0.1801**
WF↑	0.3289	0.0453	0.2977	0.3358	0.3511	0.2750	0.4071	0.4064	0.5514	*0.6037*	0.5459	0.5827	0.2825	**0.5917**	0.5714	**0.5966**
OR↑	0.1451	0.1358	0.2859	0.0777	0.2672	0.1419	0.3975	0.4330	0.5596	0.5982	0.5953	0.5752	0.2423	**0.6009**	*0.6058*	**0.6021**
TIME↓	2.57	2.564	0.189	1.487	99.38	0.383	60.78	7.810	0.833	0.556	25.40	3.104	27.70	2.232	140.9	12.07

The italic values and bold values indicate the best performance.

Table 4: Quantitative performance of various salient object detection models using five criteria on the SOD dataset.

Criteria	IT	SR	FT	NP	CA	IS	LR	PD	MR	SO	BL	GP	SC	SMD	MIL	OURS
AUC↑	0.7377	0.5537	0.6436	0.7038	0.7337	0.7044	0.7820	0.7820	0.7648	0.7725	**0.8181**	**0.8063**	0.6969	0.7923	0.8006	*0.8292*
MAE↓	0.3316	0.2555	0.2674	0.4102	0.2867	0.3490	0.2569	0.2268	0.1732	*0.1630*	0.1981	0.1839	0.2912	**0.1664**	0.1695	**0.1655**
WF↑	0.2976	0.1026	0.2754	0.2939	0.3296	0.2775	0.3745	0.4056	0.5494	**0.5659**	0.5019	0.5426	0.2568	**0.5734**	0.5486	*0.5739*
OR↑	0.1878	0.2241	0.2958	0.0961	0.3135	0.1801	0.4093	0.4553	0.5438	0.5817	0.5516	0.5366	0.2316	**0.5909**	**0.5934**	*0.5938*
TIME↓	2.450	2.465	0.162	1.397	81.03	0.533	39.65	5.888	0.768	0.527	20.49	2.023	30.18	2.420	139.2	9.967

The italic values and bold values indicate the best performance.

Table 5: Quantitative performance of various salient object detection models using five criteria on the CSSD dataset.

Criteria	IT	SR	FT	NP	CA	IS	LR	PD	MR	SO	BL	GP	SC	SMD	MIL	OURS
AUC↑	0.7606	0.5087	0.6570	0.6967	0.7466	0.6259	0.8062	0.8088	0.8023	0.8044	**0.8384**	0.8212	0.7187	**0.8256**	0.8215	*0.8393*
MAE↓	0.3336	0.2905	0.2706	0.4183	0.2960	0.4170	0.2449	0.2335	0.1479	**0.1366**	0.1730	0.1519	0.2943	*0.1314*	0.1428	**0.1407**
WF↑	0.3239	0.1059	0.2931	0.3096	0.3459	0.2745	0.4260	0.4380	0.6283	**0.6628**	0.5651	0.6253	0.2821	*0.6749*	0.6366	**0.6568**
OR↑	0.2086	0.1775	0.2978	0.0690	0.3267	0.1138	0.4795	0.4944	0.6425	0.6654	0.6388	0.6327	0.2658	*0.6851*	**0.6712**	**0.6701**
TIME↓	1.97	1.878	0.159	1.370	65.28	0.363	21.95	4.292	0.668	0.431	21.78	1.277	42.80	2.293	109.4	7.052

The italic values and bold values indicate the best performance.

Table 6: Quantitative performance of various salient object detection models using five criteria on the NI dataset.

Criteria	IT	SR	FT	NP	CA	IS	LR	PD	MR	SO	BL	GP	SC	SMD	MIL	OURS
AUC↑	0.6721	0.5117	0.5072	0.7079	0.6143	0.6433	**0.7545**	0.6123	0.6726	0.5379	0.6253	**0.7534**	0.7095	0.6419	0.5840	*0.7931*
MAE↓	0.2407	**0.1162**	0.1252	0.2178	0.1329	0.1864	0.2193	**0.1291**	0.3609	0.1357	0.3993	0.2778	0.1552	0.1580	0.1746	*0.1049*
WF↑	0.1392	0.0269	0.0359	0.1983	0.1432	0.1565	0.2114	0.1722	0.1671	0.0741	0.1398	**0.2226**	0.1647	**0.2177**	0.1551	*0.2298*
OR↑	0.1617	0.0983	0.0968	0.2295	0.2370	0.1931	**0.2636**	0.2187	0.1431	0.0624	0.1037	0.2475	**0.2599**	0.2413	0.1837	*0.2957*
TIME↓	4.279	4.744	0.293	4.941	107.1	0.807	179.6	13.64	1.387	1.274	94.297	6.520	34.16	4.978	191.1	13.20

The italic values and bold values indicate the best performance.

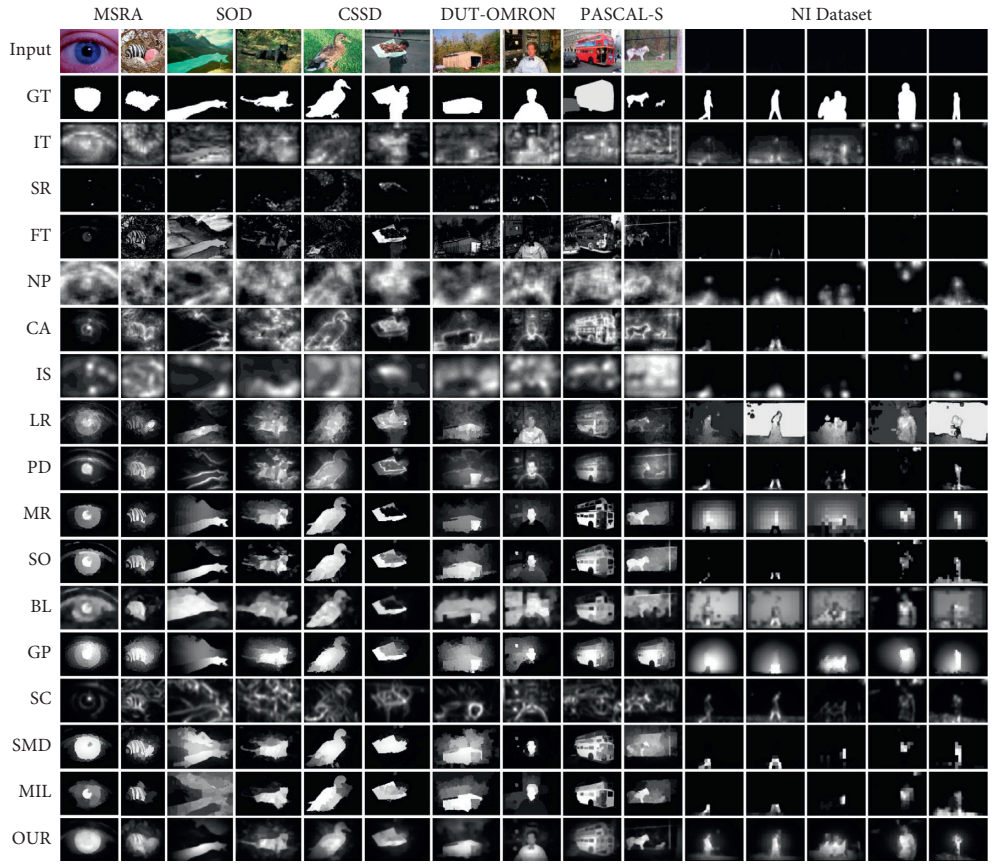


Figure 5: Visual comparisons of saliency maps construction using different approaches on the six datasets.

increase precision. These two saliency models take full advantage of the background measures, which can be of some effect in detecting the salient object under complex background conditions. Although the MAE score, WF score, and OR score of the proposed saliency model are slightly lower than the two models SO and MIL, our detection results are more competitive than the other models. The average time consumption of the MIL model is more than 100 seconds per image, which is not efficient in generating the saliency map.

On the SOD dataset (Figures 3(b) and 4(b), and Table 4), our saliency model has the best performance on the TPRs-FPRs curve, PR curve, AUC score, WF score, and OR score. In terms of the MAE criterion, the proposed model performs the second-best performance, which only has a small gap (0.0025) with the best MAE score of the SO model.

On the CSSD dataset (Figures 3(c) and 4(c), and Table 5), our saliency model performs the best performance on the TPRs-FPRs curve, PR curve, and AUC score. The MAE, WF, and OR scores of the proposed model are slighter than the best results achieved by the SMD model. The SMD model is based on the structured matrix decomposition with two regularizations, which has a strong potential in detecting the image of complex environments. The main reason for the poor performance of the proposed model on these metrics is that the selected optimal features contain less useful information that can effectively distinguish the salient objects.

On the NI dataset (Figures 3(f) and 4(f), and Table 6), our saliency model is superior, as it achieves the best performance on these criteria with a relatively short time-consuming.

The qualitative comparisons of saliency maps generated by the various salient object detection models on the six datasets are shown in Figure 5, indicating that our saliency model can detect the real salient object accurately in complex and/or weak light images (more detected saliency maps can be downloaded from https://drive.google.com/open?id=0BwVQK2zsuAQwQjZHeUJ1dlBsQms).

Since the standard real-world images and the weak light images have different properties, the proposed framework employs a feature selection strategy over the candidate feature set to pick out the most relevant features that apply to different types of images, which ensures that our model can be adapted to both standard saliency datasets and the weak light image dataset. In addition, we further optimize the saliency results through iteration to ensure robustness.

To further verify the effectiveness of our model, we have added some experiments with other five state-of-the-art deep learning-based saliency models (NLDF [56], LPS [57], BAS [58], F^3Net [59], and LDF [60]) to better illustrate the advantages of the proposed flowchart. The subjective performance comparisons of the proposed model with the latest deep saliency models are shown in Figure 6.

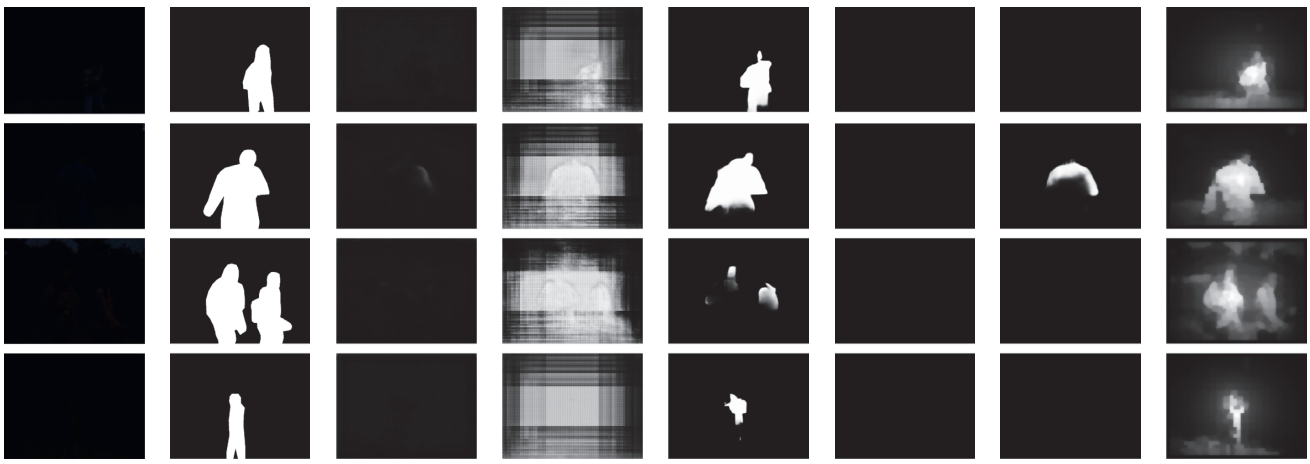


Figure 6: Visual comparisons of our saliency map with state-of-the-art deep learning-based saliency models on weak light image dataset. (a) Input. (b) GT. (c) NLDF. (d) LPS. (e) BAS. (f) F^3Net. (g) LDF. (h) OURS.

As can be seen in Figure 6, the saliency maps of the NLDF and F^3Net models cannot capture the effective salient objects in weak light images. The saliency results of the LPS model are seriously interfered with by the background noise. The saliency maps generated by the BAS model can highlight the salient objects with less noise, but the detected salient objects are incomplete. The LDF model has difficulty in detecting the whole objects and is prone to failure. Relatively speaking, the proposed model can accurately detect the real salient objects from the background on weak light images.

5. Conclusion

In this paper, we propose an optimal feature selection-based saliency seed propagation model to detect the salient object in weak light images. The main idea of the proposed saliency model is to execute saliency calculation by learning the optimal hand-crafted visual features and refining the foreground seeds and background seeds recursively. Guided by the optimized saliency seeds, the final saliency map can be achieved by fusing the multiple superpixel-level saliency maps at three different scales. Comprehensive experiments demonstrate that our saliency model performs satisfactory results against 20 state-of-the-art saliency models on five public datasets and a weak light image dataset.

Serving as a preprocessing step, salient object detection can efficiently focus on the most interesting area associated with the current visual task and it facilitates various computer vision applications such as image classification, object segmentation, visual tracking, etc. The proposed salient object detection model can be used to optimize correlational vision applications under weak light conditions, and it is of great application value to the monitoring system. In the future, we will further improve the time performance of the proposed saliency model and explore more potential applications.

Data Availability

The proposed nighttime image (NI) dataset, the code of the proposed model, and the experimental result 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 supported by the National Science Foundation of China (62006165 and 61701331).

References

[1] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 10, pp. 2074–2088, 2017.

[2] W. Wang, J. Shen, and F. Porikli, "Saliency-aware video object segmentation," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 3395–3402, Boston, MA, USA, June 2015.

[3] H. Chenini, "An embedded FPGA architecture for efficient visual saliency based object recognition implementation," in *Proceedings of the International Conference On Systems And Control*, pp. 187–192, Batna, Algeria, May 2017.

[4] X. Xiyu Yang, X. Xueming Qian, and Y. Yao Xue, "Scalable mobile image retrieval by exploring contextual saliency," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1709–1721, 2015.

[5] S. Li, M. Xu, Y. Ren, and Z. Wang, "Closed-form optimization on saliency-guided image compression for HEVC-MSP," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 155–170, 2018.

[6] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1531–1544, 2019.

[7] N. Mu, X. Xu, and X. Zhang, "Optimal feature selection for saliency seed propagation in low contrast images," in *Proceedings of the Pacific Rim Conference On Multimedia*, pp. 35–45, Hefei, China, September 2018.

[8] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: an in-depth survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2021.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915–1926, 2012.

[11] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[12] X. Xu, N. Mu, L. Chen, and X. Zhang, "Hierarchical salient object detection model using contrast-based saliency and color spatial distribution," *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2667–2679, 2016.

[13] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform and local spatial support," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 9–23, 2016.

[14] P. Hu, W. Wang, C. Zhang, and K. Lu, "Detecting salient objects via color and texture compactness hypotheses," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4653–4664, 2016.

[15] X. Huang, Y.-J. Zhang, and ", "FPS salient object detection via minimum directional contrast," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4243–4254, 2017.

[16] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proceedings of the IEEE International Conference On Computer Vision And Pattern Recognition*, pp. 1448–1457, Long Beach, CA, USA, June 2019.

[17] F. Sun, L. Huang, X. Yuan, and C. Zhao, "Salient object detection via attention-aware cascaded bottom-up feature aggregation," in *Proceedings of the IEEE International Conference On Multimedia And Expo*, Shenzhen, China, July 2021.

[18] F. Jiang, B. Kong, J. Li, K. Dashtipour, and M. Gogate, "Robust visual saliency optimization based on bidirectional Markov chains," *Cognitive Computation*, vol. 13, no. 1, pp. 69–80, 2021.

[19] J. L. Molin, C. S. Thakur, E. Niebur, and R. Etienne-Cummings, "A neuromorphic proto-object based dynamic visual saliency model with a hybrid FPGA implementation," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 15, 2021.

[20] X. Xu, N. Mu, H. Zhang, and X. Fu, "Salient object detection from distinctive features in low contrast images," in *Proceedings of the IEEE International Conference On Image Processing*, pp. 3126–3130, Quebec City, Canada, September 2015.

[21] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, "RGBD salient object detection via deep fusion," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2274–2285, 2017.

[22] N. Mu, X. Xu, X. Zhang, and H. Zhang, "Salient object detection using a covariance-based CNN model in low-contrast images," *Neural Computing & Applications*, vol. 29, no. 8, pp. 181–192, 2018.

[23] W. Wang, J. Shen, M.-M. Cheng, and L. Shao, "An iterative and cooperative top-down and bottom-up inference network for salient object detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 5968–5977, Long Beach, CA, USA, June 2019.

[24] N. Mu, X. Xu, and X. Zhang, "Salient object detection in low contrast images via global convolution and boundary refinement," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition Workshops*, pp. 1–9, Long Beach, CA, USA, June 2019.

[25] Y. Qiu, Y. Liu, H. Yang, and J. Xu, "A simple saliency detection approach via automatic top-down feature fusion," *Neurocomputing*, vol. 388, pp. 124–134, 2020.

[26] M. Zhang, Y. Zhang, Y. Piao, B. Hu, and H. Lu, "Feature reintegration over differential treatment: a top-down and adaptive fusion network for RGB-D salient object detection," in *Proceedings of the ACM International Conference On Multimedia*, pp. 4107–4115, Seattle, WA USA, October 2020.

[27] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang, "Inferring salient objects from human fixations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 1913–1927, 2020.

[28] B. Xu, H. Liang, R. Liang, and P. Chen, "Locate globally, segment locally: a progressive architecture with knowledge review network for salient object detection," in *Proceedings.of the AAAI Conference On Artificial Intelligence*, pp. 1–9, Vancouver, Canada, February 2021.

[29] B. Dong, Y. Zhou, C. Hu, K. Fu, and G. Chen, "BCNet: bidirectional collaboration network for edge-guided salient object detection," *Neurocomputing*, vol. 437, pp. 58–71, 2021.

[30] N. Mu, X. Xu, and X. Zhang, "Ant colony optimization based salient object detection for weak light images," in *Proceedings of the IEEE International Conference On Ubiquitous Intelligence And Computing*, pp. 1432–1437, Guangzhou, China, October 2018.

[31] X. Xu, S. Wang, Z. Wang, X. Zhang, and R. Hu, "Exploring image enhancement for salient object detection in low light images," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 8, pp. 1–19, 2021.

[32] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[33] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Unsupervised feature selection for salient object detection," in *Proceedings of the Asian Conference On Computer Vision*, pp. 15–26, Queenstown, New Zealand, November 2010.

[34] M. Ming Liang and X. Xiaolin Hu, "Feature selection in supervised saliency prediction," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 914–926, 2015.

[35] S. S. Naqvi, W. N. Browne, and C. Hollitt, "Feature quality-based dynamic feature selection for improving salient object detection," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 25, no. 9, pp. 4298–4313, 2016.

[36] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 2814–2821, Columbus, OH, USA, June 2014.

[37] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning optimal seeds for diffusion-based salient object detection," in

Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, pp. 2790–2797, Columbus, OH, USA, June 2014.

[38] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[39] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[40] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition Workshops*, pp. 49–56, San Francisco, CA, USA, June 2010.

[41] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1155–1162, Portland, OR, USA, June 2013.

[42] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 3166–3137, Portland, OR, USA, June 2013.

[43] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille, "The secrets of salient object segmentation," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 280–287, Columbus, OH, USA, June 2014.

[44] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[45] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1597–1604, Miami, FL, USA, June 2009.

[46] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency estimation using a non-parametric low-level vision model," in *Proceedings of the IEEE Computer Conference On Computer Vision And Pattern Recognition*, pp. 433–440, Colorado Springs, CO, USA, June 2011.

[47] X. Xiaodi Hou, J. Harel, and C. Koch, "Image signature: highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, 2012.

[48] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 853–860, Providence, RI, USA, June 2012.

[49] R. Margolin, L. Zelnik-Manor, and A. Tal, "What makes a patch distinct?" in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1139–1146, Portland, OR, USA, June 2013.

[50] N. Tong, H. Lu, and M. Yang, "Salient object detection via bootstrap learning," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1884–1892, Boston, MA, USA, June 2015.

[51] P. Jiang, N. Vasconcelos, and J. Peng, "Generic promotion of diffusion-based salient object detection," in *Proceedings of the IEEE International Conference On Computer Vision*, pp. 217–225, Santiago, Chile, December 2015.

[52] J. Zhang, M. Wang, S. Zhang, X. Li, and X. Wu, "Spatiochromatic context modeling for color saliency analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1177–1189, 2016.

[53] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank, "Salient object detection via structured matrix decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 818–832, 2017.

[54] F. Huang, J. Qi, H. Lu, L. Zhang, and X. Ruan, "Salient object detection via multiple instance learning," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1911–1922, 2017.

[55] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 248–255, Columbus, OH, USA, June 2014.

[56] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 6609–6617, Honolulu, HI, USA, July 2017.

[57] Y. Zeng, H. Lu, L. Zhang, M. Feng, and A. Borji, "Learning to promote saliency detectors," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1644–1653, Salt Lake City, UT, USA, June 2018.

[58] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: boundary-aware salient object detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 7479–7489, Long Beach, CA, USA, June 2019.

[59] J. Wei, S. Wang, and Q. Huang, "F^3Net: fusion, feedback and focus for salient object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12321–12328, 2020.

[60] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 13025–13034, Seattle, WA, USA, June 2020.

Hindawi
Scientific Programming
Volume 2021, Article ID 2292703, 11 pages
https://doi.org/10.1155/2021/2292703

Research Article

Similarity Network Fusion Based on Random Walk and Relative Entropy for Cancer Subtype Prediction of Multigenomic Data

Jian Liu,[1,2] **Wenfeng Liu**,[3] **Yuhu Cheng**,[1,2] **Shuguang Ge**,[1,2] **and Xuesong Wang**[1,2]

[1]*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China*
[2]*Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China*
[3]*Department of Information Center, Weihai Ocean Vocational College, Rongcheng 264300, China*

Correspondence should be addressed to Xuesong Wang; wangxuesongcumt@163.com

Received 6 May 2021; Revised 25 July 2021; Accepted 6 August 2021; Published 19 August 2021

Academic Editor: Liang Zhao

It is a crucial task to design an integrated method to discover cancer subtypes and understand the heterogeneity of cancer based on multiple genomic data. In recent years, some clustering algorithms have been proposed and applied to cancer subtype prediction. Among them, similarity network fusion (SNF) can integrate multiple types of genomic data to identify cancer subtypes, which improves the understanding of tumorigenesis. SNF uses a dense similarity matrix to obtain the global information of the data, and the interconnection of samples between different categories will cause noise interference. Therefore, how to construct a more robust dense similarity matrix is an important research content to improve the performance of cancer subtype identification. In this paper, we proposed similarity network fusion based on random walk and relative entropy (R^2SNF) for cancer subtype prediction. Firstly, the random walk algorithm was used to capture the complex relationship between samples in each genomic data. And the transition probability distribution of samples in the network was obtained. If two samples belong to the same class, the transition probability between the two samples is great. On the contrary, if the two samples do not belong to the same class, the transition probability between the two samples is small. In this way, the degree of correlation between samples can be well obtained, thereby reducing the noise interference caused by the interconnection of samples between different categories. Secondly, relative entropy was used to calculate the difference in the transition probability distribution between samples to construct a better dense similarity matrix which contains structural similarity information between samples. Thirdly, we iteratively fused the obtained dense similarity matrix with the KNN similarity matrix to construct the fused similarity matrix of all genomic data. Finally, by using spectral clustering, the fused similarity matrix was grouped into multiple clusters, which indicates the cancer subtypes. Experiments on seven cancer omics datasets show that the R^2SNF algorithm performs well in identifying cancer subtypes.

1. Introduction

With the rapid development of high-throughput technology, a large amount of genomic data has been generated, including gene expression data, DNA methylation data, and DNA copy number variation data. In particular, The Cancer Genome Atlas (TCGA) [1] database researches different genome, transcriptome, and epigenome information of more than 1,100 patients from more than 34 cancer types. These data have brought unprecedented opportunities to cancer research, such as driven gene selection [2] and cancer subtype prediction, so that cancer can be controlled more thoroughly and comprehensively.

Various types of genomic data are closely related to the occurrence and development of cancer. In general, cell growth and differentiation are regulated by the gene expression level, and the changes in the gene expression level will lead to transformation from normal cells to cancer cells [3]. DNA single-nucleotide polymorphisms and copy number variations in the genome affect gene instability and cancer gene activation through gene amplification or cancer suppression [4]. DNA methylation in epigenetic variation is

also common in cancer genomes. Genome-wide hypomethylation can lead to genome instability. The hypomethylation of CpG islands is also related to the inactivation of cancer suppressor genes [5]. At present, many studies have attempted to use these genomic data to predict cancer subtypes. However, the cancer genome is regulated by a variety of molecular mechanisms, the complexity and independence of which make it difficult to discover the relationship between the cancer genome and the cancer phenotype. Therefore, integrating different genomic data to capture the complexity of phenotypes and the heterogeneity of biological processes [6, 7] is the current trend in predicting cancer subtypes.

In the past few decades, many genomic data integration algorithms have been extensively developed. For example, Shen et al. [8] proposed a joint latent variable model named as iCluster, which combines the correlation between different types of genomic data and the variance-covariance structure within the data type to mine potential cancer subtypes. Akavia et al. [9] proposed an algorithm based on the Bayesian network to integrate the matching chromosome copy number and gene expression data of tumor samples to identify driving mutations and their influence processes. Liang et al. [10] proposed a multimodal deep belief network algorithm, which encodes the relationship between features of each genomic data as a multilayer network of hidden variables and then fuses common features to cluster cancer into different subtypes. Speicher and Pfeifer [11] added regularization constraints in the optimization process of multikernel learning to avoid overfitting and used one kernel for each genome data type to solve the problem of kernel function and parameter selection. Wang et al. [12] proposed a multiplexed network, which integrates heterogeneous genomic data by using the links between each node in a network slice and its corresponding nodes in each other network slice. Van et al. [13] used sequencing matrix decomposition to represent genomic data and identify cancer subtypes based on mutations and gene expression characteristics. Zhang and Ma [14] proposed a regularized multiview subspace clustering method to integrate gene expression data with the protein interaction network of dynamic modules. Network-based stratification (NBS) [15, 16] method combines genome-scale somatic mutation profiles with a gene interaction network to produce a robust subdivision of patients into subtypes. And the gene interaction network is constructed by protein-protein interactions (PPI). Simultaneous rank matrix factorization (SRF) [13] method approaches the subtyping problem by decomposing patient-mutation and patient-expression data into ranked factors.

Among these integrated algorithms, Wang et al. [6] proposed a very effective cancer subtype identification algorithm—similarity network fusion (SNF). SNF consists of three stages: network construction, network fusion, and clustering. In the network construction stage, the Euclidean distance of each omics data is used to construct a patient similarity network. In the network fusion stage, the information dissemination theory is used to perform nonlinear iterative fusion of the constructed network. Finally, the spectral clustering algorithm is used for clustering. SNF integrates mRNA expression data, DNA methylation data, and miRNA expression data and establishes a cancer subtype prediction model on five cancer datasets.

At present, many studies have improved and expanded SNF. Xu et al. [17] proposed a weighted similarity network fusion algorithm, which uses a complex miRNA-TF-mRNA regulatory network to identify cancer subtypes. In order to solve the problem that SNF is only applicable to data types containing continuous values, Yang et al. [18] used the random walk method to smooth the discrete somatic mutation data and incorporated the smoothed data into the SNF algorithm so that SNF can fuse discrete data. Yang et al. [19] proposed a deep subspace fusion clustering algorithm, which used the methods of self-encoding and data self-expression to guide the deep subspace model, which can effectively express the discriminant similarity between samples, thereby realizing the difference transfer between clustering clusters and the enhancement of compactness within clustering clusters. In view of the superior performance of SNF, it has become one of the most popular algorithms for cancer subtype identification. Therefore, this paper improves SNF from the perspective of similarity matrix construction, aiming to further improve the recognition effect of SNF on cancer subtypes.

After SNF completes network fusion, it needs to be clustered through spectral clustering [20]. The essence of spectral clustering is to map the Laplacian matrix so that the samples in the original space that are not easy to handle can be easily processed in the mapped space. The Laplacian matrix is calculated by the similarity matrix, so the construction of the similarity matrix is the key to SNF. SNF constructs two similarity matrices for each genomic data, dense similarity matrix and sparse similarity matrix, which are used to capture global and local information of genomic data, respectively. In SNF, K-nearest neighbor (KNN) algorithm is used to construct the sparse similarity matrix. KNN algorithm is the most commonly used and effective sparse similarity matrix construction method. All samples in the dense similarity matrix have connecting edges. In spectral clustering, the interconnection of samples of different categories in the dense similarity matrix will cause noise interference and affect the segmentation effect of spectral clustering. Therefore, how to optimize the dense similarity matrix has become a major problem faced by SNF.

In this paper, we proposed similarity network fusion based on random walk [17] and relative entropy (R^2SNF) for cancer subtype prediction. Random walk and relative entropy are used to measure the similarity between samples to construct a more robust dense similarity matrix on each genomic data. The similarity matrix construction method based on random walk measures the transition probability of a sample walking along a randomly selected adjacent edge to reach other samples, thereby forming a transition probability distribution of this sample. In order to better measure the similarity between samples, the relative entropy is used to calculate the difference of the transition probability distribution of them, and the similarity between them is obtained: the greater the difference between two probability

distributions is, the less similar the corresponding samples are, and vice versa. The dense similarity matrix construction method is to establish a random walk point on the basis of the conventional dense similarity matrix. It uses the difference in the transition probability distribution between samples to measure the similarity of two samples so that similar samples have a larger similarity value, and samples that are not of the same class have a smaller similarity value. Thus, a more robust dense similarity matrix is obtained. In our R^{2}SNF, we use the dense similarity matrix obtained above and the sparse similarity matrix obtained by the KNN algorithm to perform similarity network fusion for different genomic data. Finally, we use spectral clustering to cluster the fusion similarity matrix. Experimental results on multiple genomic data show that R^{2}SNF can identify biologically significant cancer subtypes.

2. Methods

In this section, we will introduce our algorithm similarity network fusion based on random walk and relative entropy (R^{2}SNF) in detail. Firstly, the probability distribution of random walk from one sample in each genomic data to other samples in the network is calculated. Secondly, relative entropy is used to calculate the difference of the probability distribution of the two samples, and the robust dense similarity matrix is constructed. Thirdly, similarity network fusion between the constructed robust dense similarity matrix and the KNN similarity matrix is performed to obtain the fused similarity matrix. Finally, spectral clustering is used for clustering the fused similarity matrix.

2.1. Construction of the Random Walk Model. Random walk [21] is a random process model that can simulate the interaction between samples in the network. The random walk on the graph can be regarded as a Markov chain of randomly selected nodes. After years of development, a variety of random walk algorithms have been produced. Here, we use the random walk with restart (RWR) algorithm proposed by Tong et al. [22].

Given a set of cancer genomic data $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^v, \ldots, \mathbf{X}^V\}$, $\mathbf{X}^v \in \mathbb{R}^{n \times m^v}$, where V represents the number of genomic data, $\mathbf{X}^v$ is the vth genomic data in $\mathbf{X}$, m^v represents that the vth genomic data have m features, and n represents the number of samples. For each genomic data $\mathbf{X}^v$, starting from the ith sample x_i^v, each step of the RWR faces two choices: choose the adjacent sample with the probability of α or return to the starting sample with the probability of $1 - \alpha$; then, the sample x_i^v will transfer to any sample and reach a stable state at the time $t + 1$. According to the Markov decision process, the current state of the system is only related to the state at the previous moment. Therefore, the stable state vector $\mathbf{r}_{t+1}^v(\mathbf{x}_i^v)$ at the time $t + 1$ can be defined as

$$\mathbf{r}_{t+1}^v(\mathbf{x}_i^v) = \alpha \mathbf{r}_t^v(\mathbf{x}_i^v)\mathbf{A}^v + (1-\alpha)\mathbf{r}_0^v(\mathbf{x}_i^v), \tag{1}$$

where $\mathbf{r}_t^v(\mathbf{x}_i^v)$ represents the state vector at the time t, $\mathbf{r}_0^v(\mathbf{x}_i^v)$ is the initialization vector with the ith element being 1 and the remaining elements being 0, and $\mathbf{A}^v \in \mathbb{R}^{n \times n}$ represents the transition probability matrix of each genomic data. Under normal circumstances, the probability transition matrix of the random walk on the graph can be represented by the adjacency matrix after data normalization. We adopt the following ideas to construct the transition probability matrix $\mathbf{A}^v$.

Firstly, we construct the similarity matrix $\mathbf{W}^v \in \mathbb{R}^{n \times n}$ for each genomic data by

$$\mathbf{W}^v(i,j) = \exp\left(-\frac{\rho^2\left(\mathbf{x}_i^v, \mathbf{x}_j^v\right)}{\mu \varepsilon_{i,j}}\right), \tag{2}$$

where $\mathbf{W}^v(i,j)$ represents the similarity between sample $\mathbf{x}_i^v$ and sample $\mathbf{x}_j^v$, μ is an empirical hyperparameter, and $\rho^2(\mathbf{x}_i^v, \mathbf{x}_j^v)$ is the Euclidean distance between samples $\mathbf{x}_i^v$ and $\mathbf{x}_j^v$. $\varepsilon_{i,j}$ can be defined as

$$\varepsilon_{i,j} = \frac{1}{3}\left(\text{mean}\left(\rho\left(\mathbf{x}_i^v, \mathbf{N}_i^v\right)\right) + \text{mean}\left(\rho\left(\mathbf{x}_j^v, \mathbf{N}_j^v\right)\right) + \rho\left(\mathbf{x}_i^v, \mathbf{x}_j^v\right)\right), \tag{3}$$

where $\text{mean}(\rho(\mathbf{x}_i^v, \mathbf{N}_i^v))$ denotes the average of the distances between the sample $\mathbf{x}_i^v$ and its neighbors.

In the process of random walk, $\mathbf{A}^v$ is a probability transition matrix, which needs to meet the condition $\sum_j \mathbf{A}^v(i,j) = 1$. We can get $\mathbf{A}^v$ by normalizing $\mathbf{W}^v$:

$$\mathbf{A}^v = \left(\mathbf{D}^v\right)^{-1}\mathbf{W}^v, \tag{4}$$

where $\mathbf{D}^v$ is the degree matrix, and its diagonal elements satisfy $\mathbf{D}^v(i,j) = \sum_j \mathbf{W}^v(i,j)$.

2.2. Construction of the Similarity Matrix Based on Relative Entropy. After calculating the stable state transition probability distribution $\mathbf{r}^v$ from the RWR in Section 2.1, the similarity $\mathbf{S}^v(\mathbf{x}_i^v, \mathbf{x}_j^v)$ between the sample $\mathbf{x}_i^v$ and sample $\mathbf{x}_j^v$ is usually defined as [23]

$$\mathbf{S}^v\left(\mathbf{x}_i^v, \mathbf{x}_j^v\right) = \mathbf{r}^v_{\mathbf{x}_i^v, \mathbf{x}_j^v} + \mathbf{r}^v_{\mathbf{x}_j^v, \mathbf{x}_i^v}, \tag{5}$$

where $\mathbf{r}^v_{\mathbf{x}_i^v, \mathbf{x}_j^v}$ is the probability of starting from $\mathbf{x}_i^v$ and arriving at $\mathbf{x}_j^v$ via random walk. However, this method only considers the probability value of the random walk between the two samples and ignores the structural similarity between them.

In order to better measure the similarity between samples, the difference in the transition probability distribution of two nodes is used to define the structural similarity. We use the relative entropy to construct the dense similarity matrix [24]. Relative entropy, also known as Kullback–Leibler (KL) divergence [25], is a method to describe the difference between two probability distributions. Here, relative entropy is used to calculate the difference of the transfer probability distribution of different samples.

For sample $\mathbf{x}_i^v$, the transition probability distribution $\mathbf{r}^v(\mathbf{x}_i^v)$ of reaching any other sample to reach a stable state after random walk can be written as

$$\mathbf{r}^v\left(\mathbf{x}_i^v\right) = \left[\mathbf{r}^v\left(\mathbf{x}_i^v, \mathbf{x}_1^v\right), \mathbf{r}^v\left(\mathbf{x}_i^v, \mathbf{x}_2^v\right), \ldots, \mathbf{r}^v\left(\mathbf{x}_i^v, \mathbf{x}_n^v\right)\right], \tag{6}$$

where n is the number of samples and $\mathbf{r}^v(\mathbf{x}_i^v, \mathbf{x}_j^v)$ is the new probability of starting from $\mathbf{x}_i^v$ and arriving at $\mathbf{x}_j^v$ via random walk. $\mathbf{r}^v(\mathbf{x}_i^v, \mathbf{x}_j^v)$ can be defined as

$$\mathbf{r}^v\left(\mathbf{x}_i^v, \mathbf{x}_j^v\right) = \frac{\mathbf{r}^v_{\mathbf{x}_i^v, \mathbf{x}_j^v}}{\sum_{k=1}^n \mathbf{r}^v_{\mathbf{x}_i^v, \mathbf{x}_k^v}}. \tag{7}$$

For the transition probability distribution $\mathbf{r}^v(\mathbf{x}_i^v)$ and $\mathbf{r}^v(\mathbf{x}_j^v)$ of any two samples $\mathbf{x}_i^v$ and $\mathbf{x}_j^v$, respectively, the relative entropy can be defined as

$$D_{\mathrm{KL}}\left(\mathbf{r}^v(\mathbf{x}_i^v)\middle\|\mathbf{r}^v\left(\mathbf{x}_j^v\right)\right) = \sum_k^n \mathbf{r}^v(\mathbf{x}_i^v, \mathbf{x}_k^v)\log_2\frac{\mathbf{r}^v(\mathbf{x}_i^v, \mathbf{x}_k^v)}{\mathbf{r}^v\left(\mathbf{x}_j^v, \mathbf{x}_k^v\right)}. \tag{8}$$

When $a = 0$ or $b = 0$, we define $\log_2(a/b) = 0$.

Relative entropy is an asymmetric measure; that is, $D_{\mathrm{KL}}(\mathbf{r}^v(\mathbf{x}_i^v)\|\mathbf{r}^v(\mathbf{x}_j^v)) \neq D_{\mathrm{KL}}(\mathbf{r}^v(\mathbf{x}_j^v)\|\mathbf{r}^v(\mathbf{x}_i^v))$. Therefore, the probability distribution difference matrix is defined as $\mathbf{C}^v$; then, the difference between any two probability distributions is $\mathbf{C}^v(i, j)$:

$$\mathbf{C}^v(i, j) = \frac{1}{2}\left(D_{\mathrm{KL}}\left(\mathbf{r}^v(\mathbf{x}_i^v)\middle\|\mathbf{r}^v\left(\mathbf{x}_j^v\right)\right) + D_{\mathrm{KL}}\left(\mathbf{r}^v\left(\mathbf{x}_j^v\right)\middle\|\mathbf{r}^v(\mathbf{x}_i^v)\right)\right). \tag{9}$$

Finally, $\mathbf{C}^v$ is transformed into a similarity matrix $\mathbf{S}^v$, where the elements are defined as $\mathbf{S}^v(i, j)$:

$$\mathbf{S}^v(i, j) = 1 - \frac{\mathbf{C}^v(i, j)}{\mathbf{C}^v_{\max}}, \tag{10}$$

where $\mathbf{C}^v_{\max}$ is the maximum in $\mathbf{C}^v$. From equation (8), we can get the following: when the transition probability distribution between samples $\mathbf{x}_i^v$ and $\mathbf{x}_j^v$ differs greatly, that is, the value of $\mathbf{C}^v(i, j)$ is very large, a smaller value of $\mathbf{S}^v(i, j)$ is assigned. On the contrary, when the difference of the transition probability distribution between samples $\mathbf{x}_i^v$ and $\mathbf{x}_j^v$ is small, that is, the value of $\mathbf{C}^v(i, j)$ is small, a great value of $\mathbf{S}^v(i, j)$ is assigned. Thus, the construction of the similarity matrix based on relative entropy is realized.

2.3. Similarity Network Fusion Based on Random Walk and Relative Entropy (R^2SNF). Through the above two steps, the similarity matrix $\mathbf{S}^v$ is obtained. In the similarity network fusion stage, we use $\mathbf{S}^v$ as a dense similarity matrix to obtain the global structure between samples and use the KNN similarity matrix to capture the local structure.

For any samples $\mathbf{x}_i^v$, KNN defines the similarity matrix $\mathbf{K}^v$ between $\mathbf{x}_i^v$ and its k most similar samples. The element $\mathbf{K}^v(i, j)$ in $\mathbf{K}^v$ is defined as

$$\mathbf{K}^v(i, j) = \begin{cases} \dfrac{\mathbf{W}^v(i, j)}{\sum_{k\in\mathbf{N}_i^v}\mathbf{W}^v(i, k)}, & j \in \mathbf{N}_i^v, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $\mathbf{N}_i^v$ is the neighbors of $\mathbf{x}_i^v$.

Assume that there is a total of V genomic data to be integrated. In the same way as SNF, we performed nonlinear iterative fusion for dense similarity matrix $\mathbf{S}^v$ and sparse similarity matrix $\mathbf{K}^v$ of each dataset. The fusion process can be described as

$$\widetilde{\mathbf{S}}^v = \mathbf{K}^v \times \left(\frac{\sum_{k\neq v}\mathbf{S}^k}{V-1}\right) \times \left(\mathbf{K}^v\right)^{\mathrm{T}}, \quad v = 1, 2, \ldots, V. \tag{12}$$

According to equation (12), we can obtain the similarity matrix $\widetilde{\mathbf{S}}^v$ of the cross-diffusion of the vth genomic data with other data. Then, the final fused similarity matrix $\mathbf{S}$ can be obtained by averaging all $\widetilde{\mathbf{S}}^v$:

$$\mathbf{S} = \frac{1}{V}\sum_{v=1}^{V}\widetilde{\mathbf{S}}^v. \tag{13}$$

2.4. Spectral Clustering on the Fused Similarity Matrix. Suppose we want to identify c cancer subtypes from multiple genomic data, so we need to use spectral clustering to cluster cancer samples into c clusters. For the ith sample, we defined a cluster indicator vector $\mathbf{y}_i \in \{0, 1\}$. When the ith sample belongs to the jth cluster, $\mathbf{y}_i(j) = 1$; otherwise, $\mathbf{y}_i(j) = 0$. The cluster indicator matrix can be written as $\mathbf{Y} = (\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_n^T)$.

With the fused similarity matrix $\mathbf{S}$, spectral clustering can be performed by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{U}} \operatorname{tr}\left(\mathbf{U}^T\mathbf{L}\mathbf{U}\right) \\ &\text{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \end{aligned} \tag{14}$$

where $\mathbf{U} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}$, $\mathbf{U} \in \mathbb{R}^{n\times c}$, is the scaled partition matrix. According to the fused similarity matrix $\mathbf{S}$, $\mathbf{L}$ as the normalized Laplacian matrix can be defined as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the degree matrix, which satisfies $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_n)$, $d_i = \sum_{j=1}^n \mathbf{S}(i, j)$. In this way, we can capture the global structure of the fused similarity matrix through spectral clustering.

3. Results and Discussion

3.1. Datasets and Survival Analysis. In this paper, we tested the proposed algorithm on three types of genomic data, that is, mRNA expression data, miRNA expression data, and DNA methylation data. The cancer types we tested include glioblastoma multiforme (GBM), breast invasive carcinoma (BIC), kidney renal clear cell carcinoma (KRCCC), lung squamous cell carcinoma (LSCC), and colon adenocarcinoma (COAD). The above data can be downloaded from the TCGA website [5]. In addition, we also conducted experiments on the BREAST cancer and LUNG cancer datasets in [26]. The detailed information of the cancer multigenomic datasets is shown in Table 1.

This paper conducts survival analysis based on the cancer subtypes obtained by clustering to verify the survival differences among samples of different cancer subtypes found by the proposed algorithm. In statistics, hypothesis testing is usually used to quantify whether there are differences between different survival curves. Here, the Cox log-rank test [27] is used to calculate the p value. Cox log-rank test is a nonparametric hypothesis test, which is often used to assess the importance of differences in survival between subtypes.

Table 1: Detailed information on seven types of cancer multigenomic datasets.

Cancer type	Number of genes			Number of samples
	mRNA	Methylation	miRNA	
GBM	12042	1305	534	215
BIC	17814	23094	354	105
KRCCC	17899	24960	329	122
LSCC	12042	23074	352	106
COAD	17814	23088	312	92
BREAST	20531	5000	1046	622
LUNG	20531	5000	1046	337

The p value indicates that the observed difference in survival is the likelihood of an incident occurring by chance. Therefore, the smaller the p value is, the better the experimental effect is. In addition, the Kaplan–Meier estimation method [28] is usually used to estimate the survival function and further obtain the Kaplan–Meier survival curve. The x-axis of the survival curve is the time from the beginning of observation to the last observation time point. The y-axis is the survival rate of the survival sample. The curve represents the development of the event.

3.2. Experimental Results. We compared the proposed algorithm R^2SNF with several cancer subtype prediction methods, e.g., SNF [6], LRAcluster [29], iClusterPlus [30], pattern fusion analysis (PFA) [31], affinity network fusion (ANF) [32], and multiview clustering based on Stiefel manifold (MCSM) [33], to verify its effectiveness. In order to verify whether the relative entropy in the R^2SNF algorithm can improve the prediction results of cancer subtypes, we remove the relative entropy from R^2SNF and use equation (5) to construct the similarity matrix. We name the above algorithm as similarity network fusion based on random walk (RSNF). A brief introduction to these methods is as follows:

(i) SNF first uses the exponential similarity kernel method to define the similarity between the sample points of each genomic data. It uses the KNN method to define a dense similarity matrix and a sparse similarity matrix. Then, the information transfer model is proposed to fuse the above two similarity matrices, and the fused similarity matrix can be obtained by updating iteratively. Finally, spectral clustering is used to cluster the fused similarity matrix.

(ii) LRAcluster is a dimensional reduction and clustering method for multigenomic data based on low-rank approximation. It can deal with a variety of distributed data classes and guarantee the orthogonality of the low-dimensional space. It is suitable for clustering analysis of large-scale multigenomic data and has been widely concerned and applied.

(iii) iClusterPlus considers that different variable types should follow different linear probability relationships. Then, it builds a joint sparse model to complete the task of sample clustering and feature selection.

(iv) PFA uses the local information extraction method to project each genomic data in a low-dimensional space and builds a dynamic collimation method based on the idea of manifold learning. Then, it integrates the low-dimensional spatial information into a feature space containing information from different genomic data. Finally, the K-means method is used to cluster the samples.

(v) ANF first constructs a patient affinity network from each omics data and then fuses all individual networks to obtain a more robust one. In order to make the patient affinity network robust to noise, ANF mainly employs two nonlinear k-nearest-neighbor- (kNN-) based transformations: kNN Gaussian kernel and kNN graph.

(vi) MCSF establishes a binary optimization model for the simultaneous clustering problem. Then, the optimization problem is solved by the linear search algorithm based on the Stiefel manifold. Finally, it integrated the clustering results obtained from multiomics data by using the k-nearest neighbor method.

(vii) RSNF obtains the probability of each sample starting from one sample and arriving at another via random walk, calculates the similarity matrix according to the random walk probability between the two samples, and finally performs similarity network fusion according to SNF.

Since R^2SNF is an improved version of SNF, in order to make a more intuitive comparison and analysis, we used the number of clusters suggested in SNF, that is, GBM is clustered into 3 categories, BIC is clustered into 5 categories, KRCCC is clustered into 3 categories, LSCC is clustered into 4 categories, and COAD is clustered into 3 categories. For the BREAST and LUNG datasets, we also used the cancer subtype determination method in SNF to determine the number of their cancer subtypes as 3 and 2, respectively.

The specific experimental results of R^2SNF and other methods on the seven cancer multigenomic datasets are shown in Table 2. Compared with RSNF, R^2SNF had better results on the other six datasets except for KRCCC data. This shows that using relative entropy to calculate the probability distribution difference between samples is beneficial to the construction of the similarity matrix. Compared with SNF, R^2SNF has smaller p values on all datasets except for COAD. The results of RSNF on GBM, BIC, KRCCC, and LSCC are better than SNF, especially on KRCCC and LSCC data, but slightly worse than SNF on other data, which indicates that only using the probability obtained by random walk between samples to construct the similarity matrix also has a certain effect on cancer subtypes. Compared with other algorithms, R^2SNF has the best results on the whole. Only on BIC data, MCSM algorithm is better than R^2SNF.

Figure 1 shows the Kaplan–Meier survival curve of cancer subtypes identified by R^2SNF on seven cancer

Table 2: Comparison of p values between R^{2}SNF and other algorithms on seven cancer multigenomic datasets.

Cancer type	Methods							
	R^{2}SNF	SNF	LRAcluster	iClusterPlus	PFA	ANF	MCSM	RSNF
GBM	**2.4E − 05**	2.0E − 04	3.5E − 04	3.0E − 03	8.0E − 05	5.8E − 04	1.1E − 03	1.2E − 04
BIC	1.1E − 04	1.1E − 03	4.3E − 02	3.5E − 02	9.3E − 03	3.6E − 04	**3.1E − 05**	1.2E − 04
KRCCC	7.0E − 03	2.9E − 02	3.2E − 02	1.1E − 01	7.5E − 03	2.9E − 02	8.0E − 02	**2.1E − 04**
LSCC	**1.5E − 05**	2.0E − 02	5.7E − 02	5.2E − 02	4.0E − 03	8.9E − 03	1.6E − 02	2.0E − 04
COAD	1.8E − 03	**1.3E − 03**	9.9E − 03	5.0E − 02	6.7E − 02	9.0E − 03	3.6E − 01	6.0E − 03
BREAST	**4.5E − 09**	1.0E − 08	3.0E − 01	1.5E − 01	3.6E − 07	1.9E − 08	7.4E − 07	2.3E − 08
LUNG	**5.6E − 03**	1.1E − 02	4.6E − 01	6.9E − 01	2.0E − 01	1.0E − 02	8.0E − 02	8.2E − 02

The best results have been highlighted in bold.

genomic datasets. It can be seen that, on GBM, KRCCC, LSCC, COAD, BREAST, and LUNG, there is a big difference between the cancer subtypes recognized by R^{2}SNF, indicating that R^{2}SNF is an effective method for identifying cancer subtypes. On BIC data, SNF suggested to divide it into 5 cancer subtypes. As shown in Figure 1(b), R^{2}SNF is not very effective when divided into 5 subtypes, but it can clearly divide it into 3 subtypes. Moreover, the p value of SNF on the BIC data is lower than the p value of SNF. Therefore, we recommend that BIC should be divided into 3 subtypes. The number of clusters given in the BREAST dataset in [26] is 3, which can be found in Figure 1(f). This further verifies our conclusion.

3.3. Analysis on the GBM Dataset. Glioblastoma multiforme (GBM) is the most common and lethal malignant primary brain tumor in adults and is one of a group of tumors known as gliomas. Many studies have carried out research on GBM at the molecular level. And clinically, some studies have given definite cancer subtypes and corresponding treatment plans. For example, based on mRNA expression data, Verhaak et al. [34] divided GBM into four cancer subtypes: mesenchymal, classical, neural, and proneural. In [35], according to the difference of the CpG island methylator phenotype (CLMP), GBM was divided into two cancer subtypes: G-CLMP and non-G-CLMP.

On GBM data, we counted the distribution of clustering results obtained by R^{2}SNF on the cancer subtypes determined in the above two studies and summarized the results in Table 3. Table 3 shows that the patients in subtype 1 are more than in subtype 3. Most patients in subtype 1 are grouped into non-G-CLMP (accounted for 99.3%); also, they are distributed on four subtypes in [34]. Subtypes 2 and 1 have similar distributions. It is worth noting that most of the 19 patients with subtype 3 are of the G-CLMP subtype (accounted for 73.7%), and all of them are of the proneural subtype.

To further analyze the obtained cancer subtypes by R^{2}SNF, the clinical data for all patients of GBM were downloaded from the cBio Cancer Genomics Portal database. We drew a boxplot of the age distribution of patients in the three cancer subtypes (Figure 2). Figure 2 proves that the cancer subtypes identified by R^{2}SNF have a clear age distribution difference. Combining Figures 1 and 2, we can find that the age of patients in subtype 3 with the best survival advantage in Figure 1 is also lower than that of patients in subtypes 1 and 2.

Furthermore, we drew Kaplan–Meier survival curves of GBM patients' response to the drug temozolomide (TMZ) in Figure 3. The patients within the three cancer subtypes were divided into two parts: patients treated with drug TMZ and those not treated with drug TMZ. TMZ is a drug that is commonly used to treat GBM, but only responds well to a subset of patients. The p values of survival analysis in the Cox log-rank model of the three cancer subtypes are 5.42×10^{-6}, 3.78×10^{-4}, and 0.36, respectively, which indicate that TMZ has no effect on the patients in cancer subtype 3.

In summary, subtype 3 of GBM identified by R^{2}SNF has the following characteristics. First, most of the patients with subtype 3 are of the G-CLMP subtype, and all of them are of the proneural subtype. Second, the age of patients in subtype 3 with the best survival advantage is also lower than that of patients in subtypes 1 and 2. Third, TMZ has no effect on the patients in cancer subtype 3. Therefore, we believe that subtype 3 identified by R^{2}SNF is a biologically significant cancer subtype. In addition, it can be inferred that we get a potential cancer subtype, which contains patients belonging to both G-CLAMP and Proneural. This verified the study reported by Brennan et al. that the proneural subtype granted by the G-CIMP phenotype has unique properties [36].

3.4. Analysis on the BREAST Dataset. Breast cancer refers to a malignant tumor in which cancer cells have penetrated the basement membrane of breast ducts or lobular alveoli and invaded the interstitium. Many scholars have carried out a series of studies and analyses on the gene level and have given specific subtypes and treatment programs. Based on the microarray predictive analysis model, Parker et al. proposed a 50-gene classifier (known as PAM50) to classify BIC into five subtypes: basal-like, luminal A, luminal B, HER2-enriched, and normal-like [37]. On BREAST data, we counted the distribution of clustering results obtained by R^{2}SNF on the cancer subtypes basal-like, luminal A, luminal B, and HER2-enriched in Table 4. It can be seen from Table 4 that subtype 1 is mainly distributed in luminal A and luminal B (accounted for 80.6%), subtype 2 is mainly distributed in basal-like (accounted for 74.6%), and subtype 3 is mainly distributed in luminal A and luminal B (accounted for

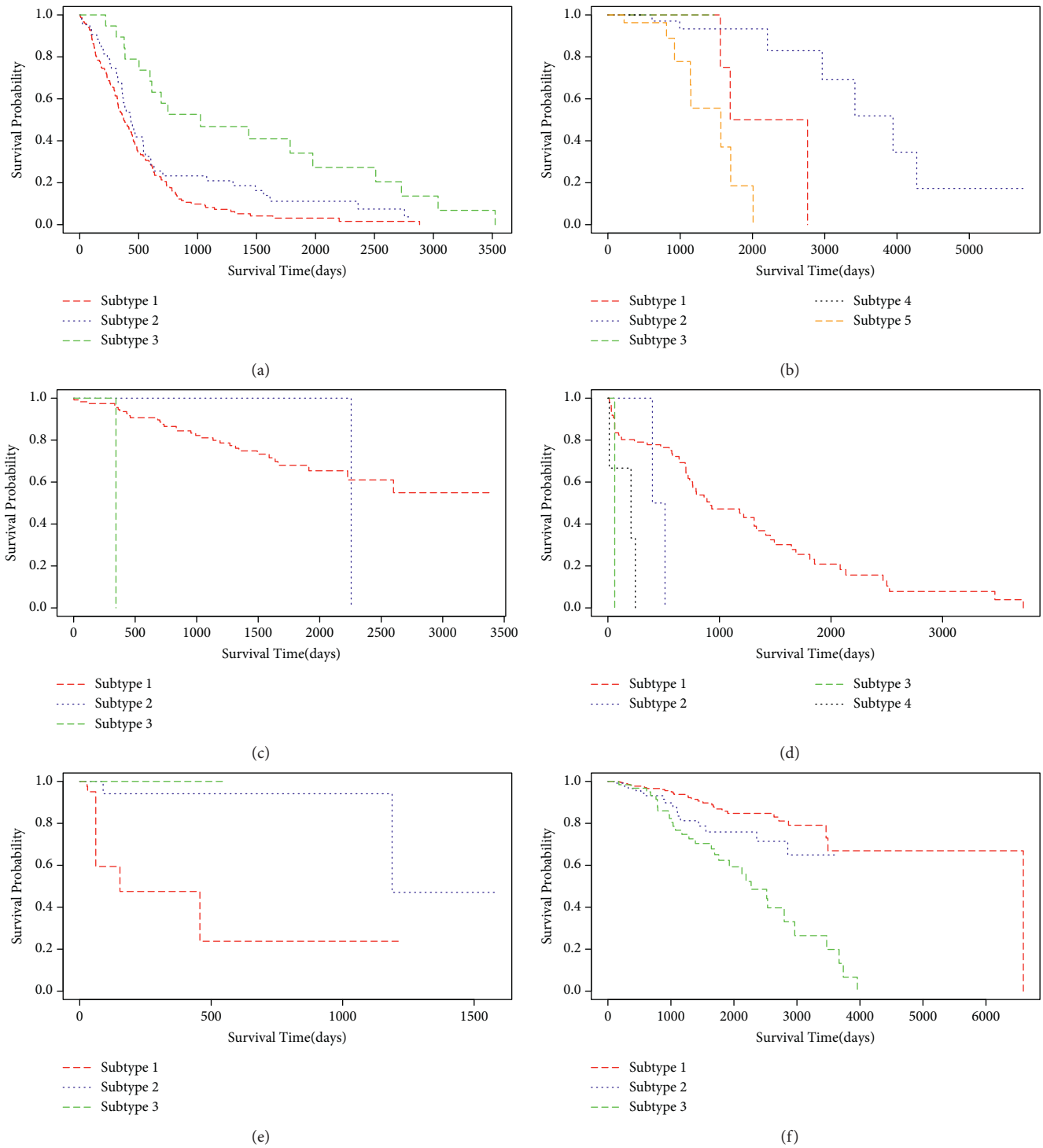
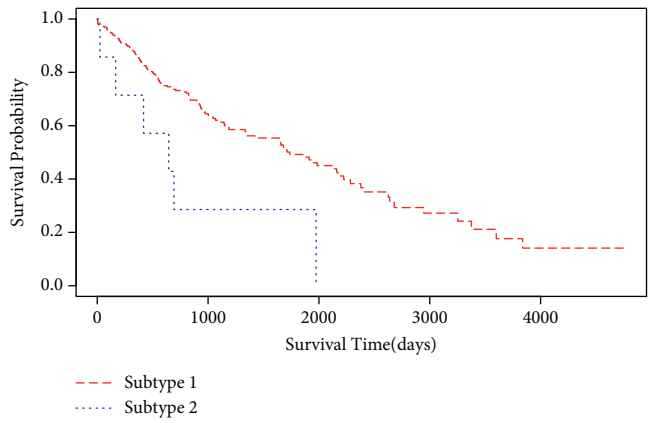


Figure 1: Continued.



(g)

Figure 1: Kaplan–Meier survival curves of different subtypes on cancer multigenomic datasets: (a) GBM, (b) BIC, (c) KRCCC, (d) LSCC, (e) COAD, (f) BREAST, and (g) LUNG.

Table 3: The distribution of subtypes obtained by R^2SNF on the subtypes determined in [34, 35].

R^2SNF subtypes	Subtypes in [34]				Subtypes in [35]	
	Mesenchymal	Classical	Neural	Proneural	G-CLMP	Non-G-CLMP
Subtype 1	46	51	26	30	1	152
Subtype 2	20	6	8	9	4	39
Subtype 3	0	0	0	19	14	5

The values in this table represent the number of patients counted.

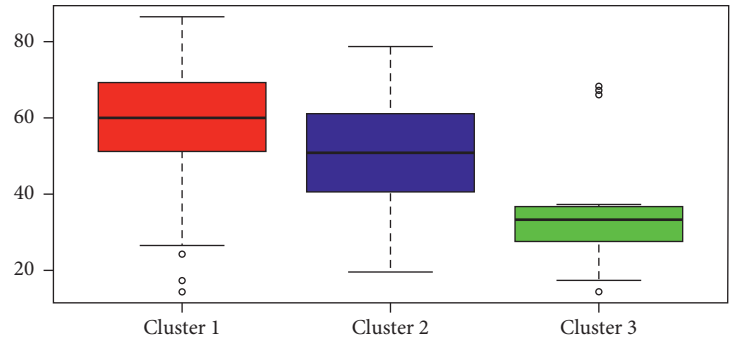


Figure 2: Boxplot of the age distribution of patients in the three cancer subtypes. The black bar represents the median of each subtype.

70.8%). In addition, we can also find that HER2-enriched is mainly distributed in subtypes 1 and 2 (accounted for 89.1%), and normal-like is mainly distributed in subtype 1 (accounted for 78.3%).

We also chose two clinical labels for which we tested enrichment: Pathologic M and Pathologic N. Pathologic M and Pathologic N are regional lymph nodes' distant metastasis stage (M) and clinical stage (N) of breast cancer, respectively. Pathologic M includes three stages: M0, M1, and MX. Pathologic N roughly includes five stages: N0, N1, N2, N3, and NX. Generally, the numbers or letters after N and M provide more details about these factors, and the higher the number, the more severe the cancer.

We used the chi-square test to verify whether there was a significant difference in our analysis among these clinical labels. The p values on Pathologic M and Pathologic N are 6×10^{-3} and 9×10^{-3}, respectively. The detailed distributions of subtypes obtained by R^2SNF on Pathologic M and Pathologic N are shown in Tables 5 and 6, respectively. In Table 5, subtype 1, subtype 2, and subtype 3 have the similar distribution: mainly distributed in M0. We calculated the proportion of samples belonging to the M0 stage in the three subtypes as 74.9%. In Table 6, subtype 1, subtype 2, and subtype 3 have the similar distribution: mainly distributed in N0 and N1. The proportion of samples belonging to the N0 stage and N1 stage in the three subtypes is 46.3% and 33.8%, respectively.

From the above analysis, we can draw the following conclusion. First, subtypes 1 and 3 are mainly distributed in luminal A and luminal B, which are the breast cancer subtypes with the best prognosis. Second, subtype 2 is mainly distributed in basal-like, in which clinical prognosis

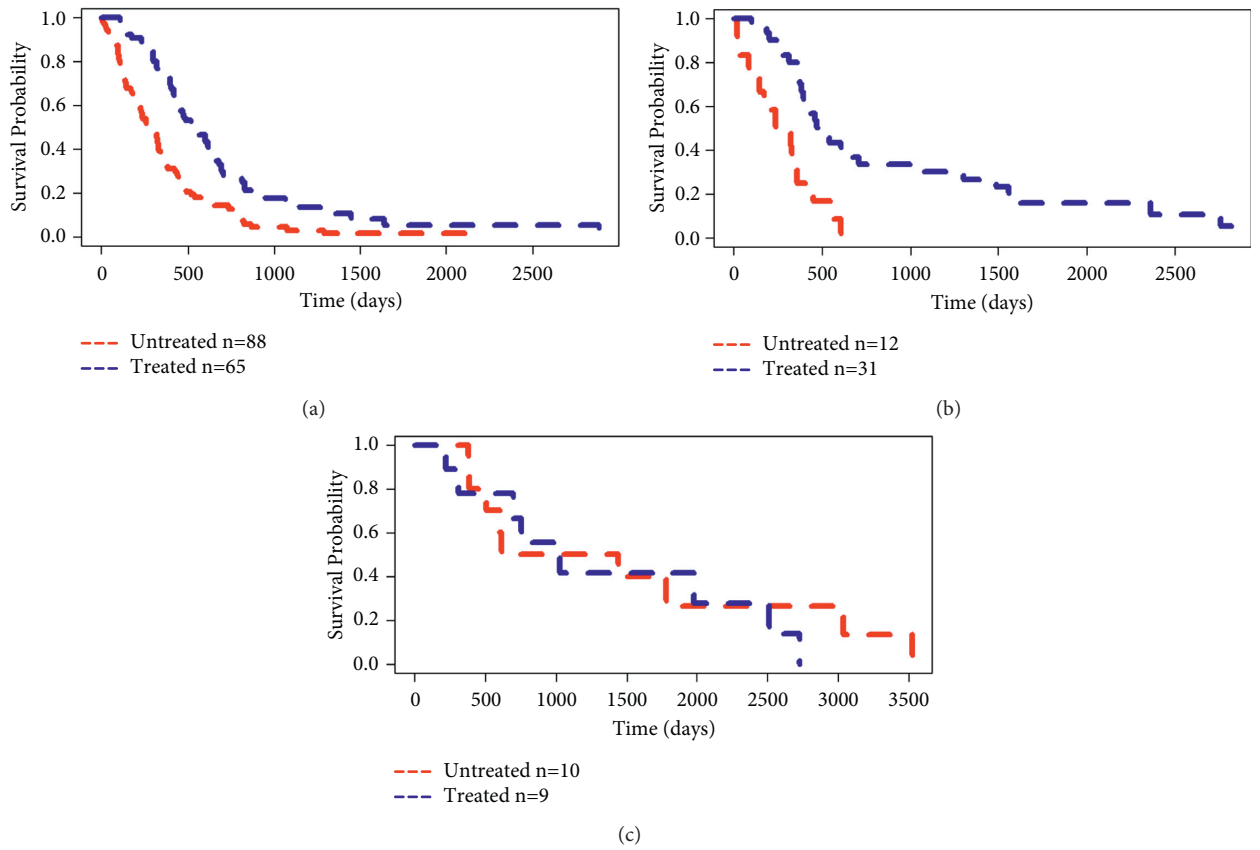


Figure 3: The Kaplan–Meier survival curves of the identified cancer subtypes by R^{2}SNF: (a) subtype 1, (b) subtype 2, and (c) subtype 3 of TMZ response. "Untreated" represents the patients who did not receive TMZ treatment, and "Treated" represents the patients who received TMZ treatment.

Table 4: The distribution of subtypes obtained by R^{2}SNF on the subtypes determined by PAM50.

R^{2}SNF subtypes	Subtypes determined by PAM50				
	Basal-like	HER2-enriched	Luminal A	Luminal B	Normal-like
Subtype 1	8	28	242	102	47
Subtype 2	97	21	2	4	6
Subtype 3	6	6	27	19	7

Table 5: The distribution of subtypes obtained by R^{2}SNF on Pathologic M.

R^{2}SNF subtypes	Pathologic M		
	M0	M1	MX
Subtype 1	305	4	118
Subtype 2	101	2	27
Subtype 3	60	1	4

Table 6: The distribution of subtypes obtained by R^{2}SNF on Pathologic N.

R^{2}SNF subtypes	Pathologic N				
	N0	N1	N2	N3	NX
Subtype 1	185	149	45	42	6
Subtype 2	79	34	13	4	0
Subtype 3	24	27	9	2	3

is poor. Third, the patients in BREAST data are mainly in the early stages of breast cancer and have high survival rate. All these conclusions can also be verified in Figure 1(f).

4. Conclusions

How to construct a robust dense similarity matrix is a key issue in SNF. In this paper, we analyzed the problems existing in the construction of the dense similarity matrix in SNF and proposed the similarity network fusion based on random walk and relative entropy (R^2SNF) method for cancer subtypes' prediction. We proposed to use the random walk with restart algorithm to characterize the complex relationship between genomic data samples and obtained the stable state transition probability distribution of each sample. We further used relative entropy to calculate the difference in the transition probability distribution between samples to construct a better dense similarity matrix which contains structural similarity information between samples. Then, the constructed dense similarity matrix and the KNN similarity matrix were nonlinearly iteratively fused. Finally, spectral clustering was used to cluster the fused similarity matrix. On seven cancer genomic datasets (GBM, BIC, KRCCC, LSCC, COAD, BREAST, and LUNG) containing three data types (mRNA expression data, miRNA expression data, and DNA methylation data), R^2SNF was compared with a variety of classical cancer subtype prediction algorithms. Experimental results show that R^2SNF has better performance in identifying cancer subtypes than the comparison algorithms. And through the analysis of the results of GBM and BREAST experiments, it can be proved that R^2SNF can discover cancer subtypes with biological significance. In addition to relative entropy, there are other methods to measure the difference between two probability distributions, such as Jensen–Shannon divergence, Wasserstein distance, and cross-entropy. In future work, we will devote ourselves to finding a more suitable method to calculate the difference between probability distributions and then to obtain a similarity matrix that is conducive to cancer subtype prediction.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61906198, 61976215, and 61772532) and the Natural Science Foundation of Jiangsu Province (Grant no. BK20190622).

References

[1] G. Getz, S. Gabriel, K. Cibulskis et al., "Integrated genomic characterization of endometrial carcinoma," *Nature*, vol. 116, no. 7447, pp. 67–73, 2013.

[2] J. Xi, X. Yuan, M. Wang et al., "Inferring subgroup-specific driver genes from heterogeneous cancer samples via subspace learning with subgroup indication," *Bioinformatics*, vol. 36, no. 6, pp. 1855–1863, 2019.

[3] C. M. Croce, "Oncogenes and cancer," *New England Journal of Medicine*, vol. 358, no. 5, pp. 502–511, 2008.

[4] A. Chen, G. Fu, Z. Xu et al., "Detection of bladder cancer via microfluidic immunoassay and single-cell DNA copy number alteration analysis of captured urinary exfoliated tumor cells," *Cancer Research*, vol. 78, no. 14, pp. 4073–4085, 2017.

[5] D. Capper, D. T. W. Jones, M. Sill et al., "DNA methylation-based classification of central nervous system tumours," *Nature*, vol. 555, no. 7697, pp. 469–474, 2018.

[6] B. Wang, A. M. Mezlini, F. Demir et al., "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, no. 3, pp. 333–337, 2014.

[7] S. Hanash, "Integrated global profiling of cancer," *Nature Reviews Cancer*, vol. 4, no. 8, pp. 638–644, 2004.

[8] R. Shen, A. B. Olshen, M. Ladanyi et al., "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis," *Bioinformatics*, vol. 26, no. 2, pp. 292-293, 2010.

[9] U. D. Akavia, O. Litvin, J. Kim et al., "An integrated approach to uncover drivers of cancer," *Cell*, vol. 143, no. 6, pp. 1005–1017, 2010.

[10] M. Liang, Z. Li, T. Chen, and J. Zeng, "Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 4, pp. 928–937, 2015.

[11] N. K. Speicher and N. Pfeifer, "Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery," *Bioinformatics*, vol. 31, no. 12, pp. i268–75, 2015.

[12] H. Wang, H. Zheng, J. Wang, C. Wang, and F.-X. Wu, "Integrating omics data with a multiplex network-based approach for the identification of cancer subtypes," *IEEE Transactions on NanoBioscience*, vol. 15, no. 4, pp. 335–342, 2016.

[13] T. L. Van, L. M. Van, and A. C. Fierro, "Simultaneous discovery of cancer subtypes and subtype features by molecular data integration," *Bioinformatics*, vol. 32, no. 17, pp. 445–454, 2016.

[14] E. Zhang and X. Ma, "Regularized multi-view subspace clustering for common modules across cancer stages," *Molecules*, vol. 23, no. 5, p. 1016, 2018.

[15] M. Hofree, J. P. Shen, H. Carter et al., "Network-based stratification of tumor mutations," *Nature Methods*, vol. 10, no. 11, pp. 1108–1115, 2014.

[16] J. K. Huang, T. Jia, D. E. Carlin, and T Ideker, "pyNBS: a Python implementation for network-based stratification of tumor mutations," *Bioinformatics*, vol. 34, no. 16, pp. 2859–2861, 2018.

[17] T. Xu, L. T. Duy, L. Lin, R. Wang, B. Sun, and J. Li, "Identifying cancer subtypes from miRNA-TF-mRNA regulatory networks and expression data," *PloS One*, vol. 11, no. 4, Article ID e0152792, 2016.

[18] C. Yang, S.-G. Ge, and C.-H. Zheng, "ndmaSNF: cancer subtype discovery based on integrative framework assisted by network diffusion model," *Oncotarget*, vol. 8, no. 51, pp. 89021–89032, 2017.

[19] B. Yang, Y. Zhang, S. Pang, X. Shang, X. Zhao, and M. Han, "Integrating multi-omic data with deep subspace fusion clustering for cancer subtype prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 1, pp. 216–226, 2019.
[20] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
[21] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.
[22] H. Tong, C. Faloutsos, J. Pan et al., "Fast random walk with restart and its applications," in *Proceedings of the International Conference on Data Mining*, pp. 613–622, Washington, DC, USA, December 2006.
[23] V. Martinez, F. Berzal, and J. Cubero, "A survey of link prediction in complex networks," *ACM Computing Surveys*, vol. 49, no. 4, p. 69, 2017.
[24] W. Zheng, S. Liu, and J. Mu, "A random walk similarity measure model based on relative entropy," *Journal of Nanjing University (Natural Science)*, vol. 55, no. 6, pp. 984–999, 2019.
[25] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[26] N. Rappoport and R. Shamir, "Multi-omic and multi-view clustering algorithms: review and cancer benchmark," *Nucleic Acids Research*, vol. 46, no. 20, pp. 10546–10562, 2018.
[27] K. Akazawa, T. Nakamura, and Y. Palesch, "Power of logrank test and cox regression model in clinical trials with heterogeneous samples," *Statistics in Medicine*, vol. 16, no. 5, pp. 583–597, 1997.
[28] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958.
[29] D. Wu, D. Wang, M. Q. Zhang, and J. Gu, "Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification," *BMC Genomics*, vol. 16, no. 1, p. 1022, 2015.
[30] Q. Mo, S. Wang, V. E. Seshan et al., "Pattern discovery and cancer gene identification in integrated cancer genomic data," *Proceedings of the National Academy of Sciences*, vol. 110, no. 11, pp. 4245–4250, 2013.
[31] Q. Shi, C. Zhang, M. Peng et al., "Pattern fusion analysis by adaptive alignment of multiple heterogeneous omics data," *Bioinformatics*, vol. 33, no. 17, pp. 2706–2714, 2017.
[32] T. Ma and A. Zhang, "Affinity network fusion and semi-supervised learning for cancer patient clustering," *Methods*, vol. 145, pp. 16–24, 2018.
[33] J. Tian, J. Zhao, and C. Zheng, "Clustering of cancer data based on Stiefel manifold for multiple views," *BMC Bioinformatics*, vol. 22, no. 1, p. 268, 2021.
[34] R. G. W. Verhaak, K. A. Hoadley, E. Purdom et al., "Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1," *Cancer Cell*, vol. 17, no. 1, pp. 98–110, 2010.
[35] H. Noushmehr, D. J. Weisenberger, K. Diefes et al., "Identification of a CpG island methylator phenotype that defines a distinct subgroup of glioma," *Cancer Cell*, vol. 17, no. 5, pp. 510–522, 2010.
[36] C. W. Brennan, R. G. Verhaak, A. McKenna et al., "The somatic genomic landscape of glioblastoma," *Cell*, vol. 155, no. 2, pp. 462–477, 2013.
[37] J. S. Parker, M. Mullins, M. C. U. Cheang et al., "Supervised risk predictor of breast cancer based on intrinsic subtypes," *Journal of Clinical Oncology*, vol. 27, no. 8, pp. 1160–1167, 2009.

Hindawi
Scientific Programming
Volume 2021, Article ID 9937061, 14 pages
https://doi.org/10.1155/2021/9937061

Research Article

Improved Deep Hashing with Scalable Interblock for Tourist Image Retrieval

Jiangfan Feng and Wenzheng Sun

Chongqing University of Posts and Telecommunications, College of Computer Science and Technology, Chongqing, China

Correspondence should be addressed to Jiangfan Feng; fengjf@cqupt.edu.cn

Received 17 March 2021; Revised 16 June 2021; Accepted 5 July 2021; Published 14 July 2021

Academic Editor: Boxiang Dong

Tourist image retrieval has attracted increasing attention from researchers. Mainly, supervised deep hash methods have significantly boosted the retrieval performance, which takes hand-crafted features as inputs and maps the high-dimensional binary feature vector to reduce feature-searching complexity. However, their performance depends on the supervised labels, but few labeled temporal and discriminative information is available in tourist images. This paper proposes an improved deep hash to learn enhanced hash codes for tourist image retrieval. It jointly determines image representations and hash functions with deep neural networks and simultaneously enhances the discriminative capability of tourist image hash codes with refined semantics of the accompanying relationship. Furthermore, we have tuned the CNN to implement end-to-end training hash mapping, calculating the semantic distance between two samples of the obtained binary codes. Experiments on various datasets demonstrate the superiority of the proposed approach compared to state-of-the-art shallow and deep hashing techniques.

1. Introduction

With the rise of cheap sensors, mobile terminals, and social networks, research on tourist images is making good progress, which results in an explosive growth of image retrieval in social networks. This trend imposes great challenges on developing scalable indexing approaches, supporting retrieving relevant images of such massive tourist images. However, current tourist image retrieval mainly relies on manual tags in sensor types, tourist sights, and geographical locations. For example, SIFT [1] uses local descriptors to encode image regions of interest, for example, HOG [2] and BOW [3]. Consequently, it is highly dependent on the availability and quality of tags.

Due to the fast query speed and low storage cost, learning-based hash has been attracting research interests and was applied to applications such as large-scale object retrieval [4], image classification [5], and detection [3]. Recently, deep learning using hash methods has shown promising performance [6, 7]. Due to the high efficiency of binary hash code in the computation of Hamming distance and the advantage of storage space, it is very efficient in large-scale image retrieval. Convolutional neural network hashing (CNNH) [8] incorporates deep neural networks into hash coding to learn the image representations and hash codes. Network in network hashing (NINH) [9] presents a triplet ranking loss to capture the relative similarities of images. The image representation learning and hash coding can benefit each other within a staged framework. Deep semantic hashing [10] ultimate hash codes produced by the learned hash functions maintain sentiment-level similarity. Other hashing methods have also been proposed [11–13].

Although hashing methods have achieved remarkable performance, they still suffer from the two following problems:

(1) Existing methods learn binary hash codes with hand-crafted feature representations, which cannot accurately capture the inherent semantic similarities of images

(2) In most existing hashing methods for images, the semantic similarities are defined at the image level,

and each picture is represented by one piece of hash code

This paper considers large-scale retrieval for multilabel tourist image data, which includes semantic hashing and category-aware hashing. We propose an architecture of deep convolution networks designed for hash learning, which has substantially superior performance on large-scale tourist images by end-to-end learning discriminative short binary code. As a whole, the main contributions of this paper are as follows:

(1) For binary hash optimization, we propose a discrete hash optimization strategy based on the inner relationship for learning hash codes without relaxing the quantization information loss.

(2) We provide an improved divide-code layer, substituting for fully connected layers to learn binary hash code to reduce high redundancy and parameters in the retrieval task. Besides, we use an improved triplet loss function to guarantee the feature similarity to the binary code features to improve the algorithmic efficiency while training.

(3) In terms of applications, the deep hash method is employed for large-scale tourist image retrieval. Consequently, this paper illustrates ways to design and train a deep network of large-scale tourist image retrieval.

2. Related Works

This section briefly reviews two topics: (1) tourist image retrieval models and (2) hashing retrieval models.

2.1. Tourist Image Retrieval Model. Numerous tourist image retrieval methods based on landmark datasets have been proposed. They often use visual descriptors to describe images. The key is how to improve the expressive ability of visual descriptors. For example, Hao et al. [14] and Xiao et al. [15] used multidimensional models to sort in space property and utilized the three-dimensional visual phrase to describe the landmark images. However, these methods have the disadvantages of long-time modeling and high retrieval cost. Recently, to reduce the cost of retrieval, many researchers began to devote themselves to the research of binary images that compose the landmark features of high-dimensional visual words. Ji et al. [16] proposed a Location Discriminative Vocabulary Coding (LDVC) scheme, which achieves deficient bit rate query transmission, discriminative landmark description, and scalable descriptor delivery in a unified framework. Duan et al. [17] combined multiple information, such as image, GPS, and crowd-sourced hotspot Wi-Fi, to extract location discriminative compact image descriptors. Zhou et al. [18] used the scalable cascaded hashing (SCH) method to implement the landmark hashing retrieval. Zhu et al. [19] used a discrete multimodal hash scheme (Cv-Dmh) based on a canonical view to learn binary code through a new three-stage learning process. Jing et al. [20] investigated the spatiotemporal dynamic patterns of inbound tourism. Cui et al. [21] proposed a Scalable deep hashing (SCADH) to learn enhanced hash codes for social image retrieval.

Furthermore, complex network theory has been used to mine tourism flow patterns [22]. These methods are based on the feature extraction of the image, and then the hashing algorithm is used for iterative computation. However, no method of them is an end-to-end method to learn the hash function. Furthermore, most methods still use hand-craft features to extract image features, which have a weak generalization and migration ability.

Recent examples in which deep learning has made significant advances in tourist image retrieval include positioning the city [23] and tourist photo classification [24]. In addition, many studies have been conducted to analyze the tourist's urban image by modifying the classifier part of the CNN model [25] or considering local characteristics [26]. However, these studies are limited in reflecting the unique landscape or regional characteristics in the area.

2.2. Hashing Retrieval Model. Learning-based hashing retrieval methods can be divided into unsupervised methods and supervised methods. Unsupervised learning has a catalytic effect in reviving interest in hashing retrieval but has been overshadowed by the successes of purely supervised learning. The researchers introduced unsupervised learning procedures that only use the information on image samples without requiring supervision information for hashing. Notable examples in this category include local sensitive hashing (LSH) [27], iterative quantization (ITQ) [28], direct graph hashing (DGH) [29], scalable graph hashing (SGH) [30], and spectral hashing (SH) [31]. Unsupervised training of hashing retrieval is regarded as a "pretraining" phase whose role is to discover good features that model the structure in the input domain. Besides, supervised methods learn hash coding using both feature information and label, including minimum loss hashing [32], kernel-based supervised hashing (KSH) [33], ranking-based supervised hashing (RSH) [34], and column generation hashing (CGH) [35].

New advances in machine learning using deep neural networks enable automated learning of hashing functions. Xia et al. [36] applied deep hashing using a similarity matrix and minimized loss function to discover an approximate hash code. Although it has dramatically improved the retrieval performance, it is still not an accurate end-to-end method. Zhao et al. [37] proposed a deep hashing algorithm for sorting tags. Since image retrieval aims to return an image based on the correlation among the pictures, this approach is optimized for the final evaluation index. Lin et al. [38] proposed a straightforward method to obtain hash values. They added a fixed-length hidden layer to the CNN network that is limited by the activation function. After fine-tuning the CNN network, the hidden layer value is extracted directly. The number of nodes in the hidden layer is the length of the hash code. Although the eigenvalues obtained by this method contain the high-level semantics of the image, the process does not consider the correlation of the

Hamming space features. Therefore, it cannot guarantee the retrieval effect of the elements in the Hamming space.

Later, Lai et al. [9] proposed a training method based on the triplet. Training the objective function is to distance similar images in the Hamming space closer than dissimilar images. Recently, some semisupervised deep hashing models are proposed to utilize unlabeled data to improve retrieval accuracy. Yan et al. [39] proposed the BGDH method to learn embeddings and features simultaneously, as well as hash codes. Zhang and Peng [40] developed a deep hashing method SSDH, which maintains the underlying data structures and the semantic similarity simultaneously to learn hash functions. Both ways use a graph to model unlabeled training samples, which are computationally expensive and memory hog, especially with a large-scale dataset. Shi et al. [41] used the GAN and a discriminative model to learn from both the unlabeled data and labeled data to augment the training dataset, which may not be adapted to semantic representation. Tu et al. developed RDUH [42], which focuses on reducing noisy points by investigating the various input data structures.

Recently, cross-modal hashing methods have provided insight into capturing the intrinsic relationships between various modalities [43] and quantization-based cross-modal similarity [44]. Furthermore, Deng et al. [45] showed that semantic similarity of the training data could perform binary hash codes in an unsupervised manner. However, natural images can have significant intraclass and minor interclass variations. Thus, learning hash codes with class-specific representation centers is required [46]. To further bridge the inherent modality gap, a multitask consistency-preserving adversarial hashing (CPAH) [47] was proposed to fully explore the semantic consistency and correlation between different modalities for efficient cross-modal retrieval.

3. The Proposed Method

In this section, we present the details of our proposed method. We first define the notations used in this paper. Then, we introduce our deep feature learning process, deep hash model training process, and hash codes learning process. Finally, we present a hash optimization solution for solving hash codes and functions and analyzing their convergence and complexity.

3.1. Notations and Problem Definitions. For a tourist image dataset consisting of n images $\{x_i\}_{i=1}^{n}$ with l user-provided semantic tags, each image is represented by $x_i \in R^d$ and the relationships between the image and tags can be represented as l-dimensional binary-valued vector f_i. The image matrix is denoted as $\Theta \in R^{d\times n}$, and $F = [f_1, \ldots, f_n] \in R^{l\times n}$ represents the observed image-tag relation matrix.

We aim to learn a set of hash codes $B = \{b_1, b_2, \ldots, b_n\}$ with $b_i = H(x_i)$, $b_i \in \{0,1\}^c$, where c is the length of binary code and $h(\cdot)$ is the hash function. The binary code should guarantee the similarity of the original data space. Generally, the hash function $H(x)$ satisfies the following:

(1) b_i and b_j are closer in the Hamming space when $s_{ij} = 1$

(2) b_i and b_j are far away in the Hamming space when $s_{ij} = 0$

From the view of geographical position semantics, tourist images and the accompanying tags are highly correlated. These tags contain explicit semantics that is complementary to the latent image semantics. Hence, it is promising to exploit the refined auxiliary social tags for the semantic enrichment of image hash codes. To this end, we introduce a semantic correlation matrix $\mathbf{W}$ that directly correlates hash codes with refined social tags. The dynamic semantics can be directly transferred to hash codes. We aim to minimize the difference between the binary hash codes and the mapped semantic vectors from the refined tags.

We propose an architecture of deep convolution networks designed for hash learning, as shown in Figure 1. In detail, we build an end-to-end learning framework that utilizes hash mapping for tourist attraction image retrieval. The method is divided into three parts. The first is a subnetwork with multiple convolutions and pooling layers for learning discriminative image features, pretrained on the Place-2 dataset [48]. The second is the hash layer, which consists of a block coding layer and an activation function. The third is the improved triplet loss function that we use as the objective function to optimize the network. The training process is divided into many minibatches for iterative learning. Each small batch uses multiple images which belong to different categories as input.

3.2. Feature Learning and Deep Convolution Subnetwork Module. Most existing hashing methods adopt hand-crafted features for hash function learning. However, these methods may achieve limited performance because the hand-crafted features might not be optimally compatible with the hash function learning procedure. We propose our deep convolution subnetwork module, which can perform simultaneous feature learning and hash learning in the same framework. The subnetwork is used to learn the image features that can describe the image accurately. After training, the input image is processed through the network to obtain rich semantic descriptors with excellent expressiveness and robustness.

The tags from tourist images are subject to two properties: low rank and error sparsity. In such cases, we use VGG-16 as the subnetwork and transfer the model parameters trained on the Place-365 dataset to the network as the initial parameters. Since the scene recognition task has some similarities with the tourist attraction recognition task, transferring the setting from the network trained on Place-365 to the subnetwork can significantly improve the model's performance. The concrete structure of the network is shown in Table 1, which contains five large convolutional layers, five pooling layers, and two fully connected layers. Each large convolutional layer is followed by a 2×2 maximum pooling of 2 steps, and the detailed network configuration is shown in Table 1.

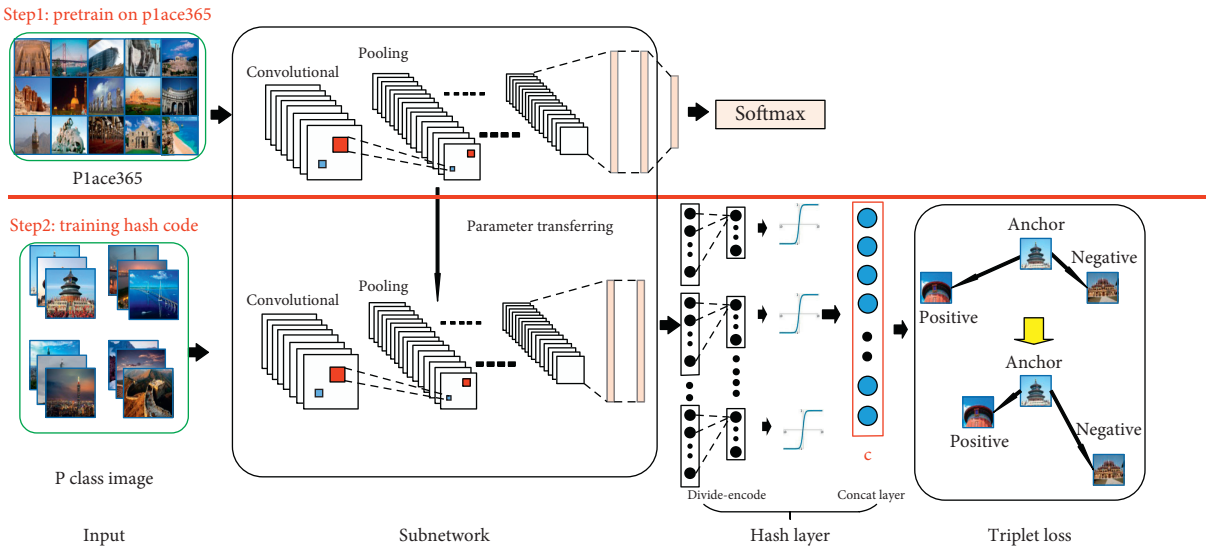


Figure 1: The proposed deep hash retrieval framework consists of the subnetwork hash layer and the improved triplet loss function. subnetwork is a multilayer convolution neural network. This hash layer comprises the divide-encode layer and the concat layer, and the length of the concat layer c represents the hash code length. The parameter transferring process transfers the pretrained parameters on Place-2 to the subnetwork as the initial parameter to train the hash mapping function.

Table 1: The subnetwork structure.

Type	Filter size/stride	Output size
Conv1-1	$3\times3/1$	$224\times224\times64$
Conv1-2	$3\times3/1$	$224\times224\times64$
Max pool1	$2\times2/2$	$112\times112\times64$
Conv2-1	$3\times3/1$	$112\times112\times128$
Conv2-1	$3\times3/1$	$112\times112\times128$
Max pool2	$2\times2/2$	$56\times56\times128$
Conv3-1	$3\times3/1$	$56\times56\times256$
Conv3-2	$3\times3/1$	$56\times56\times256$
Conv3-3	$3\times3/1$	$56\times56\times256$
Max pool3	$2\times2/2$	$28\times28\times256$
Conv4-1	$3\times3/1$	$28\times28\times512$
Conv4-2	$3\times3/1$	$28\times28\times512$
Conv4-3	$3\times3/1$	$28\times28\times512$
Max pool4	$2\times2/2$	$14\times14\times512$
Conv5-1	$3\times3/1$	$14\times14\times512$
Conv5-2	$3\times3/1$	$14\times14\times512$
Conv5-3	$3\times3/1$	$14\times14\times512$
Max pool5	$2\times2/2$	$7\times7\times512$
Fc6	—	4096
Fc7	—	4096

3.3. Hash Code Learning. Most existing studies use metric learning to train the positive and negative sample pairs to ensure the binary code similarity relationship [49–52]. However, it is challenging to represent geographic characteristics as a single binary code without losing a significant amount of helpful information. Hence, there is no need to conduct such an evaluation globally, but only among segments with users' geographic information needs. For example, a single tourist image can be represented into multiple binary vectors by treating each block as an image feature.

Tourist images and the accompanying tags are positively correlated with each other. Moreover, these tags contain explicit semantics, which is complementary to the latent image semantics. Hence, it is promising to exploit the refined auxiliary social tags for the semantic enrichment of image hash codes. To this end, we aim to minimize the difference between the binary hash codes and the mapped semantic vectors from the refined social tags.

This paper uncovered the intrinsic low-rank matrix by decomposing the image-tag relation matrix into its low-rank and sparse components. The low-rank matrix is then taken into Semantic Enhancement as a semantic source to enhance the discriminative capability of the learned hash codes. Therefore, we use a block-coded structure instead of a fully connected layer to implement a hash layer consisting of a block-coded layer, an active layer for each subblock, and a concat layer.

Consider a tourist image dataset consisting of n images $\{x_i\}_{i=1}^{n}$; we divide the features of fully connected layer $v(x_i)$ into q blocks. q denotes the length of the binary hash code for constructing the block-coded structure. The subfeatures $v(x_i)_j$ are obtained from the j-th slice layer as the input to fully connected layers, $j = 1, 2, \ldots, q$, and the output of each fully connected layer is 1-dimensional, which is expressed as follows:

$$g\left(v(x_i)_j\right) = \mathbf{W}_j v(x_i)_j, \tag{1}$$

where $\mathbf{W}_j$ is the weight matrix of the j-th subblock, the output of each subblock is the input of the active layer, and the sigmoid function is chosen as the activation function, which is denoted as follows:

$$f_j(x) = \text{Sigmoid}\left(u^{(j)}\right) = \frac{1}{1+e^{u^{(j)}}}, \tag{2}$$

where $u^{(j)} = g(v(x_i)_j)$. After the eigenvalues are converted into the eigenvector, the relaxation of the binary vector is

obtained. To improve the performance, we do not directly map the image into binary values of {0, 1}. Instead, we use the activation function to limit the eigenvalues among [0, 1] and then use the thresholding to quantize the relaxation binary into binary code.

3.4. Triplet Loss and Optimization. We propose an improved triplet loss function to optimize the network to effectively preserve semantic similarities of images into the binary hash codes.

Let x be an image, the input to the proposed deep architecture is triplets of sample images, that is, $\{x_a, x_p, x_n\}$. $s_{ap} = 1$ and $s_{an} = 0$, where S denotes the similar identity of the images; the optimization of this triplet loss function is to narrow the distance between samples x_a and x_p and to push away the distance between samples x_a and x_n. We use $\|f(x_a) - f(x_p)\|_2$ and $\|f(x_a) - f(x_n)\|_2$ to represent the Euclidean distance between them and for the relaxed binary code obtained from the samples. As Euclidean distance can approximately represent their Hamming distance, the optimization goal is $\|f(x_a) - f(x_p)\|_2 + \sigma < \|f(x_a) - f(x_n)\|_2$. In this way, the objective function can be defined as

$$L = \sum_{\substack{S_{ap}=1 \\ S_{an}=0}} \max\left\{\sigma + \left(\left\|f(x_a) - f\left(x_p\right)\right\|_2 - \|f(x_a) - f(x_n)\|_2\right), 0\right\}. \tag{3}$$

Because the Euclidean distance is more stable in the training process and the meaning of the function is more consistent with the actual definition [42], we use Euclidean distance $\|\cdot\|_2$ to measure the distance in Hamming space rather than the square of Euclidean distance $\|\cdot\|_2^2$, which is used in the classical triplet loss function. The optimization aims to distinguish between similar samples and the different samples at least margin, which can map semantically equivalent pictures to adjacent locations in the Hamming space. Thus, the semantic features of the images extracted from CNN can be preserved in the hash code.

The basic rule of designing the loss function is to preserve the similarity order, that is, minimize the gap between the approximate nearest neighbor search result computed from the hash codes and the ideal search result obtained from the input space. A widely used solution is to select sample pairs in which the distance between X_a and X_p is greater than the distance between X_a and X_n, in a minibatch. In this work, we choose the hardest positive and negative sample pairs to compute the loss. The function is defined as follows:

$$L = \sum_{i=1}^{P}\sum_{a=1}^{K} \max\left\{\sigma + \max_{p=1\ldots K}\left(\left\|f\left(x_a^i\right) - f\left(x_p^i\right)\right\|_2\right) - \min_{\substack{j=1,\ldots,P \\ n=1,\ldots,K \\ j!=i}}\left(\left\|f\left(x_a^i\right) - f\left(x_n^j\right)\right\|_2\right), 0\right\}, \tag{4}$$

where P stands for the categories in the batch, K stands for the number of images in the category, x_a^i means the ath picture in the ith class, and σ is the margin parameter.

For fast convergence, it is sensitive to the selection of triplets. Here, we use large mini-batches and only compute the hardest positive and negative samples within a minibatch instead of selecting the hardest triplets in all training data. Furthermore, these functions are differentiable almost everywhere, which means they can be used in models trained by stochastic gradient descent. On the other hand, implementation details make batches of 20–30 exemplars more efficient.

Moreover, by minimizing equation (4), the manual margin parameter σ is designed to enforce a margin between the hard positive and hard negative pairs. Therefore, we optimize the parameter through the training process with the initial value of 0.2, and implementation details make margin parameters of 0.1 to 0.8 of exemplars more efficient. How to automatically determine the margin and incorporate class-specific or sample-specific margins remains challenging.

3.5. Generate Hash Code. When the network training is completed, the given image will get a K-bit hash code. We define $\operatorname{sgn}(x)$ as a symbolic function for each component.

$$\operatorname{sgn}(x) = \begin{cases} 1, & x \geq 0.5, \\ 0, & x < 0.5. \end{cases} \tag{5}$$

If the eigenvector of image x_i extracted from the network merging layer is x_i, then the hash code of this image x_i can be described as $B_i = \operatorname{sgn}(x_i)$. We can compute all images in the database to build a binary index library. We can use the hash code to do the nearest-neighbor retrieval in the Hamming space during the retrieval process, which is very efficient because the Hamming distance can be calculated using XOR.

The main steps of the proposed method are summarized in Algorithm 1.

4. Experiments

In this section, we conduct extensive experiments on two tourist image datasets to evaluate the efficiency and

Input: $\Theta \in R^{d \times n}$, the training image matrix
q, the hash code length
j, number of sub-layers
$\mathbf{W}$, the weight matrix
Output: deep hash functions $h(x)$
(1) Initialize the deep models by the pre-trained VGG-16 Sub-Network
(2) Update $\mathbf{W}$ in training process according to loss function;
(3) For $x \in \Theta$ do
(4) For iter = 1 to j do
(5) Compute $g(v(x_i)_{\text{iter}})$;
(6) Compute $f_j(x)$;
(7) Quantize the relaxation binary into binary code with $f_j(x)$;
(8) Return $h(x_i) = \text{sgn}(x_i)$
(9) End for
(10) End for

Algorithm 1: Key steps of the approach.

effectiveness of the proposed method. The details of the experiments and the results are described in the following sections.

4.1. Datasets and Experimental Settings

4.1.1. Datasets

(1) China-60 Dataset. Most public landmarks such as Oxford5K and Paris6K present unrelated images suitable for classification frameworks. However, images representing views of the same scene are needed. Thus, we developed a dataset called China-60, randomly selected from Flickr and Baidu Images based on the keywords of 60 popular tourist attractions in China. Variability of images comes from different viewing scales, angles, lighting conditions, and image clutter. Therefore, we provide 3–5 tags to describe the image contents, such as name and places. Our research's primary purpose is tourism image retrieval, so we have developed a Chinese image dataset with attraction to verify the method's performance on the image retrieval task.

For each tourist attraction, we crawl 500 to 600 images and remove irrelevant or low-quality photos. The final dataset contains 25,890 images of 60 tourist attractions, including buildings, rivers, forests, mountains, and other types of interests, all photographed under different light, seasons, and angles. We divide the dataset into the training set, test set, and validation set in a ratio of 8 : 1 : 1. In evaluation, the images belonging to the same tourist attraction are considered similar. On the contrary, they are deemed dissimilar. Typical images are shown in Figure 2.

(2) Public Datasets. For a clear comparison and analysis, we also experiment on the different datasets Cifar-10 and Flickr30k. Cifar-10 contains 60,000 images, which are divided into ten categories, each containing 1,000 images. All photos have a 32×32 resolution. We also divide them into the training set, validation set, and test set according to the proportion of 8 : 1 : 1. Flickr30k contains 31,783 images focusing mainly on people and animals. We select 1000 outdoor images randomly for the testing set and 30783 other for the training set.

4.1.2. Baseline and Evaluating Indicators

(1) Baseline. To illustrate the benefits of the proposed method, we compare it with various approaches, including existing traditional hash approaches LSH [27], SH [31], PCAH [53], PCA-ITQ, PCA-RR [28], CBR-rand, CBR-opt [54], and DSH [55]. We also compare it with deep hashing approaches, such as DLBHC [38] and DNNH [9]. Finally, after fine-tuning, the features are extracted from the pre-trained VGG network as the mapping function input instead of handcraft features.

(2) Evaluating Indicators. Four evaluation indicators were used to assess the performance of the different methods as follows: (1) precision at N sample curve, where precision is the proportion of the correct samples in the returned images, (2) recall at N samples curve, where recall is the proportion of the accurate results in the query results to all correct results, (3) precision-recall (P-R) curve which is the curve of precision changing with recall, and (4) mean average precision (MAP), which is the area surrounded by the P-R curve.

4.2. Results and Discussion on China-60. We first evaluate the effectiveness by comparing each method's performance under different lengths of hash code, which can get a convincing result. Firstly, we assess the performance in terms of MAP, calculated for all returned samples by sorting with the Hamming distance. The MAP value is shown in Table 2, where DNNH, DLBHC, and the proposed method are deep hashing methods, while the other ways are traditional hashing methods. As shown in Table 2, the proposed method's results perform better than other methods, and the MAP values of most practices have a positive correlation with the length of the hash code. The experiments show that

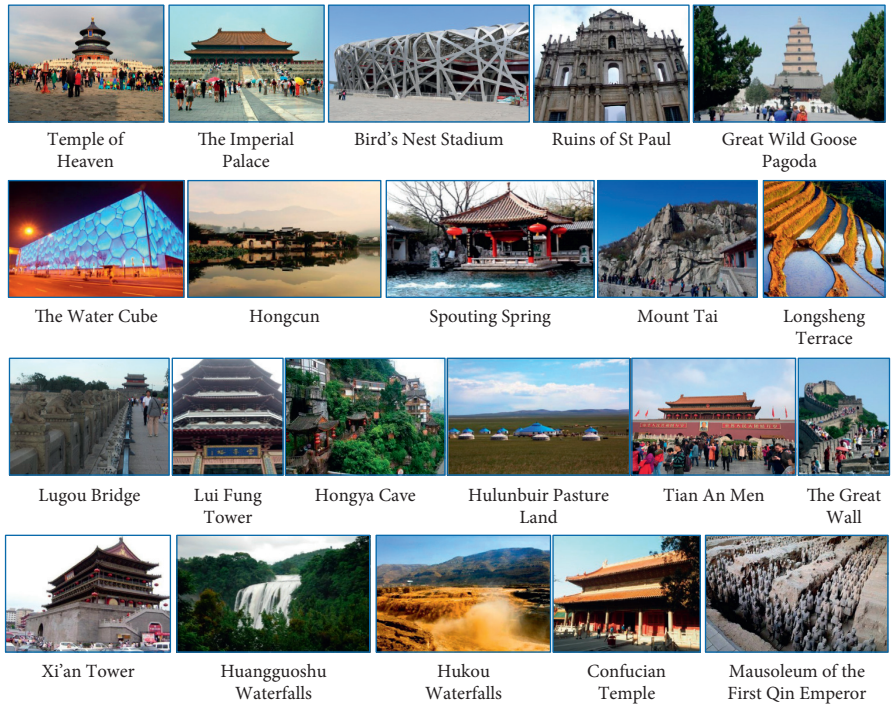


Figure 2: Sample images of China-60.

Table 2: The value of MAP for different methods on the China-60 dataset.

Method	China-60			
	32 bits	64 bits	128 bits	256 bits
CBE-rand	0.329	0.473	0.618	0.681
CBE-opt	0.338	0.496	0.628	0.694
ITQ	0.681	0.794	0.804	0.813
LSH	0.314	0.483	0.597	0.691
PCAH	0.515	0.614	0.415	0.278
SH	0.102	0.234	0.302	0.293
PCA-RR	0.517	0.652	0.694	0.728
DSH	0.238	0.283	0.397	0.465
DLBHC	0.814	0.841	0.856	0.849
DNNH	0.839	0.864	0.860	0.862
Ours	0.895	0.907	0.912	0.903

traditional hashing methods and the size of the binary feature are often highly correlated.

Figure 3 shows the precision-recall (P-R) curves for different methods on the Cifar-10 dataset. We plotted P-R curves on the hash code of four diverse lengths. It can be seen from the diagram that our approach can always maintain the highest precision rate and smaller curve slope under all-length hash code when the recall rate is low. This means that our policy has better retrieval performance. We can also find the gap between the deep hashing algorithm and the traditional algorithm in the graph. Most traditional hashing algorithms have a concave curve on the short hash code, signifying that they have terrible performance on the short hash code. However, with the increase of the length of hash code, part of the P-R curves of traditional hashing algorithms become convex curves, which signify that an extended hash code is often required to ensure the retrieval of conventional hashing effect algorithms. This is consistent with what we said before. On the other hand, the deep hashing algorithms have a slight variation in curve radian under different lengths of hash code, showing the stability and superiority of the deep hashing algorithms.

The TOP-K accuracy rate reflects the proportion of the first K returned results from the correct results of the query, which the user can intuitively perceive in the retrieval results. Therefore, the TOP-K precision rate is an important index to evaluate the retrieval algorithm's practical application performance. Figure 4 shows the precision of TOP-K retrieval results in the nearest neighbor retrieval. Similarly, the plot shows the precision curves of 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d) lengths of hash code, respectively. The horizontal coordinate of the curve is the number of returned samples, and the vertical coordinate is the precision rate. It can be seen from the diagram that the

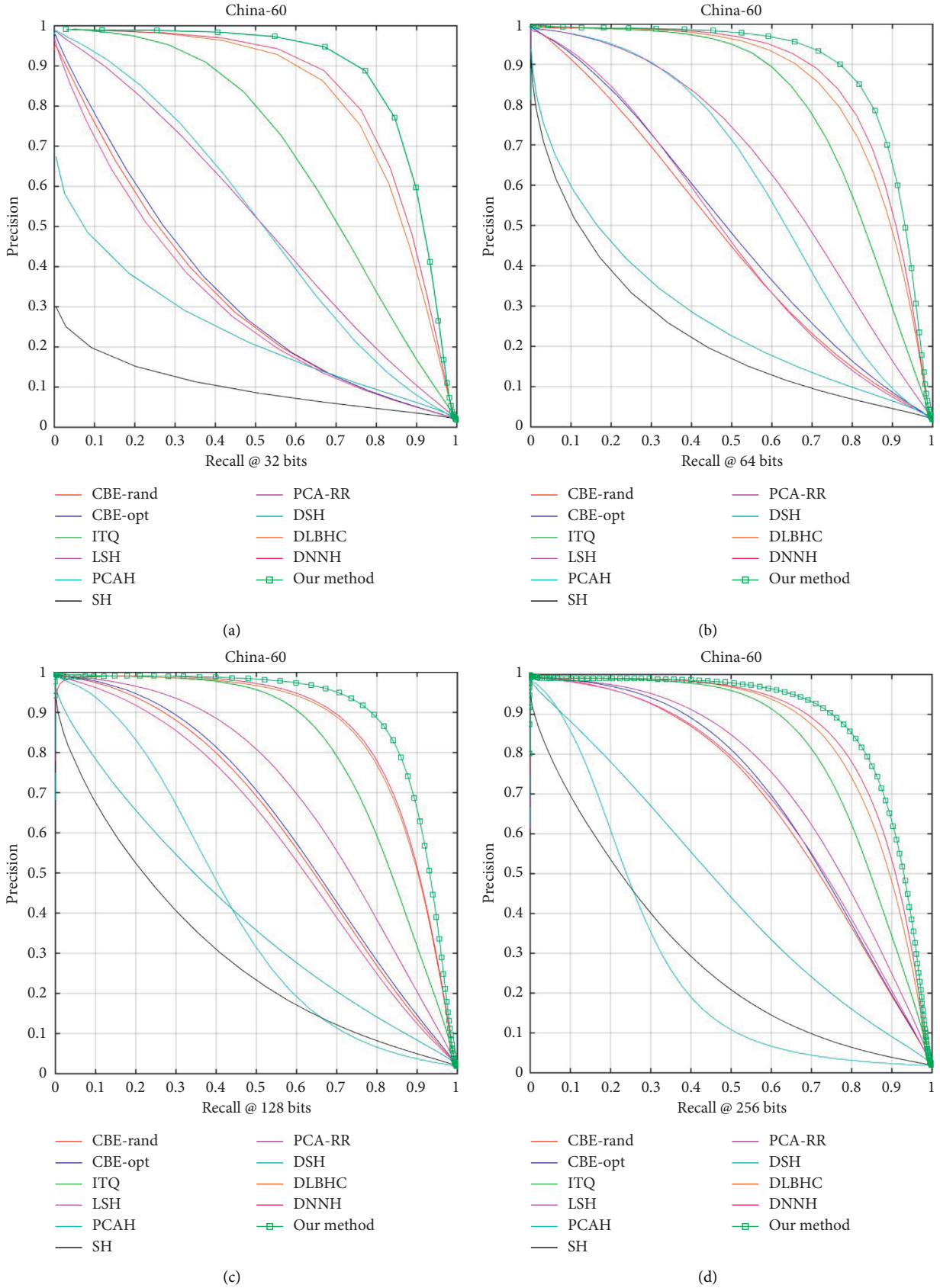


Figure 3: The precision versus recall curves. The length of hash code is 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d).

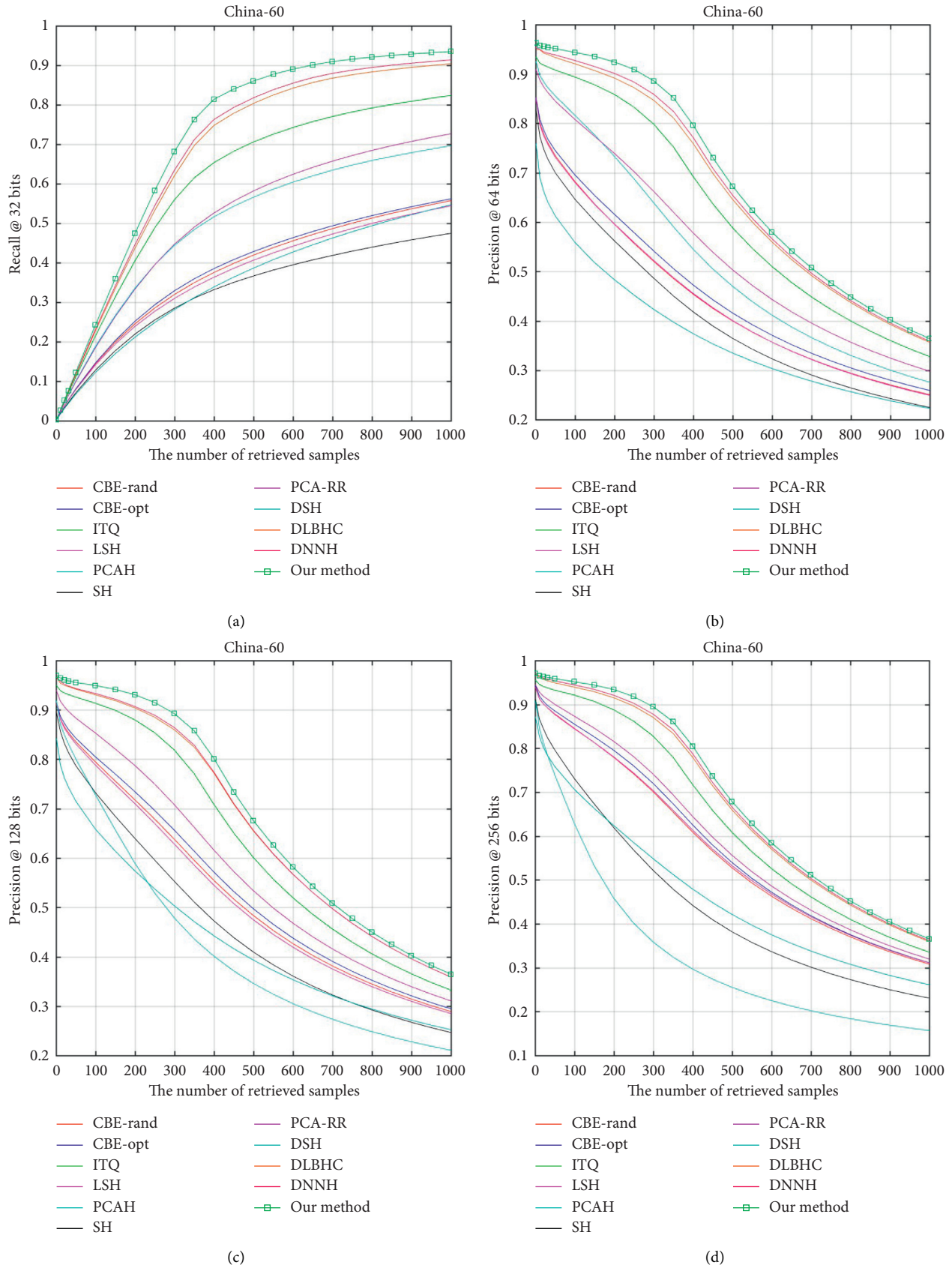


FIGURE 4: The precision versus the variable number of sample curves. The length of hash code is 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d).

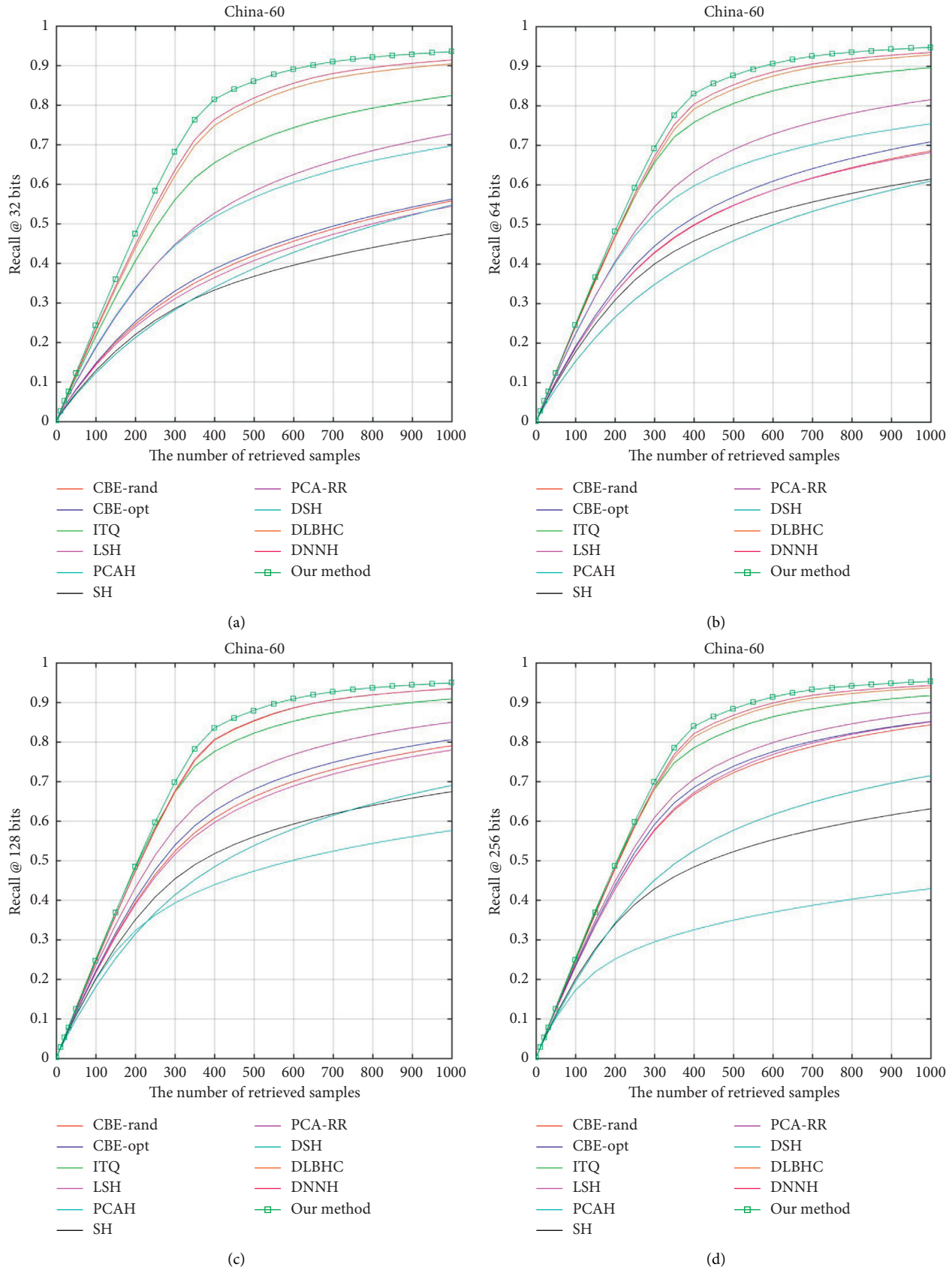


Figure 5: Recall versus variable number of sample curves. The hash code is 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d).

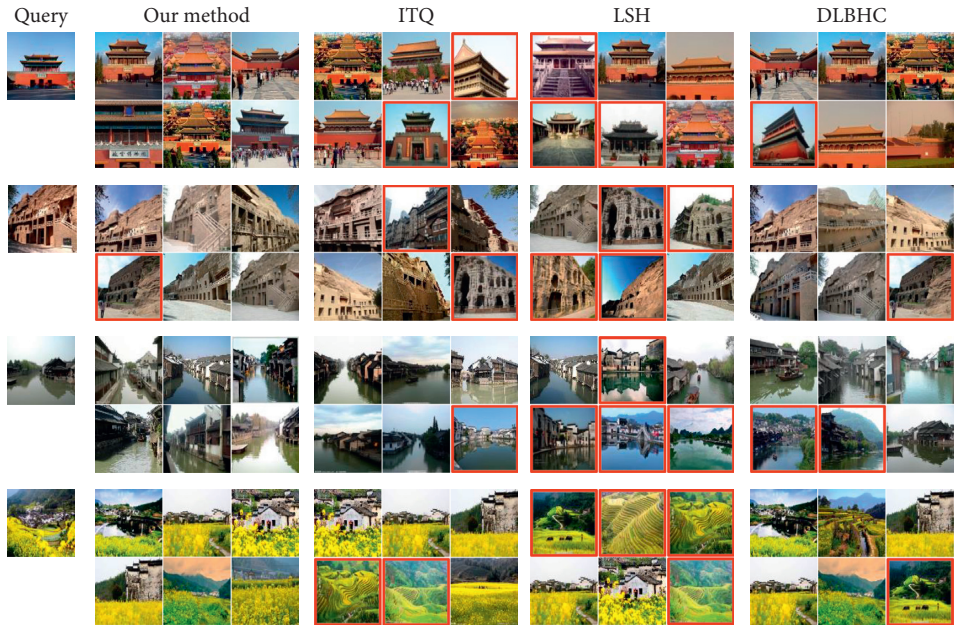


Figure 6: On the China-60 dataset, the top 6 results of 4 query images are returned by different hashing methods' retrieval. The pictures of the first column are queries, and we use 128-bit hash codes for retrieval. The image in the red line is considered not similar to the query image.

retrieval precision of our approach is always the best of all cases, and when fewer samples are returned, the retrieval precision can reach the highest value. This reflects that the correct samples can be usually returned preferentially, which makes our method sufficiently meet the requirements of image recognition and retrieval for unknown scenic spot images.

Figure 5 shows the TOP-K relation curve between the recall rate and returned sample. The horizontal coordinate is the number in returned samples, and the ordinate is the recall rate of the sample. The correct sample in the returned sample accounts for all of the correct samples in the database. This is an essential criterion of evaluation that developers and administrators of the retrieval system concern about. In addition, it reflects the retrieval success degree of the algorithm in the database. As shown in the figure, our method achieves the best TOP-K recall for all coding lengths. Figure 6 exhibits some query examples on the China-60 dataset. For each query, each method returns the top 6 query results by using the 128-bit hash code, and red represents the incorrect returned results.

4.3. Generalization to Other Image Data Sources. Although the primary purpose of this article is to explore the effect of retrieval methods on image retrieval tasks in tourist attractions, for demonstrating the universality of the process, we also conducted experiments in public image datasets. Considering that the size of the Cifar-10 dataset image is 32×32, we shorted the generated hash code length to 12 bits, 24 bits, 32 bits, and 48 bits. Thus, the hash code length is also consistent with the Flickr30 dataset.

Table 3 shows the results of MAP values on the two datasets, where CNNH, DNNH, DLBHC, and the proposed method are deep hashing methods, and the others are the no-deep methods. It can be seen from the results that our approach has a significant advantage over the no-deep hashing algorithm. The MAP value of most no-deep methods dramatically increases with the length of the hash code. In the best case, compared with the best no-deep hashing method, the deep hashing algorithm still has a significant superiority. For the deep hashing approach, the accuracy of our process has a 4% to 8% enhancement, which shows that the hash code generation strategy proposed in this paper can efficiently improve the retrieval effect.

4.4. Generalization to Cross-Datasets. To verify our method in general, we conduct experiments over the cross-datasets. The aim is to utilize two or more datasets labeled with different classes to train and evaluate a single model. For example, we train the proposed model by various datasets: the Flickr30 dataset and the Cifar-10 dataset, respectively. The performance of the trained model is tested by taking a different dataset, China-60.

The experimental results are shown in Table 4, which shows that the overall precision scores are relatively low, indicating that cross-datasets evaluation is more challenging for the retrieval task. However, it also demonstrates that the proposed method achieves the competitive performance on the cross-datasets tourist images retrieval task, demonstrating the effectiveness of our proposed method.

4.5. Time-Cost Analysis. Besides the effectiveness analysis, we also compare the proposed approach with other methods, deep and no-deep, in terms of the computation time cost. All the experiments are carried out on the same platform with Intel i7 8700K CPU, NVIDIA GTX 2080, and 64G RAM. Table 5 shows the average computation times of different

Table 3: The value of MAP for different methods on the public datasets.

Method	Cifar-10				Flickr30			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
ITQ	0.264	0.282	0.288	0.295	0.577	0.580	0.581	0.580
LSH	0.183	0.164	0.161	0.162	0.557	0.564	0.562	0.569
PCAH	0.157	0.164	0.162	0.170	0.588	0.596	0.604	0.601
SH	0.183	0.164	0.161	0.161	0.561	0.562	0.563	0.562
DSH	0.303	0.337	0.346	0.356	0.678	0.697	0.689	0.685
CNNH	0.439	0.511	0.509	0.522	0.667	0.688	0.654	0.626
DLBHC	0.553	0.580	0.578	0.557	0.692	0.710	0.703	0.707
DNNH	0.571	0.588	0.589	0.595	0.739	0.752	0.753	0.755
Ours	0.613	0.648	0.654	0.663	0.828	0.837	0.835	0.840

Table 4: The value of MAP for cross-datasets evaluation.

Method	Cifar-10				Flickr30			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
ITQ	0.027	0.032	0.035	0.034	0.040	0.051	0.066	0.070
LSH	0.028	0.034	0.041	0.042	0.051	0.064	0.077	0.089
PCAH	0.038	0.045	0.060	0.071	0.057	0.069	0.072	0.078
SH	0.019	0.024	0.027	0.033	0.017	0.018	0.019	0.022
DSH	0.058	0.070	0.073	0.078	0.069	0.092	0.104	0.109
CNNH	0.087	0.092	0.089	0.090	0.095	0.105	0.101	0.104
DLBHC	0.095	0.101	0.112	0.115	0.102	0.107	0.105	0.119
DNNH	0.089	0.105	0.104	0.107	0.107	0.105	0.112	0.116
Ours	0.105	0.113	0.112	0.115	0.134	0.142	0.147	0.145

Table 5: Comparison of the average computation time (per image) in different methods.

Method	China-60 (ms)	Cifar-10 (ms)	Flickr30 (ms)
ITQ	7.95	5.42	4.10
SDH	8.05	5.51	4.15
CNNH	6.85	4.62	3.51
DLBHC	6.95	4.65	3.53
DNNH	6.72	4.63	3.47
Ours	6.52	4.45	3.40

methods. The proposed approach is comparable with other methods.

5. Conclusion

In this paper, we proposed a deep hashing method with scalable interblock for large-scale tourist attractions. After end-to-end training of the constructed deep hash network, the network utilizes the triplet loss function to guarantee the hash code's characteristic similarity. To enhance the performance and efficiency of function optimization and the descriptive ability of hash code, we improve the network and triplet loss function. Based on the results, we report the quantitative evaluation of the proposed method to scale hash length. Experimental results on social image datasets validate the superiority of the proposed method. However, the relaxed binary code obtained from the network may cause feature loss in the threshold process. In future work, we will improve the activation function to dispose of these problems.

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 no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (41971365) and the Chongqing Research Program of Basic Science and Frontier Technology (cstc2019jcyj-msxmX0131).

References

[1] K. Yan Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision*

and Pattern Recognition, 2004. CVPR 2004, Washington D. C., USA, June 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, CA, USA, June 2005.

[3] J. Yang, Y. G. Jiang, A. G. Hauptmann, and C. W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 197–206, Augsburg, Bavaria, Germany, September 2007.

[4] J. Wang, T. Zhang, J. song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 769–790, 2018.

[5] J. Sanchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proceedings of the The 24th IEEE Conference on Computer Vision and Pattern Recognition Cvpr 2011*, pp. 1665–1672, CO, USA, June 2011.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.

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

[8] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," *Proceedings of the National Conference on Artificial Intelligence*, vol. 3pp. 2156–2162, Québec, Canada, July 2014.

[9] H. Lai, Y. Pan, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3270–3278, Boston, MA, USA, June 2015.

[10] K. Zhou, J. Zeng, Y. Liu, and F. Zou, "Deep sentiment hashing for text retrieval in social CIoT," *Future Generation Computer Systems*, vol. 86, pp. 362–371, 2018.

[11] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3893–3903, 2018.

[12] Q. Jiang, X. Cui, and W. Li, "Deep discrete supervised hashing," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5996–6009, 2018.

[13] X. Zhe, S. Chen, and H. Yan, "Deep class-wise hashing: semantics-preserving hashing via class-wise loss," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1681–1695, 2020.

[14] Q. Hao, R. Cai, Z. Li, L. Zhang, and F. Wu, "3D visual phrases for landmark recognition," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3594–3601, Providence, RI, USA, June 2012.

[15] X. Xiao, C. Xu, J. Wang, and M. Xu, "Enhanced 3-D modeling for landmark image classification," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1246–1258, 2012.

[16] R. Ji, L. Y. Duan, and J. Chen, "Location discriminative vocabulary coding for mobile landmark search," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 290–314, 2012.

[17] L. Y. Duan, J. Chen, R. Ji, T. Huang, and W. Gao, "Learning compact visual descriptors for low bit rate mobile landmark search," *AI Magazine*, vol. 34, no. 2, p. 67, 2013.

[18] W. Zhou, M. Yang, H. Li, X. Wang, Y. Lin, and Q. Tian, "Towards codebook-free: scalable cascaded hashing for mobile image search," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 601–611, 2014.

[19] L. Zhu, Z. Huang, X. Liu, X. He, J. Sun, and X. Zhou, "Discrete multimodal hashing with canonical views for robust mobile landmark search," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2066–2079, 2017.

[20] C. Jing, M. Dong, M. Du, Y. Zhu, and J. Fu, "Fine-grained spatiotemporal dynamics of inbound tourists based on geotagged photos: a case study in beijing, China," *IEEE Access*, vol. 8, Article ID 28735, 2020.

[21] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Transactions on Image Processing*, vol. 29, pp. 1271–1284, 2020.

[22] N. Mou, R. Yuan, T. Yang, H. Zhang, J. Tang, and T. Makkonen, "Exploring spatio-temporal changes of city inbound tourism flow: the case of Shanghai, China," *Tourism Management*, vol. 76, 2020.

[23] A. A. Chugunova, "Soft power digital capabilities in the tourist image construction of a big city (on the example of st. Petersburg)," in *Proceedings of the 2020 IEEE Communication Strategies in Digital Society Seminar (ComSDS)*, pp. 7–13, St. Petersburg, Russia, April 2020.

[24] N. D. Payntar, W. L. Hsiao, R. A. Covey, and K. Grauman, "Learning patterns of tourist movement and photography from geotagged photos at archaeological heritage sites in Cuzco, Peru," *Tourism Management*, vol. 82, Article ID 104165, 2020.

[25] S. Law, Y. Shen, and C. Seresinhe, "An application of convolutional neural network in street image classification: the case study of London," in *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pp. 5–9, Redondo Beach, CA, USA, November 2017.

[26] Y. Kang, N. Cho, J. Yoon, S. Park, and J. Kim, "Transfer learning of a deep learning model for exploring tourists' urban image using geotagged photos," *ISPRS International Journal of Geo-Information*, vol. 10, no. 3, p. 137, 2021.

[27] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pp. 253–262, ACM, NY, USA, June 2004.

[28] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[29] W. Liu, C. Mu, S. Kumar, and S. F. Chang, "Discrete graph hashing," *Advances in Neural Information Processing Systems*, vol. 4, pp. 3419–3427, 2014.

[30] Q. Y. Jiang and W. J. Li, "Scalable Graph Hashing with Feature Transformation," in *Proceedings of the 24th International Conference on Artificial Intelligence IJCAI*, pp. 2248–2254, Buenos Aires, Argentina, July 2015.

[31] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of the Advances in Neural Information Processing Systems, Twenty-Second Annual Conference on Neural Information Processing Systems*, pp. 1753–1760, Vancouver, British Columbia, Canada, December 2008.

[32] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 353–360, Bellevue, Washington, USA, June 2011.

[33] W. Liu, J. Wang, R. Ji, Y. G. Jiang, and S. F. Chang, "Supervised Hashing with Kernels," in *Proceedings of the 2012*

IEEE Conference on Computer Vision and Pattern Recognition, pp. 2074–2081, MI, USA, June 2012.

[34] J. Wang, W. Liu, A. X. Sun, and Y. G. Jiang, "Learning hash codes with listwise supervision," in *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV)*, pp. 3032–3039, IEEE, Sydney, NSW, Australia, December 2013.

[35] X. Li, G. Lin, C. Shen, A. Hengel, and A. Dick, "Learning Hash Functions Using Column Generation," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 142–150, Atlanta, GA, USA, June 2013.

[36] R. Xia, Y. Pan, H. Lai et al., "Supervised hashing for image retrieval via image representation learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2156–2162, Qubec, Canada, July 2014.

[37] F. Zhao, Y. Huang, and L. Wang, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1556–1564, MA, USA, June 2015.

[38] K. Lin, H. Yang, J. Hsiao, and C. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 27–35, MA, USA, June 2015.

[39] X. Yan, L. Zhang, and W.-J. Li, "Semi-supervised deep hashing with a bipartite graph," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3238–3244, AAAI Press, Palo Alto, CA, USA, Aug 2017.

[40] J. Zhang and Y. Peng, "SSDH: semi-supervised deep hashing for large scale image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 212–225, 2019.

[41] W. Shi, Y. Gong, B. Chen, and X. Hei, "Transductive semi-supervised deep hashing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.

[42] R. C. Tu, X. L. Mao, and W. Wei, "MLS3RDUH: deep unsupervised hashing via manifold based local semantic similarity structure reconstructing," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3466–3472, Yokohama, Japan, July 2020.

[43] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1618–1625, CA, USA, February 2017.

[44] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, "Shared predictive cross-modal deep quantization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5292–5303, 2018.

[45] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4032–4044, 2019.

[46] C. Yang, E. Yang and T. Liu, Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2189–2201, 2020.

[47] D. Xie, C. Deng, C. Li, X. Liu, and D. Tao, "Multi-task consistency-preserving adversarial hashing for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 29, pp. 3626–3637, 2020.

[48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: a 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.

[49] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2064–2072, HI, USA, July 2016.

[50] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep Hashing Network for Efficient Similarity Retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2415–2421, AZ, USA, February 2016.

[51] J. Lin, Z. Li, and J. Tang, "Discriminative deep hashing for scalable face image retrieval," in *Proceedings of International Joint Conference on Artificial Intelligence*, Melbourne, Australia, August 2017.

[52] Y. Zhai, X. Guo, Y. Lu, and H. Li, "In defense of the classification loss for person Re-identification," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1526–1535, CA, USA, June 2019.

[53] J. Wang, S. Kumar, and S. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3424–3431, CA, USA, June 2010.

[54] F. Yu, S. Kumar, Y. Gong, and S. F. Chang, "Circulant binary embedding," in *Proceedings of the 31st International Conference on Machine Learning, PMLR*, vol. 32, no. 2, pp. 946–954, Beijing, China, June 2014.

[55] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1362–1371, 2014.

Hindawi
Scientific Programming
Volume 2021, Article ID 5576978, 6 pages
https://doi.org/10.1155/2021/5576978

Research Article

Research Based on Multimodal Deep Feature Fusion for the Auxiliary Diagnosis Model of Infectious Respiratory Diseases

Jingyuan Zhao, Liyan Yu, and Zhuo Liu

The First Affiliated Hospital of Dalian Medical University, Dalian 116011, China

Correspondence should be addressed to Zhuo Liu; lzhuo0310@126.com

Received 22 January 2021; Accepted 30 April 2021; Published 10 May 2021

Academic Editor: Qingchen Zhang

Pulmonary infection is a common clinical respiratory tract infectious disease with a high incidence rate and a severe mortality rate as high as 30%–50%, which seriously threatens human life and health. Accurate and timely anti-infective treatment is the key to improving the cure rate. NGS technology provides a new, fast, and accurate method for pathogenic diagnosis, which can provide effective clues to the clinic, but determining the true pathogenic bacteria is a problem that needs to be solved urgently, and a comprehensive judgment must be made by the clinician combining the laboratory results, clinical information, and epidemiology. This paper intends to effectively collect and process the missing values of NGS data, clinical manifestations, laboratory test results, imaging test results, and other multimodal data of patients with infectious respiratory diseases. It also studies the deep feature fusion algorithm of multimodal data, couples the private and shared features of different modal data of infectious respiratory diseases, and digs into the hidden information of different modalities to obtain efficient and robust shared features that are conducive to auxiliary diagnosis. The establishment of an auxiliary diagnosis model for the infectious respiratory diseases can intelligentize and automate the diagnosis process of infectious respiratory, which has important significance and application value when applied to clinical practice.

1. Introduction

With the advent of the big data era, data has flooded all aspects of society. For modern medicine, the human body has become a big database, and various medical data make modern medicine show obvious data characteristics. Data, especially medical data, need to be reviewed dialectically. With the combination of big data analysis technology and biomedicine, various computational modeling methods (pattern recognition, data mining, machine learning, deep learning, etc.) have been applied to the field of medicine. Based on this, we design and establish research based on high-throughput pathogen detection system of the artificial intelligence high-performance computing platform of the sequencing platform which establishes a high-order tensor database for infectious respiratory diseases and a multimodal database that combines imaging, laboratory examination results and clinical manifestations, based on artificial intelligence for disease exploring and unifying the treatments of patients and establishing a treatment query system. This paper aims to study the combination of NGS data with clinical data and epidemiological data with the help of in-depth calculation models, in the diagnosis and treatment of infectious respiratory diseases' application.

Pulmonary infection is a common respiratory infectious disease in clinical practice with a high incidence. It ranks the first cause of death in countryside and the third in urban areas of China, especially severe pneumonia. It has an increasing trend in recent years, although the treatment methods have great progress than before, its fatality rate is still as high as 30% to 50%, which seriously threatens human life and health. The rapid and accurate diagnosis of pathogenic bacteria of respiratory tract infection is the key to treatment, which can help clinicians to optimize the use of antibacterial drugs in a timely manner, thereby speeding up recovery, increasing cure rate and improving prognosis. At present, the commonly used methods of microbial detection such as smear, culture, and polymerase chain reaction

cannot effectively meet the clinical needs. Genome analysis second-generation sequencing technology (mNGS, also known as high-throughput sequencing technology) provides a new, rapid, and accurate method for pathogen diagnosis. Compared with traditional pathogenic microorganism detection, mNGS has high sensitivity and large amount of information. It can detect pathogens early, guide the precise selection of antibacterial drugs, reduce the use of antibacterial drugs, reduce the mortality of patients, and can identify new/known pathogens infection and mixed infection.

2. Current Research Status

At present, the cause of 60% infectious diseases is still unclear [1]. Clinical metagenomics is a detection technology that uses high-throughput sequencing technology to clarify the classification and function of all microorganisms in a sample without relying on traditional microbial culture [2, 3]. This technology can simultaneously detect bacteria, fungi, viruses, and parasites in the same sample without any bias and does not require specific amplification. It is suitable for the investigation of infectious disease outbreaks of unknown pathogens and infection with negative results from traditional tests, immunodeficiency patients, and critically ill infected patients [4]. For special populations such as infants and young children, patients with advanced age or patients with underlying diseases, immunodeficiency populations, repeated hospitalizations, patients with repeated negative tests of traditional microbial detection techniques and poor treatment effects, patients with suspected infections of special pathogens, patients with unexplained infectious diseases, and patients with critical illness, it is necessary to identify pathogenic bacteria as soon as possible. On the one hand, due to the complexity of pathogenic microorganisms, traditional opportunistic pathogens may become the main pathogenic microorganisms; on the other hand, pathogenic microorganisms carry multiple antibiotic resistance genes [5]; in this case, clinical metagenomics is the best diagnostic option [4, 6–8].

3. Application of NGS in Detection of Pulmonary Pathogen Infection

The narrow clinical metagenomics technology mainly refers to the shotgun next generation sequencing technology. The main sequencing process is to break all the DNA in the sample into small fragments first and then build a library and sequence it on the computer. The informatics method splices the sequencing results and finally compares the database to clarify the detected species [9]. The broad clinical metagenomics technology also includes second-generation sequencing technology, which mainly includes sequencing technology of detection bacterial 16S ribosomal RNA and amplification subsequencing technology of detection of fungal internal transcribed spacer (ITS). The main sequencing process is to obtain all the DNA in the sample first, then use primers for specific bacteria or fungi to perform PCR amplification, build a database and sequence on the computer, use bioinformatics methods to obtain qualified sequencing data, and finally compare the database to clearly detect the species [10]. It is worth mentioning that clinical metagenomics can simultaneously identify bacteria, fungi, viruses, and protozoa in sample, can be accurate to the species level, and can also identify the antibiotic resistance of microorganisms and other functions, while amplicon sequencing technology can only identify bacteria or fungi in the sample that is accurate to the genus level, and the microbial related functions can only be inferred from the database [11].

Clinical metagenomics is considered to be the most powerful weapon to identify pathogens of infectious diseases [12], but there is no unified clinical application path yet. We combined the work characteristics of clinicians, laboratory technicians, and bioinformatics analysts in the clinical application of clinical metagenomics and summarized the application mode of clinical metagenomics in the precision diagnosis and treatment of respiratory infectious diseases. This model requires communication among clinicians, laboratory technicians, and bioinformatics analysts in order to obtain the most effective data and give precise medication.

The samples of patients with respiratory tract infection mainly include sputum, airway aspirate, and alveolar lavage fluid. Moran Losada et al. [13] used clinical metagenomics to detect induced sputum samples from patients with cystic pulmonary fibrosis at different ages and confirmed that 99% of respiratory tract microorganisms are hundreds of bacteria, mainly *Pseudomonas aeruginosa* and *Staphylococcus aureus* is predominant, while 10 types of fungi and viruses account for only about 1% of the respiratory tract microorganisms. The fungi are mainly *Candida* and *Aspergillus*, and the viruses are mainly adenovirus and herpes virus. The study also clarified that, in each respiratory sample, there is abundance of microorganism; in addition, the study confirmed the relevant antibiotic resistance genes of *Pseudomonas aeruginosa* and *Staphylococcus aureus*, which provide a basis for the precise selection of antibiotics. Langelier et al. [14] enrolled 22 bone marrow transplant patients admitted to hospital for lower respiratory tract infection and used clinical metagenomics to detect 250 μl of alveolar lavage fluid samples from each patient, and the results confirmed the existence of lungs in bone marrow transplant patients with acute respiratory infections HCOV229E, HRV-A, HHV-6, CMV, HSV, EBV, human papilloma virus, torque Tenuo virus, and other viruses, and there are also rare pathogenic bacteria: *Streptococcus mitis* (Streptococcus mitis) and *corynebacterium* (*Corynebacterium propinquum*), and the clinical symptoms of patients with coexistence of bacteria and viruses are significantly more severe. In addition, clinical metagenomics has also been used to clarify the characteristics of lung microbes in lung transplant patients secondary to lung infection [15].

4. Preprocessing of Multimodal Clinical Data of Infectious Respiratory Diseases

In view of the fact that there is no unified standard for the scope of data retrieval and database establishment of existing

infectious respiratory disease cases, through the retrospective data sorting and historical data follow-up, a large number of new infectious cases and the result are collected and tested. Derive complete high-throughput genomics data and clinical association data of pathogenic microorganisms, formulate data retrieval range, and summarize case data. Aiming at the problems of data missing and inaccurate data in aggregated multimodal data, the incomplete data-filling algorithm based on distributed subtraction clustering is studied. The incomplete data are clustered by an improved subtractive clustering algorithm, and then, the incomplete data is filled with the clustering result and weighted distance. Thereby, the data with missing attribute values can be filled in quickly and accurately, so as to prepare for subsequent tasks such as data mining and analysis:

(1) Collection and collation of case data of pulmonary infectious diseases formulate the definition, inclusion, and exclusion criteria of cases of infectious lung diseases. According to research needs, in accordance with the research plan approved by the unit's ethical approval and with the patient's informed consent, collect the case data of pathogenic microorganism genetic testing in our hospital's "National Gene Testing Application Demonstration Center" since 2018 and trace their outpatient or hospitalization information and relevant clinical data. Retrieve data through the hospital's HIS system, LIS system, and PACS system and formulate the scope of data retrieval including the name of the medical institution, unique ID number, date of onset or medical consultation, basic personal information (gender/date of birth/occupation, etc.), medical treatment department, main symptoms and signs, past history, chief complaint, main diagnosis, imaging examination, and laboratory examination (blood routine, CRP, pct, d-dimer, interleukin-6, G/GM test, Aspergillus antibody, new type Coccus capsular antibody, tuberculosis antibody, etc.). Download the diagnosis and treatment information according to the established information catalog to form the original csv database. The case data information of the target case is screened according to the researched infectious case definition and inclusion and exclusion criteria, and the infectious case data statistical table is formed. Finally, the formed data statistical table is summarized.

(2) Data filling of cases of lung infectious diseases.

Firstly, it studies the optimization of subtractive clustering algorithm by using the similarity measurement method of incomplete data and the idea of matrix multiplication and realizes the direct clustering of incomplete datasets based on the distributed subtractive clustering algorithm of multilevel MapReduce. The main time to execute the algorithm is spent on dividing the dataset S, calculating the Euclidean distance between sample points and calculating the density index of sample points. In order to reduce the time cost of the algorithm and improve the efficiency of the algorithm, for these three steps, a multilevel MapReduce process is used for distributed parallel computing. In order to make the division of the dataset S suitable for the MapReduce calculation model, the data to be processed is first stored in the form of rows so that it can be sliced by rows, and the data between slices has no correlation. In the process of subtractive clustering, the calculation of the neighborhood radius and the density of sample points need to use the distance between samples, so it is particularly important to generate the distance matrix between sample points. In order to make the data subset C suitable for MapReduce calculation model processing and then generate the distance matrix, this project uses two copies of the data subset C as the calculation matrix to perform the MapReduce implementation of matrix multiplication. In the process of using subtractive clustering to cluster the complete data subset C, it is necessary to calculate and modify the density index. It can be known from the density index formula that all the values in the ith row of the distance matrix G correspond exactly to the elements of the density index of the data object i. This feature ensures that the correction calculation of the density index is suitable for MapReduce parallel design.

After clustering incomplete data, the method of filling the missing data by studying the distance weighting coefficient between the data objects and data points in the same class is used to avoid the interference of other objects on the filling value. The key of this method is to determine the weighting coefficient of each data object. In order to determine the weighting coefficient objectively and accurately, this article uses the following formula to calculate the distance between the data objects to provide the weighting coefficient:

$$\mathrm{Dis}\left(s_i, s_j\right)\frac{m}{m'}\sqrt{\sum_{k=1}^{m}\left(s_{ik}-s_{jk}\right)^2}, \quad s_{ik}\neq * \text{ and } s_{jk}\neq *, \tag{1}$$

where Dis (S_i, S_j) represents the distance between the data object S_i and S_j, m is the number of attributes of the data object, and m' is the number of the same attributes of the two data objects that is not missing. Finally, fill incomplete data based on clustering and weighted distance.

5. Deep Feature Fusion Learning Model Based on Multimodal Data of Infectious Respiratory Diseases

This paper studies the deep nonnegative correlation feature fusion algorithm of multimodal data. Through the co-learning of unsupervised related and unrelated features, the influence of modal private features is removed from multimodal shared features, and the shared space is more effective and robust and the shared space is more effective and robust of the multimodal data-related fusion features. Research the deep migration feature fusion algorithm of unbalanced multimodal data, coupling the modal deep network

and the modal semantic correlation model, and design a unified deep network architecture based on multilayer semantic matching.

(1) Unsupervised multimodal data deep nonnegative correlation feature fusion algorithm

Given a multimodal dataset $X = \{X^{(V)}\}_{v-1}^{V}$, it contains n data instances under V modes, $X^V \in R^{d_{v\times n}}$ which represents the feature matrix of n data instances under the vth mode, and each data instance is represented as a d_v-dimensional feature vector. First, the structured sparse projection matrices $U_I^{(v)}$ and $U_C^{(v)}$ are used to convert the feature matrix $X^{(V)}$ of each mode into a mode private feature matrix $V_I^{(v)}$ and a mode shared feature matrix V_C. Then, based on the regularization of the invariant graph and the sparse projection limit, the multimodal reconstruction error function is constructed, and the function variables are jointly optimized to minimize the reconstruction error through the shared feature coupling. Finally, the cluster analysis of the data is completed on the obtained multimodal-shared feature V_C.

(2) Deep migration feature fusion algorithm for unbalanced multimodal data Based on typical correlation analysis (CCA), this project intends to construct multilayer semantic correlation model of cross-modal data. Typical correlation analysis model can project different data domains to related feature representation subspace through effective matrix conversion. The correlation between data domains is the greatest. To implement the model, first, $[X^S, X^T]$ is encoded using source and target domain depth networks, respectively, to learn the hidden layer data feature representation corresponding to the source and target domain $H^{S(l)} = f(X^{S(l-1)})$ and $H^{S(l)} = f(X^{S(l-1)})$, where f is the nonlinear activation function of the deep learning network. Then, typical correlation analysis is carried out on the obtained domain hidden layer features $H^{S(l)}$ and $H^{T(l)}$. The maximum correlation coefficient matrix corresponding to the learning source and target domain $U^{S(l)}$ and $U^{T(l)}$:

$$\Gamma\left(U^{S(l)}, U^{T(l)}\right] = \frac{U^{S(l)^T}, \sum_{ST} U^{T(l)}}{\sqrt{U^{S(l)^T} \sum_{SS} U^{S(l)}} \sqrt{U^{T(l)^T} \sum_{TT} U^{T(l)}}}. \quad (2)$$

Match the features of the first layer to a more similar modal semantic space through the correlation coefficient matrix and then carry out the semantic correlation of the next layer. The coupled modal deep network is related to each layer of modal semantics, and a deep multimodal multilayer semantic matching model can be obtained, which is defined as minimizing the reconstruction error of the source and target deep learning network, while maximizing the correlation of the cross-domain deep network. The specific objective function is as follows:

$$\min J\left(R_{S,T}\right) = \frac{J_s\left(\theta^s\right) + J_T\left(\theta^T\right)}{\Gamma\left(U^S, U^T\right)}. \quad (3)$$

$J_s(\theta^s)$ and $J_T(\theta^T)$ are the reconstruction errors of the source and target depth networks, including cost functions and parameter regularization terms, respectively.

6. Establish a Whole-Process Auxiliary Diagnosis and Reasoning Model for Infectious Respiratory Diseases

This paper takes expert experience as the core, uses existing medical dictionaries, electronic medical records, various medical guidelines, expert consensus, and other basic data to construct a domain knowledge map, and realizes it through knowledge extraction and knowledge fusion technology. Combine the in-depth feature fusion learning results of the multimodal data of infectious respiratory diseases, based on the knowledge map, and refer to the overall diagnosis process of infectious respiratory diseases in the hospital at this stage, establishing the whole process assistance for the infectious respiratory diseases' diagnostic reasoning model.

6.1. Construction of Knowledge Map of Infectious Respiratory Diseases. The data sources used to construct the knowledge graph can be divided into the following types.

Structured data: structured data extraction is done through the data integrator. The data integrator is divided into three parts: data integration design tools, data integration conversion tools, and data read-write plug-ins. Data integration design tools are used to provide users with graphical design data integration logic functions, data integration conversion tools are used to convert user designs into data integration application codes, and data read-write plug-ins are used to provide data read-write functions for data integration applications.

Semistructured data: semistructured data is characterized by a certain implicit structure, but its structure changes greatly and lacks standardization. Two types of semistructured data, encyclopedia websites and industry vertical websites, can usually be used to construct knowledge graphs in vertical domains. These data are all HTML-based Web data, and the web page elements to be extracted can be located through their label symbols. The web page mainly consists of the entry card at the top, the free-form text in the middle part, and the entry label at the bottom. The label structure of the entry card and entry label is relatively fixed. It can extract the required entity name, entity description, entity attribute, and relationship with other entities from the entry card part. The type of entity can be obtained directly or indirectly from the entry label. The free-form text part in the middle needs to extract the required knowledge through a long- and short-term memory network (LSTM).

Unstructured data: unstructured text is processed, through the named entity recognition method, in which the entity and the category of the entity have been identified. Then, the semantic relationship between entities is extracted

from the text through the relationship extraction module. For this task, first, train a relationship classifier, through which it determines whether there is a certain predefined relationship between two entities in a piece of text. This is essentially a classification problem of sequence data, which is solved by using a relationship extraction method based on remote supervision.

6.2. Auxiliary Diagnosis and Reasoning Model for Infectious Respiratory Diseases. The auxiliary diagnosis and reasoning model of infectious respiratory diseases is based on the domain knowledge map. After the entities and relationships of the examples are embedded, the encoding part and the decoding part are designed, and finally, the infectious respiratory diseases are classified and predicted. The coding part first constructs a convolutional layer to process multimodal data, inserts an attention module to extract features of instance data, and then combines the deep feature fusion model studied in this project to explore the deep information of different modal data. The decoding part finally predicts the type of disease in the case to achieve the purpose of auxiliary diagnosis.

7. Discussion

In the era of big data, with the rapid development of multimedia technology and the richness of data description methods, multisource, heterogeneous, and other multimodal data are widely available [16, 17]. Multimodal data refers to data obtained through different fields or perspectives for the same description object. By using the complementation of information between modalities, more accurate data characteristics can be learned, and subsequent data prediction and decision-making tasks can be effectively supported [18–20]. Feature learning of multimodal data requires effective data fusion methods. However, in practical applications, multimodal data usually has low-quality characteristics such as inaccuracy, incompleteness, and imbalance: inaccuracy refers to the possibility in multimodal data. It will contain nonrelated information such as noise or irrelevant items; incompleteness means that part of the modal information or part of the attribute information of some data instances in the multimodal data is missing; imbalance means that there are more instances of some modal data. And, other modal data instances are relatively small, so it is necessary to use modalities containing more instances to assist modalities containing fewer instances for analysis and learning. The abovementioned characteristics pose great challenges for the design of multimodal data fusion methods.

Deep neural networks can effectively filter data noise and deep abstract features of learning data through multilayer nonlinear conversion and promote similar semantic fusion [21]. Therefore, this project extends the deep neural network to inaccurate, incomplete, and unbalanced multimodal data and studies the corresponding in-depth fusion algorithm of low-quality multimodal data. Through the multilayer correlation and matching of modal data, a cross-modal integration deep feature fusion model of coupled modal network and shared features is obtained.

Data Availability

The data used to support the findings of this study are currently under embargo, while the research findings are commercialized.

Conflicts of Interest

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

Authors' Contributions

The data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Jingyuan Zhao and Liyan Yu contributed equally as co-first authors.

Acknowledgments

This study was partially funded by the National Natural Science Foundation of China (81871725) and the Foundation of Education Department of Liaoning Province (LZ2020010).

References

[1] R. Schlaberg, C. Y. Chiu, S. Miller, G. W. Procop, and G. Weinstock, "Validation of metagenomic next-generation sequencing tests for universal pathogen detection," *Archives of Pathology & Laboratory Medicine*, vol. 141, no. 6, pp. 776–786, 2017.

[2] E. Ruppé, G. Greub, and J. Schrenzel, "Messages from the first international conference on clinical metagenomics (ICCMg)," *Microbes Infect*, vol. 19, no. 4-5, pp. 223–228, 2017.

[3] E. A. Franzosa, T. Hsu, A. Sirota-Madi et al., "Sequencing and beyond: integrating molecular "omics" for microbial community profiling," *Nature Reviews Microbiology*, vol. 13, no. 6, pp. 360–372, 2015.

[4] P. Parize, E. Muth, C. Richaud et al., "Untargeted next-generation sequencing-based first-line diagnosis of infection in immuno- compromised adults: a multicentre, blinded, prospective study," *Clin Microbiol Infect*, vol. 23, no. 8, pp. 574.e1–574.e6, 2017.

[5] F. Barbier, A. Andremont, M. Wolff, and L. Bouadma, "Hospital-acquired pneumonia and ventilator-associated pneumonia," *Current Opinion in Pulmonary Medicine*, vol. 19, no. 3, pp. 216–228, 2013.

[6] K. M. Pendleton, J. R. Erb-Downward, Y. Bao et al., "Rapid pathogen identification in bacterial pneumonia using real-time metagenomics," *American Journal of Respiratory and Critical Care Medicine*, vol. 196, no. 12, pp. 1610–1612, 2017.

[7] E. Ruppé, A. Cherkaoui, V. Lazarevic, S. Emonet, and J. Schrenzel, "Establishing genotype-to-phenotype relationships in bacteria causing hospital-acquired pneumonia: a prelude to the application of clinical metagenomics," *Antibiotics*, vol. 6, no. 4, pp. 30–45, 2017.

[8] S. Grumaz, P. Stevens, C. Grumaz et al., "Next-generation sequencing diagnostics of bacteremia in septic patients," *Genome Med*, vol. 8, no. 1, p. 73, 2016.

[9] C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata, "Shotgun metagenomics, from sampling to analysis," *Nature Biotechnology*, vol. 35, no. 9, pp. 833–844, 2017.
[10] E. L. Van Dijk, H. Auger, Y. Jaszczyszyn, and C. Thermes, "Ten years of next-generation sequencing technology," *Trends in Genetics*, vol. 30, no. 9, pp. 418–426, 2014.
[11] A. Zhernakova, A. Kurilshikov, M. J. Bonder et al., "Population-based metagenomics analysis reveals markers for gut microbiome composition and diversity," *Science*, vol. 352, no. 6285, pp. 565–569, 2016.
[12] P. J. Simner, S. Miller, and K. C. Carroll, "Understanding the promises and hurdles of metagenomic next-generation sequencing as a diagnostic tool for infectious diseases," *Clinical Infectious Diseases*, vol. 66, no. 5, pp. 778–788, 2018.
[13] P. Moran Losada, P. Chouvarine, M. Dorda et al., "The cystic fibrosis lower airways microbial metagenome," *ERJ Open Research*, vol. 2, no. 2, pp. 00096–02015, 2016.
[14] C. Langelier, M. S. Zinter, K. Kalantar et al., "Metagenomic sequencing detects respiratory pathogens in hematopoietic cellular transplant patients," *American Journal of Respiratory and Critical Care Medicine*, vol. 197, no. 4, pp. 524–528, 2018.
[15] D. W. Lewandowska, P. W. Schreiber, M. M. Schuurmans et al., "Metagenomic sequencing complements routine diagnostics in identifying viral pathogens in lung transplant recipients with unknown etiology of respiratory infection," *PLoS One*, vol. 12, no. 5, Article ID e0177340, 2017.
[16] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: a survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.
[17] Y. Zhao, L. Bo, X. Hua et al., "Preface to the special topic on multimedia big data processing and analysis," *Journal of Software*, vol. 29, no. 4, pp. 897–899, 2018.
[18] H. Lin, Y. Wang, Y. Jia et al., "Overview of knowledge fusion methods for network big data," *Chinese Journal of Computers*, vol. 40, no. 1, pp. 1–27, 2017.
[19] Q. Zhao and Z. Li, "Cross-modal social image clustering," *Chinese Journal of Computers*, vol. 41, no. 1, pp. 98–111, 2018.
[20] J. Hu and J. Pei, "Subspace multi-clustering: a review," *Knowledge and Information Systems*, vol. 56, no. 2, pp. 257–284, 2018.
[21] 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.

Hindawi
Scientific Programming
Volume 2021, Article ID 6668496, 7 pages
https://doi.org/10.1155/2021/6668496

Research Article

A Comparison of Analgesic Effect between Preoperative and Postoperative Transversus Abdominis Plane (TAP) Blocks for Different Durations of Laparoscopic Gynecological Surgery

Meiyu Wei, Ming Liu, Jie Liu, and Haitao Yang

Department of Anesthesia, The Second Hospital of Dalian Medical University, Dalian, China

Correspondence should be addressed to Jie Liu; liujaye@hotmail.com and Haitao Yang; yanghaitaodlry@163.com

Received 25 December 2020; Revised 18 March 2021; Accepted 1 April 2021; Published 13 April 2021

Academic Editor: Liang Zhao

Aim. This study aims to compare the postoperative analgesia between preoperative and postoperative ultrasound-guided transversus abdominis plane (TAP) blocks for different durations of laparoscopic gynecological surgery. *Methods*. A total of 120 patients, ASA I-III, 18–65 years of age, were divided randomly into 2 groups: preoperative TAP group (pre-TAP group) and postoperative TAP group (post-TAP group). Patients in the pre-TAP group ($n = 60$) and post-TAP group ($n = 60$) received bilateral TAP blocks of 0.375% ropivacaine, 40 mL, preoperatively and postoperatively, respectively. Duration of surgery, postoperative pain score, consumption of analgesics, and postoperative nausea and vomiting (PONV) during the first 24 h postoperatively were recorded. *Results*. For all the patients in the two groups, similar analgesia was obtained with no statistical difference. The same results were found in duration of surgery <180 min. Meanwhile, patients undergoing surgery >180 min in the post-TAP group obtained lower postoperative pain score, lower analgesics consumption, and higher satisfaction score than those in the pre-TAP group. *Conclusion*. Postoperative TAP block could offer better postoperative analgesia than preoperative TAP block for patients undergoing surgery >180 min. No difference was found in analgesia effect between preoperative TAP block and postoperative TAP block for patients undergoing surgery <180 min.

1. Introduction

There are several methods to offer postoperative analgesia for abdominal surgery, oral analgesics, patient-controlled intravenous analgesia (PCIA), patient-controlled epidural analgesia (PCEA), and regional nerve block [1–3].

PCEA plays a direct role in the near operative region and possesses a more immediate analgesic mechanism; thus, it could offer fast, clear, and accurate analgesic effect and reduce the use of opioids. PCEA seems to be an ideal method for postoperative pain control; it could be demonstrated to have a good postoperative analgesic effect in many common operations such as abdominal and gynecological surgeries [3, 4]. PCIA acts on the whole body through intravenous analgesics with relatively longer and more rapid analgesic effect with the PCA device. It has the advantages of simple operation and a wide range of drug uses, including the narcotic analgesics and nonsteroidal anti-inflammatory drugs. PCIA is applicable to pain, postoperative pain, wound pain, after-burn pain, and inflammatory pain [5, 6].

However, the side effects of PCEA and PCIA are also notable [4, 7]. PCEA might cause low blood pressure, pruritus, paresthesia, nausea, and vomiting, and PCIA could cause itching and respiratory depression due to the inevitable use of opioids [4, 7, 8]. At the same time, with the development of laparoscopic minimally invasive surgery, the surgical incision is reduced, and the postoperative pain is not as strong as open surgery [2, 9–11].

In view of the shortcomings of PCEA and PCIA, nerve block has the advantages of less trauma, less impact on systemic circulation, no inhibition of the respiratory center, and relief of nausea and vomiting [2, 9, 12, 13]. Transversus abdominis plane (TAP) block is a regional technique for analgesia of the anterolateral abdominal wall [14] and may

offer good analgesia on abdominal surgery, especially gynecologic surgery [15–18]. TAP block seems to be an interesting alternative in patients with, for example, severe obesity, where epidural or spinal anesthesia/analgesia is technically difficult and/or poses a risk [19–21].

Previous studies showed that it was inconsistent to determine the optimal time on TAP for patients undergoing surgeries. Some investigators recommended that TAP be performed before surgery [17, 20, 22], some preferred postoperative performance [15, 16, 23, 24], and others found that there was no difference between the two time points.

Till now, there have been no report to show the analgesic effect of TAP on patients undergoing surgeries of different duration. We aimed to compare analgesia between preoperative and postoperative TAP blocks for different duration of laparoscopic gynecological surgery.

2. Materials and Methods

This prospective, randomized, single-blind clinical trial was approved by the ethics committee of the Second Hospital of Dalian Medical University and clinical trials registration number is ChiCTR1900027881.

2.1. Patient Population. We assumed that the difference in consumption of postoperative analgesics between groups was 20%; thus, at least 58 patients should be recruited in each group. For convenience, we planned to recruit 60 patients in each group.

The inclusion criteria were planned as follows:

(1) ASA I-III

(2) 18–65 years of age

(3) Patients scheduled for laparoscopic gynecological surgery under general anesthesia in the Second Hospital of Dalian Medical University

The exclusion criteria were planned as follows:

(1) Patients with history of chronic pain therapy during the past half year

(2) Addiction (including opioids and benzodiazepines)

(3) Allergy to prescription medications

(4) Psychological disorders

(5) Pregnancy

(6) Any contraindication to TAP block

(7) Refusal of consent

2.2. Procedure. After signing the written consent, all the patients were allocated into 2 groups randomly, pre-TAP group and post-TAP group. Heart rate (HR), blood pressure (BP), saturation of oxygen (SpO_2), and bispectral index (BIS) were monitored and data collected every 5 minutes. All the patients received standard general anesthesia. Induction of general anesthesia was induced using 0.03 mg/kg Midazolam (Jiangsu Nhwa Pharmaceutical Co., Ltd., Xuzhou, China), 0.3 μg/kg Sufentanil (Yichang Humanwell Pharmaceutical Co., Ltd., Yichang, China), 0.3 mg/kg Etomidate (Jiangsu Nhwa Pharmaceutical Co., Ltd., Xuzhou, China), and 0.3 mg/kg Cisatracurium (Jiangsu Hengrui Medicine Co., Ltd., Jiangsu, China). Following intubation, maintenance of general anesthesia was total intravenous anesthesia including propofol (4–12 mg/kg/h), remifentanil, and dexmedetomidine. The dosage was determined by the anesthesiologist according to keeping BIS at the scope 50 ± 5, systolic blood pressure (SBP) was controlled within 20% of the base value, and the mean arterial blood pressure (MBP) was not lower than 65 mmHg. Cisatracurium was added at 0.05 mg/kg according to the requirements of surgeons. Crystals and colloidal liquid solutions are used for volume displacement, and all aspects of anesthesia management are performed by the anesthesiologist in accordance with current clinical practice. The tidal volume was set at 6–8 ml/kg, respiratory rate was set at 12 breaths/min, and end-expiratory partial pressure of carbon dioxide (CO_2) was maintained at 35–45 mmHg. Dizocine (10 mg) and Ramosetron 0.3 mg were given intravenously 30 minutes before the end of surgery. If HR was lower than 45 bpm or higher than 120 bpm, we intravenously injected atropine 0.5–1 mg and esmolol 0.5 mg/kg, respectively; if BP was lower than 80% of basic value or SBP was higher than 160 mmHg, we administered norepinephrine 8–12 ug/min and nicardipine 2–10 ug/kg/min, respectively.

After induction of anesthesia, patients in the pre-TAP group received USG bilateral TAP block with 0.375% ropivacaine (Beijing Taide Pharmaceutical Co., Ltd., Beijing, China), 20 mL each side, before incision and patients in the post-TAP group were given same medications after the end of surgery and before extubation. After extubation, patients were transferred to postanesthesia care unit (PACU). When patients reached the criteria of leaving PACU, they could be transferred to ward. At ward, patients would be asked for the pain score, which was visual analogue scale (VAS) at 0, 2, 6, 12, and 24 h postoperatively. Flurbiprofen axetil (Jiangsu Hengrui Medicine Co., Ltd., Jiangsu, China), 50 mg intravenously, used as postoperative analgesic should be given if VAS was more than or equal 4.

2.3. Outcome Measures. Vital signs including HR, RR, BP, SpO_2, and BIS value were recorded every five minutes from entrance to operating room to leaving the PACU. Consumption of opioids during surgery, consumption of postoperative analgesics, times of rescue, which means times of postoperative analgesics demanding, duration of surgery, pain score at 0, 2, 6, 12, and 24 h postoperatively, mean duration of first analgesic demanding after surgery, postoperative nausea and vomiting (PONV), and satisfaction scores of patients and surgeons were also recorded.

VAS of 0 indicated no pain. VAS of 10 meant an ultimate pain. The VAS of patients was measured by a researcher who did not know this study. Degree of PONV was measured with a categorical scoring system (none = 0; mild = 1; moderate = 2; severe = 3). Satisfaction score ranges from 0 to 10, and 0 means totally unsatisfactory, while 10 means totally satisfactory. Duration of surgery was regrouped to three

subgroups: subgroup S, in which duration of surgery was <90 mins; subgroup M, in which duration of surgery was 90–180 mins; and subgroup L, in which duration of surgery was >180 mins. Only the patients were blinded to the group assignment.

2.4. Statistical Analysis. GraphPad Prism version 5 (GraphPad Software, Inc.) was used for data analysis. Demographic data was analyzed by chi-square test and *t*-test. Haemodynamic data, pain score and consumption of analgesics and opioids, were analyzed by repeated-measures analysis of variance and post hoc pairwise comparison for different stages of anesthesia. $P < 0.05$ was considered to have a statistically significant difference.

3. Results

3.1. Subject Characteristics. We totally recruited 132 patients. 3 patients were deleted because they refused to cooperate postoperatively, 7 patients were excluded because they were changed from laparoscopic surgeries to open ones during the surgeries, and 2 patients were deleted because of being diagnosed as retroperitoneal tumor during surgeries and they received abdominal surgery instead of gynecological surgery. Thus, finally we recruited 60 patients in each group.

There was no significant difference in gender, height, weight, ASA status, and duration of surgery between the two groups (Table 1). There was also no significant difference in cases, height, weight, and ASA status among the three subgroups (Table 2).

3.2. Clinical Results. No difference was found in vital signs between pre-TAP group and post-TAP group; and no severe accident happened.

There was no significant difference in postoperative pain score between pre-TAP group and post-TAP group (Figure 1).

In subgroup L, VAS in the pre-TAP group was higher than that in the post-TAP group at 0, 2, and 6 hours postoperatively (2.0 ± 1.3 versus 0.5 ± 0.7, 2.5 ± 1.3 versus 1.0 ± 0.8, and 3.1 ± 1.5 versus 1.6 ± 1.3, respectively) with significant difference, $P < 0.01$ (Figure 2). No statistical difference in postoperative pain score was found in subgroup S and subgroup M between the pre-TAP group and post-TAP group.

Data in Table 3 show that, for all the patients recruited, there was no difference in duration of first rescue, times of rescue, and satisfaction scores of patients and surgeons between the pre-TAP group and post-TAP group. Consumption of opioids (remifentanil) in the pre-TAP group was significantly lower than that in the post-TAP group (269.7 ± 86.4 versus 324.6 ± 136.4, $P = 0.03$). Degree of PONV in the pre-TAP group was lower than that in the post-TAP group (0.5 ± 0.8 versus 0.8 ± 0.9, $P = 0.03$).

In the three subgroups, consumption of opioids and degree of PONV in the pre-TAP group were lower than those in the post-TAP group with statistical difference. For subgroup S and subgroup M, no statistical difference was found in duration of first rescue, times of rescue, and satisfaction scores of patients and surgeons between pre-TAP and post-TAP groups. Meanwhile, in subgroup L, duration of first rescue in the pre-TAP group was lower than that in the post-TAP group (3.4 ± 2.8 h versus 11.0 ± 1.8 h, $P = 0.01$). Times of rescue in the pre-TAP group was 1.0 ± 0.5, which was significantly lower than that in the post-TAP group (0.5 ± 0.5) with $P = 0.03$. Patients in the pre-TAP group gave higher satisfaction score compared to their counterparts in the post-TAP group (7.4 ± 0.9 versus 8.8 ± 1.0), and the difference was significant ($P = 0.04$) (see Table 4).

Table 1: Demographic data.

	Pre-TAP ($n = 60$)	Post-TAP ($n = 60$)	P
Age (Y)	44.6 ± 11.3	46.4 ± 10.4	0.93
ASA I	51	52	0.98
ASA II	9	8	0.98
BMI (kg/m^2)	24.5 ± 3.6	24.5 ± 3.6	0.87
Duration (min)	133.4 ± 68.2	132.5 ± 71.6	0.74

ASA: American Society of Anesthesiologists.

4. Discussion

This was the first study to compare postoperative analgesia effect between pre-TAP and post-TAP blocks for different duration of surgeries.

Previous studies showed that it was inconsistent to determine the optimal time on TAP for patients undergoing surgeries. Some investigators recommended that TAP be performed before surgery [17, 20, 22]. Mansouri et al. found that bilateral intrapleural block performed before cardiac surgery could get better analgesia than postoperative manipulation due to preemptive analgesia [25]. Niraj et al. obtained same results in patients undergoing open appendicectomy [26]. Some researchers concluded that the analgesic effect of TAP block performed postoperatively was prior to emergence from anesthesia [15, 16, 23, 24]. Mcdonnell et al. found that the sensory block produced by lidocaine 0.5% receded over 4 to 6 hours, which was supported by magnetic resonance imaging studies that showed a gradual reduction in contrast in the transversus abdominis plane over time. French et al. reported that general anesthesia with postoperative supplementary bilateral ultrasound-guided TAP blocks was chosen to reduce the requirements for postoperative opioids and the risk of postoperative respiratory depression [27, 28]. Meanwhile other clinicians like Fibla et al. found that blocking time did not seem to affect postoperative pain scores [29].

In our study, for all the patients in pre-TAP and post-TAP groups, no difference was found in postoperative pain score, which was similar to the results of previous studies [29, 30]. In subgroups, no difference was found in subgroup S and subgroup M between pre-TAP and post-TAP group. Meanwhile, in subgroup L, postoperative scores in pre-TAP groups were significantly lower than those in post-TAP group at 0, 2, and 6 hours postoperatively, and the duration

Table 2: Demographic data of subgroups.

	Group S: <90 min			Group M: 90–180 min			Group L: >180 min		
	Pre-TAP	Post-TAP	*P*	Pre-TAP	Post-TAP	*P*	Pre-TAP	Post-TAP	*P*
Case	22	21	1	19	20	1	19	19	1
Age (Y)	45.9 ± 11.4	45.6 ± 10.7	0.92	39.8 ± 11.5	49.6 ± 9.3	0.33	47.2 ± 10.2	44.2 ± 10.4	0.88
ASA I	18	18	1	16	17	0.95	17	17	1
ASA II	4	3	0.95	3	3	1	2	2	1
BMI (kg/m^2)	25.0 ± 3.8	23.6 ± 2.6	0.19	24.5 ± 3.9	25.4 ± 4.7	0.38	24.0 ± 3.1	24.6 ± 3.0	0.78
Duration (min)	54 ± 19	56 ± 18	0.55	126 ± 23	122 ± 25	0.66	221 ± 31	226 ± 40	0.13

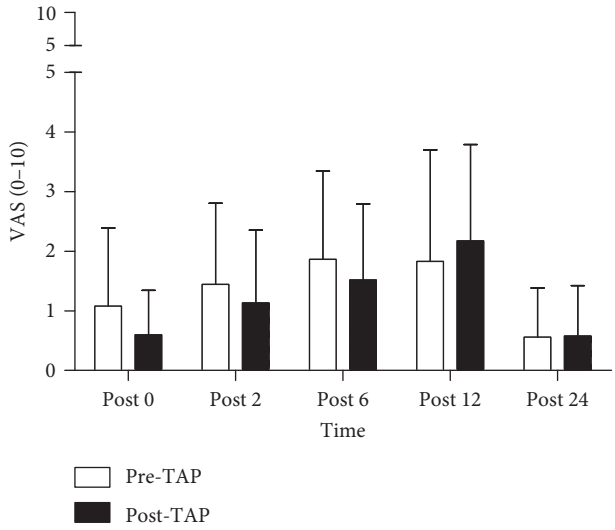


Figure 1: Postoperative pain scores of the pre-TAP group and post-TAP group. No significant difference was found.

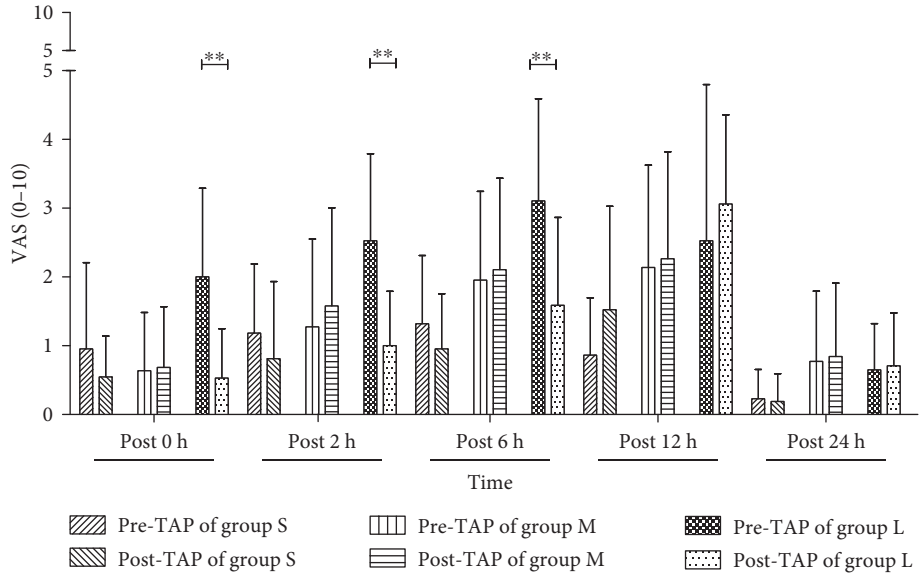


Figure 2: Postoperative pain score in subgroups. In group L, VAS in the pre-TAP group was higher than that in the post-TAP group at 0, 2, and 6 hours postoperatively. $^{**}P < 0.01$.

of first rescue was 4.4 hours postoperatively in pre-TAP group and 8.0 hours postoperatively in post-TAP group. The above results indicate that the analgesic effect of bilateral TAP of 0.375% ropivacaine began to fade at about 4 hours after manipulation and began to disappear at about 8 hours after manipulation. This block duration in our study also

Table 3: Clinical data of all the patients.

	Pre-TAP	Post-TAP	P
Duration of first rescue (h)	5.1 ± 5.5	7.2 ± 4.8	0.56
Times of rescue	0.5 ± 0.6	0.5 ± 0.6	0.90
Degree of PONV	0.5 ± 0.8	0.8 ± 0.9	0.03*
Remifentanil (ug/kg·min)	269.7 ± 86.4	324.6 ± 136.4	0.03*
Satisfaction score of patients	8.5 ± 1	7.8 ± 0.8	0.78
Satisfaction score of surgeons	9.1 ± 0.8	8.9 ± 0.9	0.88

*$P < 0.05$.

Table 4: Clinical data of subgroups.

	Subgroup S: <90 min			Subgroup M: 90–180 min			Subgroup L: >180 min		
	Pre-TAP	Post-TAP	P	Pre-TAP	Post-TAP	P	Pre-TAP	Post-TAP	P
Duration of first rescue (h)	6.5 ± 7.6	5.3 ± 5.3	0.23	5.2 ± 4.7	5.8 ± 4.2	0.57	4.4 ± 2.8	8.0 ± 1.8	0.01*
Times of rescue	0.4 ± 0.5	0.3 ± 0.5	0.33	0.5 ± 0.5	0.7 ± 0.7	0.32	1.0 ± 0.5	0.5 ± 0.5	0.03*
Degree of PONV	0.4 ± 0.8	0.6 ± 0.9	0.04*	0.5 ± 0.9	0.7 ± 0.7	0.04*	0.6 ± 0.8	1.3 ± 1.0	0.02*
Remifentanil (ug/kg·min)	189.0 ± 28.8	239.0 ± 36.9	0.04*	279 ± 66.5	313.8 ± 163.2	0.04*	338.9 ± 95.5	404.3 ± 116.8	0.02*
Satisfaction score of patients	9.1 ± 0.8	9.0 ± 0.9	0.87	8.9 ± 1.2	8.8 ± 1.0	0.78	7.4 ± 0.9	8.8 ± 1.0	0.04*
Satisfaction score of surgeons	8.8 ± 1.0	8.8 ± 0.9	0.76	8.9 ± 0.8	8.9 ± 0.9	0.82	8.6 ± 1.0	8.8 ± 0.9	0.23

*$P < 0.05$.

explains why no difference was found in subgroup S and subgroup M between pre-TAP and post-TAP groups. The durations of surgery in subgroup S and subgroup M were <2 hours, which indicates that the analgesic effect of bilateral TAP of 0.375% ropivacaine had not begun to fade yet. But this block duration was shorter than that in Stoving et al.'s study, approximately 10 hours with large variation [31]. Although the concentration was the same in these two studies, difference of block duration might be due to the different pharmaceutical factory. From the above results, we might conclude that it was necessary to decide the time of TAP block according to the duration of surgery.

In our study, the consumption of opioids and dosage of remifentanil in pre-TAP group were significantly lower than those in post-TAP group (269.7 ± 86.4 versus 324.6 ± 136.4 ug/kg.min, $P = 0.03$). The same results happened in all the subgroups. In subgroup S, subgroup M, and subgroup L, patients in pre-TAP group consumed less remifentanil compared to their counterparts in post-TAP group with statistical difference. This result showed that pre-TAP block could offer better analgesia than postoperative manipulation due to preemptive analgesia, which leads to the lower intraoperative consumption of opioids. This result was also consistent with studies from Mansouri [25] and Niraj et al. [26]. However, in subgroup L, patients in pre-TAP group consumed less postoperative analgesics compared to their counterparts in post-TAP group. This difference might be due to the fact that the block duration of bilateral TAP of 0.375% ropivacaine was 4–8 hours after manipulation. The lower pain score and lower consumption of rescue could be the reason why in subgroup L patients in post-TAP group got higher satisfaction score compared to their counterparts in post-TAP group.

The degree of PONV in pre-TAP group was lower than that in post-TAP group, and the same results happened in all the subgroups. This might be due to the lower consumption of opioids in pre-TAP group. This result was consistent with Shin et al.'s study [30].

5. Conclusion

It was necessary to decide the time of TAP block according to the duration of surgery. For patients undergoing laparoscopic gynecological surgery, preoperative TAP block was recommended for duration of surgery <180 min for lower consumption of intraoperative opioids, while postoperative TAP block was better than preoperative manipulation for duration of surgery >180 min, which might obtain lower postoperative pain score, less postoperative analgesics, and higher satisfaction score. Further research is warranted to investigate whether the TAP block technique can be improved by optimizing dose and technique-related factors.

Data Availability

All the underlying data supporting the results of this study can be found in IRB of the Second Hospital of Dalian Medical University.

Disclosure

Meiyu Wei and Ming Liu are the co-first authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Meiyu Wei and Ming Liu contributed equally to this work.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. H81471373).

References

[1] S. Hashiguchi, "PCEA and IVPCA for acute postoperative pain management," *Masui*, vol. 59, no. Suppl, pp. S49–S53, 2010.

[2] P. Lirk, J. Thiry, M.-P. Bonnet, G. P. Joshi, and F. Bonnet, "Pain management after laparoscopic hysterectomy: systematic review of literature and PROSPECT recommendations," *Regional Anesthesia & Pain Medicine*, vol. 44, no. 4, pp. 425–436, 2019.

[3] M. B. Rodríguez-Campoó, A. Curto, M. González, and C. Aldecoa, "Patient intermittent epidural boluses (PIEB) plus very low continuous epidural infusion (CEI) versus patient-controlled epidural analgesia (PCEA) plus continuous epidural infusion (CEI) in primiparous labour: a randomized trial," *Journal of Clinical Monitoring and Computing*, vol. 33, no. 5, pp. 879–885, 2019.

[4] P. Tian, X. Fu, Z. J. Li, and X. L. Ma, "Comparison of patient-controlled epidural analgesia and patient-controlled intravenous analgesia after spinal fusion surgery: a meta-analysis of randomized controlled trials," *BMC Musculoskelet Disord*, vol. 16, p. 388, 2015.

[5] L. Han, Y. Su, H. Xiong et al., "Oxycodone versus sufentanil in adult patient-controlled intravenous analgesia after abdominal surgery," *Medicine*, vol. 97, no. 31, p. e11552, 2018.

[6] S. Guo, G. Duan, J. Wang, X. Chi, L. Zhang, and X. Zhang, "Comparison of sufentanil-tramadol PCIA between laparoscopic cholecystectomy and gynecological laparoscopy," *Zhonghua Wai Ke Za Zhi*, vol. 53, no. 2, pp. 150–154, 2015.

[7] S. Baranovic, B. Maldini, M. Milosevic, R. Golubic, and T. Nikolic, "Peripheral regional analgesia with femoral catheter versus intravenous patient controlled analgesia after total knee arthroplasty: a prospective randomized study," *Collegium Antropologicum*, vol. 35, no. 4, pp. 1209–1214, 2011.

[8] S. H. Kim, Y.-S. Shin, Y. J. Oh, J. R. Lee, S. C. Chung, and Y. S. Choi, "Risk assessment of postoperative nausea and vomiting in the intravenous patient-controlled analgesia environment: predictive values of the Apfel's simplified risk score for identification of high-risk patients," *Yonsei Medical Journal*, vol. 54, no. 5, pp. 1273–1281, 2013.

[9] M. Zhou, L. Wang, C. Wu et al., "Efficacy and safety of different doses of dezocine for preemptive analgesia in gynecological laparoscopic surgeries: a prospective, double blind and randomized controlled clinical trial," *International Journal of Surgery*, vol. 37, no. Suppl 1, pp. 539–545, 2017.

[10] E. M. Sandberg, A. R. H. Twijnstra, S. R. C. Driessen, and F. W. Jansen, "Total laparoscopic hysterectomy versus vaginal hysterectomy: a systematic review and meta-analysis," *Journal of Minimally Invasive Gynecology*, vol. 24, no. 2, pp. 206–217, 2017.

[11] J. Joo, H. K. Moon, and Y. E. Moon, "Identification of predictors for acute postoperative pain after gynecological laparoscopy (STROBE-compliant article). Medicine (Baltimore)," vol. 98, no. 42, p. e17621, 2019.

[12] X. Wang, W. Liu, Z. Xu et al., "Effect of dexmedetomidine alone for intravenous patient-controlled analgesia after gynecological laparoscopic surgery," *Medicine (Baltimore)*, vol. 95, no. 19, p. e3639, 2016.

[13] E. J. Ahn, G. J. Choi, H. Kang, C. W. Baek, Y. H. Jung, and Y. C. Woo, "Comparison of ramosetron with palonosetron for prevention of postoperative nausea and vomiting in patients receiving opioid-based intravenous patient-controlled analgesia after gynecological laparoscopy," *BioMed Research International*, vol. 2017, Article ID 9341738, , 2017.

[14] H. C. Tsai, T. Yoshida, T. Y. Chuang et al., "Transversus abdominis plane block: an updated review of anatomy and techniques," *Biomed Res Int*, vol. 2017, Article ID 8284363, 2017.

[15] J. Y. Yap, M. Bhat, W. McMullen, and K. Ragupathy, "Novel use of laparoscopic-guided TAP block in total laparoscopic hysterectomy," *Journal of Obstetrics and Gynaecology*, vol. 38, no. 5, p. 736, 2018.

[16] F. W. Abdallah, S. H. Halpern, and C. B. Margarido, "Transversus abdominis plane block for postoperative analgesia after Caesarean delivery performed under spinal anaesthesia? a systematic review and meta-analysis," *British Journal of Anaesthesia*, vol. 109, no. 5, pp. 679–687, 2012.

[17] K. K. Khan and R. I. Khan, "Analgesic effect of bilateral subcostal tap block After laparoscopic cholecystectomy," *Journal of Ayub Medical College, Abbottabad*, vol. 30, no. 1, pp. 12–15, 2018.

[18] M. Shahait and D. I. Lee, "Application of TAP block in laparoscopic urological surgery: current status and future directions," *Current Urology Reports*, vol. 20, no. 5, p. 20, 2019.

[19] J. Ruiz-Tovar, E. Albrecht, A. Macfarlane, and F. Coluzzi, "The TAP block in obese patients: pros and cons," *Minerva Anestesiologica*, vol. 85, no. 9, pp. 1024–1031, 2019.

[20] J. Jakobsson, L. Wickerts, S. Forsberg, and G. Ledin, "Transversus abdominal plane (TAP) block for postoperative pain management: a review," *F1000Research*, vol. 4, 2015.

[21] B. Pirrera, V. Alagna, A. Lucchi et al., "Transversus abdominis plane (TAP) block versus thoracic epidural analgesia (TEA) in laparoscopic colon surgery in the ERAS program," *Surgical Endoscopy*, vol. 32, no. 1, pp. 376–382, 2018.

[22] A. Özdilek, Ç. A. Beyoğlu, Ç. Demirdağ et al., "Perioperative analgesic effects of preemptive ultrasound-guided subcostal transversus abdominis plane block for percutaneous nephrolithotomy: a prospective, randomized trial," *Journal of Endourology*, vol. 34, no. 4, pp. 434–440, 2020.

[23] A. Kupiec, J. Zwierzchowski, J. Kowal-Janicka et al., "The analgesic efficiency of transversus abdominis plane (TAP) block after caesarean delivery," *Ginekologia Polska*, vol. 89, no. 8, pp. 421–424, 2018.

[24] N. Ma, J. K. Duncan, A. J. Scarfe, S. Schuhmann, and A. L. Cameron, "Clinical safety and effectiveness of transversus abdominis plane (TAP) block in post-operative analgesia: a systematic review and meta-analysis," *Journal of Anesthesia*, vol. 31, no. 3, pp. 432–452, 2017.

[25] M. Mansouri, K. Bageri, E. Noormohammadi, M. Mirmohammadsadegi, A. Mirdehgan, and A. G. Ahangaran, "Randomized controlled trial of bilateral intrapleural block in cardiac surgery," *Asian Cardiovascular and Thoracic Annals*, vol. 19, no. 2, pp. 133–138, 2011.

[26] G. Niraj, A. Searle, and M. Mathews, "Analgesic efficacy of ultrasound-guided transversus abdominis plane block in patients undergoing openappendicectomy," *British Journal of Anaesthesia*, vol. 4, pp. 601–605, 2011.

[27] J. G. McDonnell, B. D. O'Donnell, T. Farrell et al., "Transversus abdominis plane block," *Regional Anesthesia and Pain Medicine*, vol. 32, no. 5, pp. 399–404, 2007.

[28] J. L. H. French, J. McCullough, P. Bachra, and N. M. Bedforth, "Transversus abdominis plane block for analgesia after

caesarean section in a patient with an intracranial lesion," *International Journal of Obstetric Anesthesia*, vol. 18, no. 1, pp. 52–54, 2009.

[29] J. J. Fibla, L. Molins, J. M. Mier, A. Sierra, and G. Vidal, "A prospective study of analgesic quality after a thoracotomy: paravertebral block with ropivacaine before and after rib spreading," *European Journal of Cardio-Thoracic Surgery*, vol. 36, no. 5, pp. 901–905, 2009.

[30] H.-J. Shin, S. T. Kim, K. H. Yim, H. S. Lee, J. H. Sim, and Y. D. Shin, "Preemptive analgesic efficacy of ultrasound-guided transversus abdominis plane block in patients undergoing gynecologic surgery via a transverse lower abdominal skin incision," *Korean Journal of Anesthesiology*, vol. 61, no. 5, pp. 413–418, 2011.

[31] K. Stoving, C. Rothe, C. V. Rosenstock, E. K. Aasvang, L. H. Lundstrom, and K. H. Lange, "Cutaneous sensory block Area, muscle-relaxing effect, and block duration of the transversus abdominis plane block: a randomized, blinded, and placebo-controlled study in healthy volunteers," *Regional Anesthesia and Pain Medicine*, vol. 40, no. 4, pp. 355–362, 2015.

Hindawi
Scientific Programming
Volume 2021, Article ID 5573751, 9 pages
https://doi.org/10.1155/2021/5573751

Research Article

CPGAN: An Efficient Architecture Designing for Text-to-Image Generative Adversarial Networks Based on Canonical Polyadic Decomposition

Ruixin Ma and Junying Lou

School of Software, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Ruixin Ma; maruixin@dlut.edu.cn

Received 20 January 2021; Revised 24 February 2021; Accepted 11 March 2021; Published 2 April 2021

Academic Editor: Liang Zou

Text-to-image synthesis is an important and challenging application of computer vision. Many interesting and meaningful text-to-image synthesis models have been put forward. However, most of the works pay attention to the quality of synthesis images, but rarely consider the size of these models. Large models contain many parameters and high delay, which makes it difficult to be deployed on mobile applications. To solve this problem, we propose an efficient architecture CPGAN for text-to-image generative adversarial networks (GAN) based on canonical polyadic decomposition (CPD). It is a general method to design the lightweight architecture of text-to-image GAN. To improve the stability of CPGAN, we introduce conditioning augmentation and the idea of autoencoder during the training process. Experimental results prove that our architecture CPGAN can maintain the quality of generated images and reduce at least 20% parameters and flops.

1. Introduction

Text-to-image synthesis is a challenging cross modal generation which generates images according to given texts. It extracts the common modal data from texts and transfers the semantic data into images. Text-to-image synthesis plays a more and more important role in computer vision. Images were edited by images in the past. With the development of text-to-image synthesis, images can also be edited by text, which greatly expands the application of computer vision. Text-to-image synthesis can be widely applied in human-computer interaction, such as cross modal retrieval [1] and artistic creation [2, 3].

Traditional text-to-image synthesis used variational autoencoder (VAE), attention mechanism, and recurrent neural network (RNN) to generate images step by step [4, 5]. Limited by generative ability of VAE, generated images are not as clear as real images. A new generative model GAN was proposed by Goodfellow et al. in 2014 [6]. GAN becomes a popular model in image generation task due to its strong generating ability. Reed et al. [7] proved that GAN could be used to generate clear images from text description and proposed GAN-int-cls. It uses DCGAN as the backbone, text embedding, and random noises as inputs of the generator. The generated images, text embedding, and real images are inputs of the discriminator. Subsequently, many sophisticated models were proposed. These models can generate images according to general text, scene graph, or dialog. The quality of generated images has been improved a lot.

However, these models introduced many constraints and modules to generate realistic images. These will greatly increase parameters and floating-point operations per second (flops) of models. It will require more and more hardware resources (CPU, GPU, memory, and bandwidth) to deploy these models. High complexity also leads to high latency. This greatly limits application of text-to-image GAN in mobile terminal. It is necessary to compress text-to-image GAN. Canonical polyadic decomposition (CPD) is an easy and efficient way to compress and accelerate model in tensor decomposition. Many implementations of convolutional

neural networks (CNN) compression based on CPD [7–9] have already been proposed.

In this paper, we propose a general compressed architecture CPGAN for designing text-to-image GAN to reduce parameters and flops. CPGAN redesigns each layer of the original neural network by using CPD. The original convolution layer is decomposed into three convolution layers with different ranks and small size. A layer with a smaller rank has few parameters. According to the needs of the application, we can design architectures with different compression ratios by setting different ranks. During the training process of models with different ranks, it is time-consuming to select the appropriate learning rate. To this end, we use cyclical learning rate (CLR) [11] method to select the optimal learning rate for the redesigned architecture. In addition, GAN has the problem of unstable training. CPGAN is a deeper architecture than the classical GAN and is difficult to train from scratch. To solve this problem, we add conditioning augmentation module and introduce the idea of autoencoder method.

Our contributions can be summarized as follows:

(i) We propose CPGAN to reduce parameters and maintain the generative ability of text-to-image GAN. It is a general method to design the light-weight architecture of GAN.

(ii) To reduce high resource consumption caused by decomposition operation, we train CPGAN from scratch and do not need to pretrain the model. To the best of our knowledge, it is the first time to use CPD to design text-to-image GAN without using pretrained model.

(iii) To stable the end-to-end training, we introduce the idea of autoencoder. The added decoder modules can be removed after training.

Experimental results on two representative cross modal datasets (Oxford-102 and CUB) prove that our architecture CPGAN can maintain the quality of generated images and reduce parameters and flops of original model effectively at the same time. In Oxford-102 and CUB, CPGAN performs better in inception score (IS) and Fréchet inception distance (FID) than original model. It reduces 8.8×10^9 flops and 1.31×10^6 parameters in Oxford-102. These show that our architecture can efficiently redesign text-to-image GAN without loss of image quality.

The rest of the paper is organized as follows. The work related to our paper is introduced in Section 2. In Section 3, we propose the efficient architecture CPGAN of text-to-image generative adversarial networks (GAN) based on canonical polyadic decomposition (CPD). Section 4 describes experimental settings and experimental results. Finally, we conclude this paper in Section 5.

2. Related Work

2.1. Canonical Polyadic Decomposition. The essence of neural network is the matrix transformation process of input data matrix using weight parameters. Each layer of neural network is a large tensor, which can be decomposed into several small tensors. Canonical polyadic decomposition (CPD) is a standard tensor decomposition method. It was proposed by Hitchcock in 1927 [12]. It can decompose a tensor into a sum of rank-one tensors. CPD has been applied in psychometrics [13], signal processing [14], computer vision [15], data mining [16], and elsewhere. It also performs well in model compression.

Denton et al. [8] used CPD to approximate the original convolution kernel and presented two methods of improving approximation criterion. They performed fine-tuning on the decomposed kernels by fixing other layers. Jaderberg et al. [9] applied CPD to decompose a 4D kernel into two small kernels and use two methods to reconstruct the original filters. Lebedev et al. [10] used CPD to decompose the 4D convolution kernel tensor into four small kernels with nonlinear least squares and then replace original layer. Then, they fine-tuned the entire network using backpropagation. Lebedev et al.'s [10] method accelerated the second convolutional layer of AlexNet by 6.6 times at the cost of 1% accuracy loss. This exceeded the other two works, where Denton et al. [8] got 2 times speed-up and Jaderberg et al. [9] got 4.5 times speed-up at the cost of 1% accuracy loss.

Astrid et al. [17] proposed a CNN compression method based on CPD : CP-TPM. It achieved 6.98 times parameter reduction and 3.53 times speeding-up in AlexNet. It is better than the Tucker-based method [18] in the same network. Zhang et al. [19] and Tai et al. [20] also applied CPD to compress CNN. Original layers are pretrained to minimize the difference between the decomposed layer and the original tensor in the models of Astrid et al. [17], Zhang et al. [19], and Tai et al. [20]. Because CP decomposition operation consumes extensive resources, we do not decompose the pretrained weight tensor, but directly use CP decomposition to design an efficient architecture in text-to-image GAN.

2.2. Text-to-Image Synthesis. Text-to-image synthesis is a branch of computer vision which generates images according to given texts. It can be used for image editing, cross modal retrieval, and artistic creation. GAN has strong generating ability. It can generate realistic images and has been widely used for image generation. Since Reed et al. [7] first successfully used GAN for text image generation, GAN has also become a popular model in text-to-image synthesis.

Reed et al. [7] proposed GAN-int-cls by revising DCGAN and successfully generated plausible 64×64 images of birds and flowers from texts. In order to produce high resolution images, multiple stages generating was introduced into text-to-image synthesis, such as StackGAN [21], StackGAN++ [22], HDGAN [23], and LAPGAN [24]. StackGAN [21] stacked two conditional GANs to generate high resolution and plain images in two stages. Multiple generators were used to generate images of different scales using tree structure in StackGAN++ [22]. HDGAN [23] adopted hierarchically-nested discriminators to help the single-stream generator generate high resolution images.

LAPGAN [24] put forward a Laplacian pyramid framework through integrating a set of generators.

Xu et al. [25] and Qiao et al. [26] added attention mechanisms to synthesize image with fine-grained details. Besides, Reed et al. [27] adapted bounding box and key part information to improve quality of generated images. ACGAN [28] and TAC-GAN [29] used auxiliary class information to generate diversity images. Because these models show excellent cross modal generative ability, text-to-image GAN has been used for image editing [30, 31], cross modal retrieval [1], story visualization [2], and painting [3]. However, these models are too complicated to be deployed on the mobile end. To this end, we propose an end-to-end compression framework based on CPD. Compared to Shu et al. [32] and Li et al. [33], we do not need to pretrain GAN model. We design and train the compression model from scratch.

3. Canonical Polyadic Generative Adversarial Networks (CPGAN)

In this section, we introduce the designing of the efficient architecture (CPGAN) and the training process. Section 3.1 describes how to replace 4-dimensional convolutional weight tensors with three small kernels. Section 3.2 describes techniques for stabling training process of the redesigned architecture.

3.1. Canonical Polyadic Decomposition. GAN consists of a generator and a discriminator in general, both of which are convolutional neural networks. The weight tensor for convolution is a 4-dimensional tensor $\mathcal{W} \in \mathbb{R}^{K\times K\times S\times T}$, which maps input $\mathcal{X} \in \mathbb{R}^{I\times J\times S}$into another representation $\mathcal{Y} \in \mathbb{R}^{X\times Y\times T}$. It can be written as

$$\mathcal{Y}(x, y, t) = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{s=1}^{S}\mathcal{W}(k,k,s,t)\mathcal{X}(i, j, s), \tag{1}$$

where the first two dimensions of $\mathcal{W}(k,k,s,t)$ are the spatial dimension (K is typically 3 or 5), the third dimension is the input channel, and the fourth dimension is the output channel.

CPD is an approximation method which decomposes a tensor into a sum of rank-one tensors. In CPD, tensor $\mathcal{W} \in \mathbb{R}^{K\times K\times S\times T}$ can be represented as

$$\mathcal{W}(i, j, s, t) = \sum_{r=1}^{R}\mathcal{W}_{i,r}^{(1)}\mathcal{W}_{j,r}^{(2)}\mathcal{W}_{s,r}^{(3)}\mathcal{W}_{t,r}^{(4)}, \tag{2}$$

where R is the tensor rank and it is the sum of rank-one tensors, $\mathcal{W}_{i,r}^{(1)}$, $\mathcal{W}_{j,r}^{(2)}$, and $\mathcal{W}_{s,r}^{(3)}$, $\mathcal{W}_{t,r}^{(4)}$ are tensors of size $K \times R$, $K \times R$, $S \times R$, $T \times R$, respectively. Rank-one tensor is the vector outer product. Rank selection decides the compression ratio and it is a NP-hard problem in rank decomposition.

In convolutional layer, spatial dimension K does not have to be decomposed because the benefits of spatial decomposition are quite small. By using the variant of CP decomposition, tensor can be decomposed as

$$\mathcal{W}(i, j, s, t) = \sum_{r=1}^{R}\mathcal{W}_{s,r}^{(1)}\mathcal{W}_{i,j,r}^{(2)}\mathcal{W}_{t,r}^{(3)}, \tag{3}$$

where $\mathcal{W}_{i,j,r}^{(2)}$ is a tensor of size $K \times K \times R$. Substituting equation (3) into equation (2), we obtain the following approximate representation of the convolution:

$$\mathcal{Y}(x, y, t) = \sum_{r=1}^{R}\mathcal{W}_{t,r}^{(3)}\left(\sum_{j=1}^{D}\sum_{i=1}^{D}\mathcal{W}_{r,j,i}^{(2)}\left(\sum_{r=1}^{R}\mathcal{W}_{r,s}^{(1)}\mathcal{X}(i, j, s)\right)\right). \tag{4}$$

Performing rearranging and combining, we can get the following three consecutive expressions:

$$\mathcal{Y}^{(1)}(i, j, r) = \sum_{s=1}^{S}\mathcal{W}_{r,s}^{(1)}\mathcal{X}(i, j, s), \tag{5}$$

$$\mathcal{Y}^{(2)}(x, y, r) = \sum_{j=1}^{D}\sum_{i=1}^{D}\mathcal{W}_{r,j,i}^{(2)}\mathcal{Y}^{(1)}(i, j, r), \tag{6}$$

$$\mathcal{Y}(x, y, t) = \prod_{r=1}^{R}\mathcal{W}_{t,r}^{(3)}\mathcal{Y}^{(2)}(x, y, r), \tag{7}$$

where $\mathcal{Y}^{(1)}$ and $\mathcal{Y}^{(2)}$ are the intermediate tensors of sizes $I \times J \times R$ and $I' \times J' \times R$, respectively. The original big layer can be decomposed into three small layers, as shown in Figure 1. For example, the third convolution layer of GAN-int-cls has 128 input channels, 512 output channels, and 3×3 filters ($128 \times 512 \times 3 \times 3$); we can decompose it into three convolution layers with the following parameters: $128 \times R \times 1 \times 1$, $R \times R \times 3 \times 3$, and $R \times 512 \times 1 \times 1$. R is the rank which can be set as different values according to the need of tasks.

3.2. Overall Framework. We take the classical model GAN-int-cls as the original model to compress. This model has the most compact structure and parameters. The main convolution layers of the generator in other text-to-image GAN models are similar to GAN-int-cls. We redesign GAN-int-cls to show the effectiveness and generality of our compression architecture. As shown in Figure 2, the proposed CPGAN contains two novel components which can stabilize the training of decomposed GAN: conditioning augmentation and autoencoder module.

Conditioning augmentation *(CA)* is proposed by Zhang et al. [21] which alleviates the difficulty of GAN training caused by text embedding sparsity. CA is to randomly sample the hidden variables as the input of the generator from the independent Gaussian distribution $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$. φ_t is the text embedding which is generated by encoding text description. $\mu(\varphi_t)$ and $\Sigma(\varphi_t)$ are the mean and diagonal covariance matrix functions of the text embedding φ_t, respectively. We use pretrained char–CNN–RNN [34] to get the text embedding φ_t. Then, we feed φ_t into CA and obtain $\mu(\varphi_t)$ and $\Sigma(\varphi_t)$. Similar to StackGAN [21], we also add the Kullback–Leibler (KL) divergence into our training objectives, which is the KL

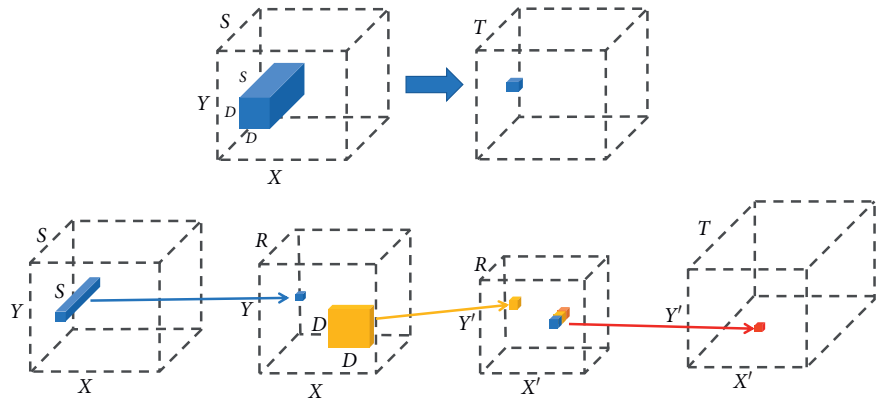


Figure 1: Original convolution layer and decomposed three convolution layers based on CPD. The top figure is the original filter of size $D \times D \times S \times T$. The bottom figure is the three decomposed filters of sizes of $1 \times 1 \times S \times R$, $K \times K \times R \times R$, and $1 \times 1 \times R \times T$. The three decomposed filters can approximate the original filter.

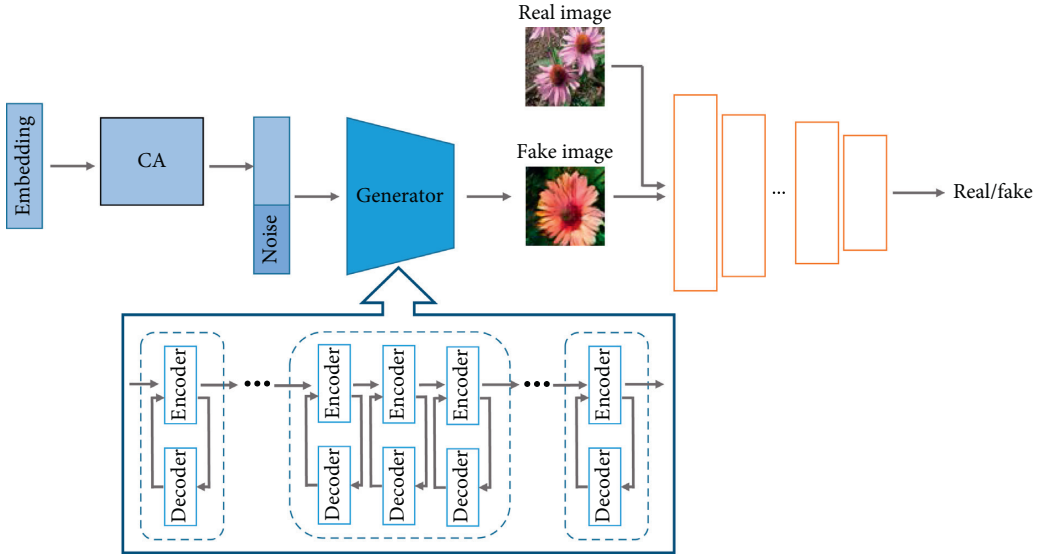


Figure 2: The architecture of CPGAN. The three encoding layers in the blue dotted box are the three decomposed filters obtained by decomposing the original convolution layer.

divergence between the standard Gaussian distribution $\mathcal{N}(0, I)$ and the conditioning Gaussian distribution $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$, as shown in the following equation:

$$D_{\mathrm{KL}}\left(\mathcal{N}\left(\mu\left(\varphi_t\right), \Sigma\left(\varphi_t\right)\right) \| \mathcal{N}(0, I)\right). \tag{8}$$

Autoencoder (AE) is used for representation learning by reconstructing input. The decomposed architecture is deeper than the original model, which increases the instability of training. So, we use AE to stabilize the training process. AE is composed of an encoder and a decoder in general. We regard each convolution layer as an encoder and add a decoder corresponding to each convolution layer. The training objective of AE is the reconstruction loss. We use mean square error (MSE) $\|x_1 - h(x_1)\|_2^2$ as the AE loss, where x_1 is the input of layer and $h(\cdot)$ is the function of AE. The decoder will be removed after training.

The generator objective of original GAN-int-cls contains matching-aware loss and interpolation loss, as shown in

$$G_{\mathrm{ori}} = G_1(z, t) + G_2\left(z, \beta t_1 + (1 - \beta) t_2\right), \tag{9}$$

where z is the random noise, t_1 and t_2 are text embeddings, and β is a decimal between 0 and 1 and used to interpolate between text embeddings t_1 and t_2.

In the generator objective of our model, we add KL divergence and MSE reconstruction loss into the original model objective, as shown in the following equation:

$$G = G_{\mathrm{ori}} + D_{\mathrm{KL}}\left(\mathcal{N}\left(\mu\left(\varphi_t\right), \Sigma\left(\varphi_t\right)\right) \| \mathcal{N}(0, I)\right) + \left\|x_1 - h\left(x_1\right)\right\|_2^2. \tag{10}$$

The discriminator objective of the original model and our model is both matching-aware loss:

$$D = D_1(x,t) + D_2(x,\hat{t}) + D_3(G(z),t). \quad (11)$$

We use the above scheme to train an efficient architecture from scratch. The training algorithm is shown in Algorithm 1. Firstly, original convolutions are decomposed into three layers through equations (5)–(7). Secondly, each layer is regarded as an encoder and a decoder is added corresponding to each layer. Thirdly, we encode matching text t and mismatching text $\hat{t}$ and get text embeddings. Then, we use CA to process text embeddings and get independent Gaussian distribution. From the independent Gaussian distribution, we sample variables and concatenate it with random noise. The following training process is the same as GAN-int-cls with different training objectives of generator. The objective function of our model adds the loss of CA and autoencoder on the basis of the original model's objective function. Until the training is finished, we remove added decoder layers and obtain the model of CPGAN.

4. Experiments

We conduct extensive experiments to evaluate the proposed CPGAN. In Section 4.1, we introduce the experimental dataset and evaluation index. Section 4.2 describes the setting of learning rate and the other experimental hyperparameters. In Section 4.3, we compare our CPGAN with previous GAN-int-cls models for text-to-image synthesis.

4.1. Overall Framework. To show the generality of our method, we choose the classic model GAN-int-cls as our original model. Same as GAN-int-cls, our method is evaluated on CUB [35] and Oxford-102 [36]. The CUB dataset covers 200 kinds of birds, including 5,994 training images and 5,794 test images. In addition to category labels, each image contains bounding box, bird key part of bird information, and bird attributes. Oxford-102 flowers dataset is a flower dataset which contains 8,189 images. It is divided into 102 categories and each category contains 40 to 258 images. Each image has large scale, pose and light variations. The dataset is divided into a training set, a validation set, and a test set. Both datasets are benchmark image datasets and each image corresponds to 10 single sentence descriptions.

In order to evaluate our model, we use inception score (IS) and Fréchet inception distance (FID) to evaluate the quality of the generated images. IS uses pretrained InceptionNet-V3 to judge whether the generated image is clear and diverse. High IS score means that images are clear and diverse. FID calculates feature distance between the real image and the fake image as a supplement of IS evaluation index. These two indicators are widely used to evaluate the quality of generated images.

4.2. Implementation Details. Learning rate is a very important hyperparameter in deep learning. Reasonable learning rate can make the model converge to the minimum point instead of the local optimal point or saddle point. In this paper, we use the method CLR [11] and MultistepLR to set learning rate and learning rate attenuation.

CLR was proposed by Smith. It changes learning rate periodically in the iterative process, rather than a fixed value. It is used to find the optimal learning rate automatically instead of manual experiments. We use CLR to get a learning rate setting. CLR method needs to set three parameters, minimum learning rate (min_lr), maximum learning rate (max_lr), and iteration. min_lr and max_lr are the smallest value and the biggest value of learning rate, respectively. Iteration is the number of test iterations at each learning rate. We increase the learning rate from 0.00001 to 0.001 and get the loss curve under different learning rates (see Figure 3).

We choose the appropriate learning rate according to maximum absolute slope criterion. According to Figure 3, we select 0.0002 and 0.00015 as the learning rates of the Oxford-102 dataset and 0.0001 and 0.00008 as the learning rates of the CUB dataset.

MultistepLR is a learning rate attenuation method in PyTorch. It has three hyperparameters: initial learning rate (ini_lr), epoch to update learning rate (epo), and multiplication factor(mfc). ini_lr is the initial learning rate during the training. epo is the epoch when we change the learning rate. mfc is the attenuation coefficient of learning rate. In the experiment using MultistepLR, the initial learning rate is *ini_lr*. When the experiment runs epo epochs, the learning rate is changed to ini_lr $*$ mfc.

In this paper, we set the MultistepLR hyperparameters ini_lr, epo, and mfc as 0.0001, 600, and 0.8 in CUB and 0.0002, 600, and 0.75 in Oxford-102. The batch size in our experiment is 64. The optimizer of CPGAN is Adam [37] with momentum of 0.5.

4.3. Comparison with Original Model. In CP decomposition, rank represents compression ratio and it is hard to select. Due to the need of text-to-image synthesis task, we design the lightweight model on the premise of ensuring the quality of generated images. We do extensive experiments to balance the performance and the compression ratio.

As shown in Table 1, we do a large number of experiments to find the balance. The ratio is rank ratio, where 1.0 is full rank decomposition and 0.9 means about 0.9 times of original layer's number of input channels. A layer with a smaller rank has few parameters. Table 1 shows that with increasing of rank, flops and parameters grow. When the decomposition rank is close to 0.7, the parameters begin to exceed the original model's parameters (5.76×10^6). With the increase of rank, the quality of images generated by the model has not been greatly improved. The value of FID decreases first and then changes slightly with the increase of model parameters, while IS is not stable. It may be that the calculation of IS needs to use the edge distribution of data, but generated samples in Oxford-102 are not enough to get accurate edge distribution.

As shown in Table 1, FID gets the best value when rank ratio is 0.5. The model is compressed by about 23% parameters and 29% flops. The generated images are better than the original model on FID and IS. It can prove that our method can generate better images with less parameters than the original

Input: mini-batch images x, text description t, and number of training batch steps S.
Output: CPGAN model.
(1) Use equations (5)–(7) to decompose the original convolutional layer in generator;
(2) Add CA module for text embedding and add decoders layers;
(3) Select an appropriate learning rate for the decomposed model;
(4) **For** $N = 1$ **to** S **do**
(5) Encode text description into embedding t;
(6) Feed t into CA and obtain $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$;
(7) Sample $\hat{c}$ from $\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t))$ and random noise z;
(8) Concatenate z and $\hat{c}$ and feed it into the generator;
(9) Update discriminator D by equation (11);
(10) Update generator G by equation (10);
(11) **End for**
(12) Discard all decoders and get a trained CPGAN.

Algorithm 1: Overall scheme of CPGAN algorithm.

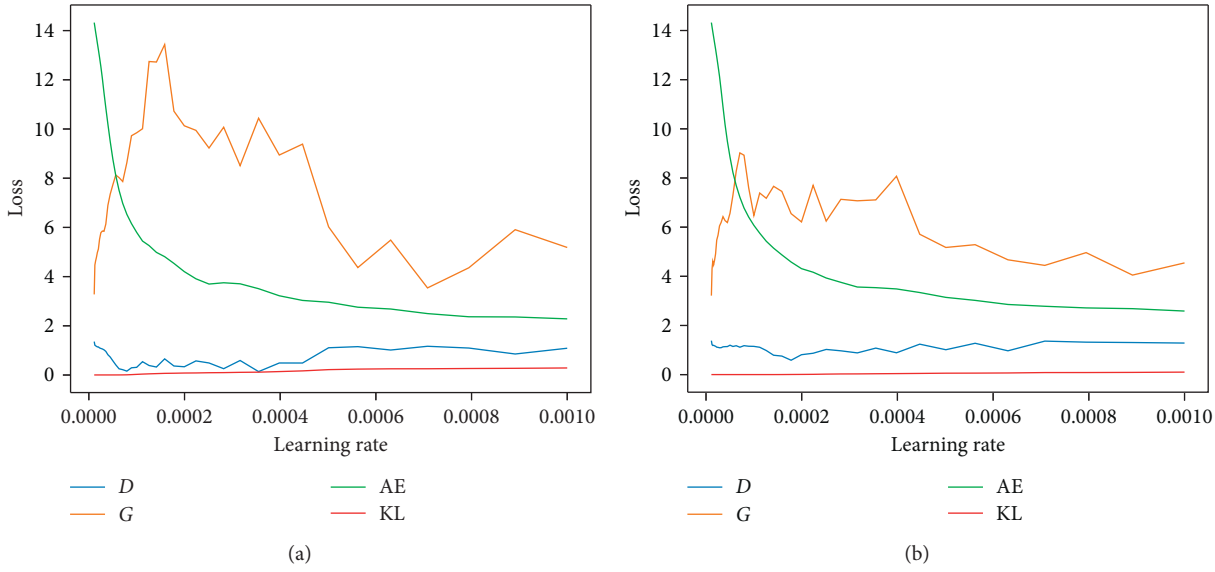


Figure 3: Selection for appropriate learning rate for CPGAN. (a) Loss of different learning rates on Oxford-102. (b) Loss of different learning rates on CUB.

Table 1: Experimental results of different rank ratios for the CPGAN in Oxford-102.

Ratio	FID	IS	Flops	#Parameters
0.1	158.74	3.26 ± 0.05	1.20×10^{10}	2.91×10^{6}
0.2	98.06	3.43 ± 0.06	1.35×10^{10}	3.13×10^{6}
0.3	85.08	2.98 ± 0.05	1.54×10^{10}	3.46×10^{6}
0.4	81.17	2.96 ± 0.04	1.80×10^{10}	3.90×10^{6}
0.5	**74.69**	$\mathbf{3.04 \pm 0.05}$	2.14×10^{10}	4.45×10^{6}
0.6	77.59	3.54 ± 0.06	2.53×10^{10}	5.10×10^{6}
0.7	76.50	2.80 ± 0.06	3.00×10^{10}	5.87×10^{6}
0.8	79.04	3.56 ± 0.05	3.53×10^{10}	6.74×10^{6}
0.9	76.97	3.17 ± 0.05	4.14×10^{10}	7.72×10^{6}
1.0	77.07	3.23 ± 0.05	4.83×10^{10}	8.82×10^{6}

model. It is effective to use CP decomposition to reconstruct the model and design compact text-to-image GAN without loss of image quality. Although 8.8×10^{9} flops and 1.31×10^{6} parameters are reduced, the images generated by CPGAN get a little improvement on IS and FID. This shows that the image generated by the model with more parameters may not be better. So around the rank of 0.5, we look for a better model ensuring the quality of generated images.

Table 2 shows the comparison between our best generative model and the original model on IS, FID, parameters, and flops. FID and IS of the original model are 79.55 and 2.66 ± 0.03 in Oxford-102, while those of our best model are 74.40 and 3.68 ± 0.08, respectively. In CUB, the images generated by our best model get 65.94 on FID and 5.03 ± 0.07 on IS, while those of original model are 68.79 and 2.88 ± 0.04, respectively. The comparison of representative images on Oxford-102 and CUB dataset can be seen in Figures 4 and 5, respectively. The better generated images of CPGAN indicate that our proposed method can generate more realistic images from text descriptions. These results also prove that there are redundant parameters in existing text-to-image GAN. A more concise and efficient text-to-image GAN model can be designed based on CPD.

Table 2: Comparison between our model and the original model.

Model	Oxford-102		CUB		Flops	#Parameters
	FID	IS	FID	IS		
Original	79.55	2.66 ± 0.03	68.79	2.88 ± 0.04	3.02×10^{10}	5.76×10^{6}
Redesigned	74.40	3.68 ± 0.08	65.94	5.03 ± 0.07	2.33×10^{10}	5.07×10^{6}

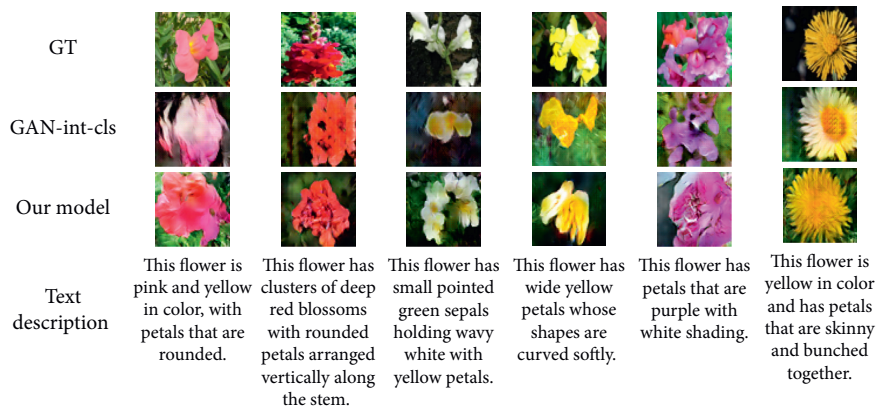


Figure 4: Generated images by our proposed model and the original model on Oxford-102.

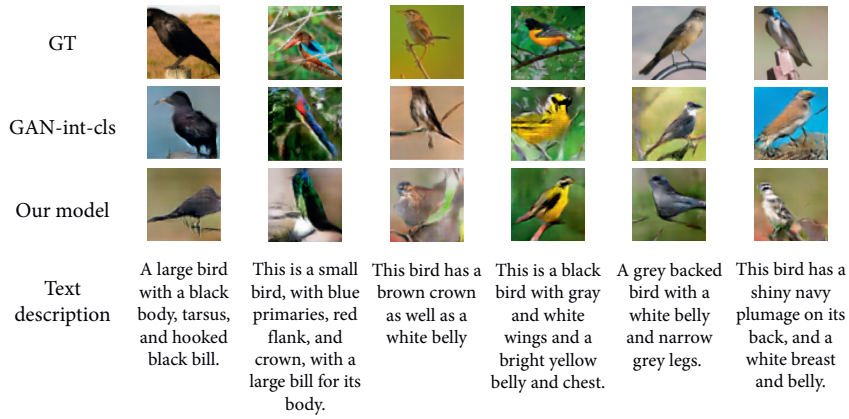


Figure 5: Generated images by our proposed model and the original model on CUB.

5. Conclusions

In this paper, we propose a simple and efficient architecture CPGAN based on CPD. CPGAN can reduce extensive parameters and flops of the original model. It also improves the quality of generated images at the same time. In the process of designing CPGAN model, we replace the convolution layer with three CP decomposed small layers to achieve a certain compression. In order to stabilize the training process, we introduce conditioning augmentation to reduce the instability caused by text embedding sparsity. To further improve the end-to-end training of our model, the idea of autoencoder is integrated into the model. Each decomposed layer can be regarded as an encoder layer and is paired with an added decoder layer. The decoder layers can be removed after training. Experiments demonstrate that CPGAN reduces about 23% parameters and 29% flops with a little improvement of generated image quality in Oxford-102. Extensive experimental results demonstrate that our proposed CPGAN can design an efficient text-to-image GAN. We have also decomposed similar convolution layers in other GAN models and these experiment results were similar to the experiment results of GAN-int-cls. The main convolution layers of the generator in other text-to-image GAN models are similar to GAN-int-cls. It is applicable for other cross modal GANs to use CPD. In the existing methods, the rank is set manually, which is time-consuming. Therefore, the automatic selection of rank may be a research direction in the future.

Data Availability

The datasets used in this paper are public datasets which can be accessed via the following websites: http://www.vision.caltech.edu/visipedia/CUB-200-2011.html and https://www.robots.ox.ac.uk/~vgg/data/flowers/102/

Conflicts of Interest

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

References

[1] J. Gu, J. Cai, S. R. Joty et al., "Look, imagine and match: improving textual-visual cross-modal retrieval with generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7181–7189, Salt Lake City, UT, USA, 2018.

[2] Y. Li, Z. Gan, Y. Shen et al., "StoryGAN: a sequential conditional GAN for story visualization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6329–6338, Seattle, WA, USA, 2019.

[3] W. Li, P. Zhang, L. Zhang et al., "Object-driven text-to-image synthesis via adversarial training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12174–12182, Seattle, WA, USA, 2019.

[4] 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.

[5] A. van den Oord, N. Kalchbrenner, L. Espeholt et al., "Conditional image generation with PixelCNN decoders," *Advances in Neural Information Processing Systems*, vol. 29, pp. 4790–4798, 2016.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 2, pp. 2672–2680, 2014.

[7] S. Reed, Z. Akata, X. Yan et al., "Generative adversarial text to image synthesis," *Proceedings of Machine Learning Research*, vol. 48, pp. 1060–1069, 2016.

[8] E. Denton, W. Zaremba, J. Bruna et al., "Exploiting linear structure within convolutional networks for efficient evaluation," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1269–1277, 2014.

[9] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," *Proceedings of the British Machine Vision Conference*, 2014.

[10] V. Lebedev, Y. Ganin, M. Rakhuba et al., "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," 2014, http://arxiv.org/abs/1412.6553.

[11] 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.

[12] 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.

[13] P. M. Kroonenberg, *Applied Multiway Data Analysis*, Wiley-Interscience, Hoboken, NJ, USA, 2008.

[14] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI)," *Signal Processing*, vol. 93, no. 9, pp. 2722–2738, 2013.

[15] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: tensorfaces," in *Proceedings of the European Conference on Computer Vision*, pp. 447–460, Copenhagen, Denmark, 2002.

[16] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy et al., "Modeling and multiway analysis of chatroom tensors," in *Proceedings of the International Conference On Intelligence And Security Informatics*, pp. 256–268, San Diego, CA, USA, 2005.

[17] M. Astrid and S. I. Lee, "CP-decomposition with tensor power method for convolutional neural networks compression," in *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 115–118, Jeju, South Korea, 2017.

[18] Y. D. Kim, E. Park, S. Yoo et al., "Compression of deep convolutional neural networks for fast and low power mobile applications," *Computer Science*, vol. 71, no. 2, pp. 576–584, 2015.

[19] Q. Zhang, L. T. Yang, Z. Chen et al., "An improved deep computation model based on canonical polyadic decomposition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1657–1666, 2017.

[20] C. Tai, T. Xiao, Y. Zhang et al., "Convolutional neural networks with low-rank regularization," 2015, http://arxiv.org/abs/1404.3978.

[21] 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.

[22] 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, 2018.

[23] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6199–6208, Salt Lake City, UT, USA, 2018.

[24] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 1486–1494, 2015.

[25] T. Xu, P. Zhang, Q. Huang et al., "AttnGAN: fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1316–1324, Salt Lake City, UT, USA, 2018.

[26] T. Qiao, J. Zhang, D. Xu et al., "Learn, imagine and create: text-to-image generation from prior knowledge," *Advances in Neural Information Processing Systems*, pp. 887–897, 2019.

[27] S. E. Reed, Z. Akata, S. Mohan et al., "Learning what and where to draw," *Advances In Neural Information Processing Systems*, pp. 217–225, 2016.

[28] A. Odena and C. Olah, "Conditional image synthesis with auxiliary classifier GANs," in *Proceedings of the International Conference on Machine Learning*, pp. 2642–2651, Sydney, Australia, 2017.

[29] A. Dash, J. C. Gamboa, S. Ahmed et al., "TAC-GAN - text conditioned auxiliary classifier generative adversarial network," 2017, http://arxiv.org/abs/1703.06412.

[30] H. Dong, S. Yu, C. Wu et al., "Semantic image synthesis via adversarial learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5706–5714, Venice, Italy, 2017.

[31] D. M. Vo and A. Sugimoto, "Paired-D GAN for semantic image synthesis," in *Proceedings of the Asian Conference on Computer Vision*, pp. 468–484, Salt Lake City, UT, USA, 2018.

[32] H. Shu, Y. Wang, X. Jia et al., "Co-evolutionary compression for unpaired image Translation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3234–3243, Seoul, Korea, 2019.

[33] M. Li, J. Lin, Y. Ding et al., "GAN compression: efficient architectures for interactive conditional GANs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5283–5293, Seoul, Korea, 2020.

[34] S. Reed, Z. Akata, H. Lee et al., "Learning deep representations of fine-grained visual descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58, Las Vegas, NV, USA, 2016.

[35] C. Wah, S. Branson, P. Welinder et al., "The Caltech-UCSD birds-200-2011 dataset," Computation & Neural Systems Technical Report, 2011.

[36] M. E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of the 2008 Sixth Indian Conference On Computer Vision, Graphics & Image Processing*, pp. 722–729, IEEE, Bhubaneswar, India, 2008.

[37] 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.

Hindawi
Scientific Programming
Volume 2021, Article ID 6692975, 13 pages
https://doi.org/10.1155/2021/6692975

Research Article

A Clustering Algorithm via Density Perception and Hierarchical Aggregation Based on Urban Multimodal Big Data for Identifying and Analyzing Categories of Poverty-Stricken Households in China

Hui Liu,[1,2] **Yang Liu**,[3] **Ran Zhang**,[1] **and Xia Wu**[4]

[1]*Kaifaqu Campus of Dalian University of Technology, No. 321, Tuqiang Street, Dalian Economic and Technological Development Zone, Dalian, Liaoning 116600, China*
[2]*Faculty of Business and Management, Universiti Teknologi MARA, Cawangan Sarawak, Jalan Meranek, 94300 Kota Samarahan, Sarawak, Malaysia*
[3]*International School of Shenyang Jianzhu University, No. 25, Hunnan East Road, Hunnan New District, Shanyang, Liaoning 110168, China*
[4]*Huawei Nanjing Research & Development Center, No. 101 Software Avenue, Yuhuatai District, Nanjing, Jiangsu 210012, China*

Correspondence should be addressed to Yang Liu; liuyang@sjzu.edu.cn

Received 11 November 2020; Revised 28 December 2020; Accepted 2 February 2021; Published 23 February 2021

Academic Editor: Liang Zou

Nowadays, urban multimodal big data are freely available to the public due to the growing number of cities, which plays a critical role in many fields such as transportation, education, medical treatment, and land resource management. The successful completion of poverty-relief work can greatly improve the quality of people's life and ensure the sustainable development of the society. Poverty is a severe challenge for human society. It is of great significance to apply machine learning to mine different categories of poverty-stricken households and further provide decision support for poverty alleviation. Traditional poverty alleviation methods need to consume a lot of manpower, material resources, and financial resources. Based on the density-based spatial clustering of applications with noise (DBSCAN), this paper designs the hierarchical DBSCAN clustering algorithm to identify and analyze the categories of poverty-stricken households in China. First, the proposed method adjusts the neighborhood radius dynamically for dividing the data space into several initial clusters with different densities. Then, neighbor clusters are identified by the border and inner distances constantly and aggregated recursively to form new clusters. Based on the idea of division and aggregation, the proposed method can recognize clusters of different forms and deal with noises effectively in the data space with imbalanced density distribution. The experiments indicate that the method has the ideal performance of clustering, which identifies the commonness and difference in characteristics of poverty-stricken households reasonably. In terms of the specific indicator "Accuracy," the accuracy increases by 2.3% compared with other methods.

1. Introduction

With the development of Information and Communication Technology, the era of multimodal big data has arrived comprehensively. Cities are the important places which are of prime importance for big data distribution, such as population, economy, transportation, and landscape [1–3]. The urban multimodal big data obtained by traditional data collection methods such as field survey and questionnaire interview cannot objectively and accurately reflect the status quo of urban development and the law of residents' activities in a wide range of time and space. Also, the obtained urban operation information has a large lag. Multimodal big data can make up for the above defects and deeply depict the urban physical space and social environment. This not only provides the possibility to objectively understand the urban

system and summarize its development rules but also provides important support for urban planning and related research studies such as poverty-relief work and urban education.

It must be admitted that urban planning based on urban multimodal big data is a very challenging task for poverty-relief work. It can improve urban environments, quality of life, and smart city systems [4, 5]. Due to the short time, heavy task of targeted poverty alleviation in the early stage, the basic information of each impoverished object and the causes of poverty are not comprehensive and accurate enough, which needs to be further enriched and improved. Poor object management mechanism is not perfect. Due to the large number of poor people in the poor villages and the complicated family situation, the number of people coming out of the basin and returning to poverty is in constant change [6]. In addition, the management mechanism of poor objects at the village level is not sound enough, so there is a lack of changes in the poor population in the poor villages.

In this paper, we focus on the tasks of identifying and analyzing categories of poverty-stricken households in China. Eradication of poverty is the historical task facing the international community. With the development of artificial intelligence (AI) technologies such as machine learning and deep learning, a growing number of researchers are making great efforts to develop and unleash the huge potential of these AI technologies in alleviating poverty [7]. China, as the largest developing country worldwide, has made a significant contribution to global poverty alleviation. In the year of 2013, the Chinese government raised the concept of targeted poverty alleviation, which aims to take targeted measures to assist each truly poverty-stricken household and eliminate various factors leading to poverty fundamentally, thus achieving the goal of sustainable poverty alleviation [8]. On the basis of the policy, this paper adopts the clustering algorithm [9] to divide the data of poverty-stricken households in China reasonably and thus identify different categories of poverty-stricken households for supporting the formulation and implementation of antipoverty measures.

Poverty-oriented scientific research depends on the analysis of poverty data. The Chinese poverty data generally come from population censuses carried out by the country, society, and universities [10]. Due to the wide coverage of population and the individual differences in educational level and psychology, respondents may not answer questionnaires according to actual conditions, which results in the subjectivity of questionnaire data. Additionally, faults in processes such as data entry and storage can easily lead to outliers and missing values in datasets. Since the quality of poverty datasets obtained by population censuses is hard to guarantee, it brings certain difficulties for the design and application of clustering algorithms.

The design of clustering algorithms for poverty datasets should make reasonable consideration of noises caused by missing values and outliers. Nowadays, common clustering methods mainly include partitional clustering, hierarchical clustering, and density-based clustering [11]. The K-means clustering algorithm achieves clustering through the partition, which assigns each sample to the closest cluster according to distances between samples and prototypes and updates prototypes by the average of samples within clusters, then repeats the above steps until the iteration ends [12]. Although the method is easy and practicable, the number of clusters and the initial prototypes need to be predefined. The agglomerative hierarchical clustering (AHC) regards each sample as a separate cluster and then merges the two closest clusters into a new cluster constantly [13]. The AHC algorithm requires no predefined prototypes and can get the hierarchical structure of clusters, but it is sensitive to noises within data. The density-based spatial clustering of applications with noise (DBSCAN) algorithm is a representative of density-based clustering methods, which defines the cluster as the maximal set of density-connected samples and takes the sample regions with high densities as clusters, thus discovering clusters of arbitrary shapes [14] whereas the hyperparameters eps and minpts in the DBSCAN algorithm, i.e., the neighborhood radius and the minimum number of samples required to form a dense region, have a great influence on the result of clustering, and the method is not applicable to datasets with different density distribution. Many researchers improve DBSCAN in view of the existing problems in the algorithm and propose improved algorithms such as K-nearest neighbor DBSCAN (KNNDBSCAN), DVBSCAN, and varied density-based spatial clustering of applications with noise (VDBSCAN) [15–18]. For instance, Gaonkar and Sawant [19] drew a k-dist graph based on the distance between each sample and its k-th nearest neighbor so as to identify multiple values of the neighborhood radius, then finds the clusters with different densities under each value of the neighborhood radius. Fahim et al. proposed an enhanced DBSCAN (EDBSCAN) algorithm, which defined the density variation for core points and specified that a core point allowed for expansion only when its density variation was less than or equal to a threshold value and its neighborhood satisfies the homogeneity index [20]. In terms of the clustering methods, some other researchers proposed many advanced approaches such as robust FCM clustering [21], improved quantum clustering algorithm [22], and swarm clustering algorithm [23]. Chen et al. [24] proposed a fast clustering for large-scale data. Chel et al. [25] presented the HDBSCAN clustering algorithm to find a clustering pattern present in calcium spiking obtained by confocal imaging of single cells. Znidi et al. [26] introduced a new methodology for discovering the degree of coherency among buses using the correlation index of the voltage angle between each pair of buses and used the hierarchical density-based spatial clustering of applications with noise to partition the network into islands. Parmar et al. [27] proposed a residual error-based density peak clustering algorithm named REDPC to better handle datasets comprising various data distribution patterns. Specifically, REDPC adopted the residual error computation to measure the local density within a neighborhood region. Parmar et al. [28, 29] proposed the feasible residual error-based density peak clustering algorithm with the fragment merging strategy, where the local density within the neighborhood region was measured through the residual error computation and the resulting residual errors were then used to generate residual fragments for cluster formation.

Overall, the above methods have the limits of low clustering efficiency and time-consuming with high-dimensional data.

Considering that clusters in real-world datasets may have different sizes, shapes, and densities, accompanied by certain noises and outliers, this paper takes the idea of initial division and hierarchical aggregation to design a clustering algorithm named hierarchical DBSCAN (HDBSCAN). The proposed method comprises two stages of division and aggregation. Our contributions are as follows:

(1) First, it makes an initial division of the dataset based on sample densities; that is, the proposed method takes the neighbor information of samples to calculate local density values and then searches the set of density-connected samples for each unlabeled core point sequentially according to the density values in descending order, thus forming the initial clusters.

(2) Then, the method adopts the idea of hierarchical clustering to perform the aggregation of neighbor clusters. Based on the inner and border distances between clusters, the most similar clusters are regarded as neighbor clusters and merged to form a new cluster, and the process is repeated until the iteration ends.

(3) Based on the way of division and aggregation, the method can identify clusters with different forms in the dataset. Moreover, noise data cannot be integrated into high-density clusters as its density is relatively sparse, by which the proposed method can handle noise data reasonably.

The rest of this paper is organized as follows. Section 2 introduces two typical clustering algorithms, i.e., the DBSCAN clustering and the hierarchical clustering. Section 3 describes the proposed hierarchical DBSCAN algorithm in detail. Section 4 discusses the clustering performance of the proposed method, then applies it to the Chinese poverty dataset, and further analyzes the result of clustering. Finally, conclusions are presented in Section 5.

2. Theoretical Foundation

2.1. The DBSCAN Clustering. The DBSCAN algorithm regards regions with high densities as clusters and those with sparse densities as noises. It requires two hyperparameters, i.e., the neighborhood radius *eps* and the minimum number of samples required to form a dense region *minpts*.

Let $D = \{x_1, \ldots, x_n\}$ represent the dataset composed of n samples and d attributes, where $x_i = [x_{i1}, \ldots, x_{i\,d}]^{\mathrm{T}}$ denotes the i-th sample in the dataset. The *eps*-neighborhood of x_i is

$$N_{\mathrm{eps}}(x_i) = \{x_j \in D | \mathrm{dist}(x_i, x_j) \le \mathrm{eps}\}, \tag{1}$$

where $\mathrm{dist}(x_i, x_j)$ denotes the distance between samples x_i and x_j, calculated by

$$\mathrm{dist}(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_i - x_j)^2}. \tag{2}$$

If x_i satisfies equation (3), it is called the core point:

$$\left|N_{\mathrm{eps}}(x_i)\right| \ge \mathrm{minpts}. \tag{3}$$

There are several definitions in the DBSCAN algorithm, listed as follows:

(1) A sample x_j is directly reachable from x_i with respect to eps and minpts if x_i is a core sample and $x_j \in N_{\mathrm{eps}}(x_i)$

(2) A sample x_j is reachable from x_i with respect to eps and minpts if there exists a chain of samples $x_{m_1}, \ldots, x_{m_k}$, $(1 \le k, m_k \le n)$ with $x_{m_1} = x_i$ and $x_{m_k} = x_i$, where each $x_{m_{l+1}} (0 < l < k)$ is directly reachable from x_{m_l} with respect to eps and minpts

(3) A sample x_j is reachable from x_i with respect to eps and minpts if there exists a chain of samples $x_{m_1}, \ldots, x_{m_k}$, $(1 \le k, m_k \le n)$ with $x_{m_1} = x_i$ and $x_{m_k} = x_i$, where each $x_{m_{l+1}} (0 < l < k)$ is directly reachable from x_{m_l} with respect to eps and minpts

In the process of clustering, the algorithm randomly selects a core point as the initial point and takes all the core points in its eps-neighborhood for continuous expansion. The expansion ends until the maximal set of density-connected samples is found and labeled as one cluster. After that, the algorithm randomly chooses other unlabeled core points for generating new clusters. The process of clustering completes when all the core points are labeled.

2.2. Hierarchical Clustering. The hierarchical clustering can be divided into the agglomerative hierarchical clustering and the divisive hierarchical clustering. The agglomerative hierarchical clustering first takes each sample as a separate cluster, then finds the two closest clusters by measuring the distance between the clusters, and then merges them into a new cluster. Subsequently, the algorithm recalculates the distance between clusters and continues the aggregation process. The realization of the divisive hierarchical clustering is the exact opposite of the above, which regards the whole dataset as one cluster and then performs the division iteratively.

In the hierarchical clustering, the distance between C_p and C_q can be calculated by (4), i.e., the average of sample distances between two clusters. Besides, the minimum distance of samples between clusters shown in (5), or the maximum distance of samples between clusters, can also be taken to measure the distance of two clusters:

$$S_1(C_p, C_q) = \frac{1}{|C_p| \cdot |C_p|} \sum_{x_i \in C_p} \sum_{x_j \in C_q} \mathrm{dist}(x_i, x_j), \tag{4}$$

$$S_2(C_p, C_q) = \min\{\mathrm{dist}(x_i, x_j) \mid x_i \in C_p, x_j \in C_q\}. \tag{5}$$

2.3. The Hierarchical DBSCAN Algorithm. As the global hyperparameters for the DBSCAN algorithm, the numerical values of minpts and eps have a direct impact on the expansion of all the clusters. Figure 1 illustrates the expansion of clusters under different numerical values ofeps, where the red points denote the initial core points in each iteration of expansion. According to Figure 1(a), the clusters C_1 and C_2 can be identified while the other samples are regarded as noises and cannot be partitioned properly if the DBSCAN algorithm takes eps_1 as the neighborhood radius. It can be seen from Figure 1(b) that all the samples are divided into one cluster C_1 through four iterations of expansion if the algorithm takes eps_2 as the neighborhood radius.

In view of the above problem, this paper takes the way of division and aggregation to design the HDBSCAN clustering algorithm. First, the proposed method makes an initial division of the dataset according to sample densities. During the expansion of each cluster, the method adaptively adjusts the neighborhood radius based on the neighbor information of samples within the cluster. Then, the idea of hierarchical clustering is adopted to perform the recursive aggregation; that is, the method takes the cluster pair with the minimum distance as the neighbor clusters and then merges them into a new cluster. Based on division and aggregation, the method can perceive the clusters with different forms in the data space.

2.4. Initial Division. During the process of initial division, the parameter k is used to calculate the local density. Let $SN_k(x_i)$ represent the set composed of k samples closest to x_i, and the average distance between x_i and all samples in the set is

$$\overline{\text{dist}}(x_i) = \frac{1}{\left|SN_k(x_i)\right|} \sum_{x_j \in SN_k(x_i)} \text{dist}\left(x_i, x_j\right). \tag{6}$$

The distance $\overline{\text{dist}}(x_i)$ can capture the density distribution around the sample x_i. The smaller the value, the greater the density. Therefore, the local density of x_i can be defined as

$$\text{den}(x_i) = \overline{\text{dist}}(x_i)^{-1}. \tag{7}$$

The neighborhood radius of x_i, namely, $\text{eps}(x_i)$, is the distance between x_i and the maxpts-th nearest sample. The process of the initial division includes the following steps.

Step 1. Calculate the local density for each sample and then sort the samples based on the local density values so as to form the sequence:

$$O = \left\{x_{m_1}, \ldots, x_{m_n} \middle| \text{den}\left(x_{m_j}\right) > \text{den}\left(x_{m_{j+1}}\right), 1 \le j \le n-1, 1 \le m_j \le n\right\}. \tag{8}$$

The cluster label is initialed as $q = 1$.

Step 2. Select an unlabeled sample x_i from the sequence O in order and set the iteration number$t = 1$.

Step 3. Let $C_q^{(t)}$ and $Q_q^{(t)}$ represent the set of samples and the sequence of core points for the q-th cluster in the t-th iteration and $C_q^{(1)} = \{x_i\}, Q_q^{(1)} = \{x_i\}$.

Step 4. Calculate the adaptive neighborhood radius for the expansion of the current cluster by all samples in the cluster:

$$\text{eps}\left(C_q^{(t)}\right) = \frac{1}{\left|C_q^{(t)}\right|} \sum_{x_k \in C_q^{(t)}} \text{eps}(x_k). \tag{9}$$

Step 5. Select a core point x_j from the sequence $Q_q^{(t)}$ in order and continue the expansion based on $\text{eps}(C_q^{(t)})$.

Step 6. Calculate the set of neighbor samples to be expanded according to

$$\widehat{C}_q^{(t)} = \left\{x_k \middle| \text{dist}\left(x_j, x_k\right) \le \text{eps}\left(C_q^{(t)}\right) \text{and } x_k \notin C_p, p = 1, \ldots, q-1\right\}. \tag{10}$$

Step 7. Update $C_q^{(t+1)}$ and $Q_q^{(t+1)}$ by

$$C_q^{(t+1)} = C_q^{(t)} \cup \widehat{C}_q^{(t)}, \tag{11}$$

$$Q_q^{(t+1)} = Q_q^{(t)} \cup \left\{x_k \middle| x_k \in \widehat{C}_q^{(t)} \middle| N_{\text{eps}\left(C_q^{(t)}\right)}(x_k)\right| \ge \text{minpts}\} - \left\{x_j\right\}. \tag{12}$$

Step 8. The expansion of the q-th cluster C_q is completed if $Q_q^{(t+1)} = \varnothing$, then it returns to Step 9. Otherwise, it sets $t = t + 1$ and returns to Step 4.

Step 9. The initial division ends if all the samples are labeled. Otherwise, it sets the cluster label as $q = q + 1$ and returns to Step 2.

2.5. Aggregation of Neighbor Clusters. In this paper, the similarity between clusters is measured by border distance and inner distance. Figure 2 takes the clusters C_p' and C_q' during the aggregation as an example to describe two kinds of distances. In Figure 2, the red points denote the core points and the grey ones denote the border points distributed around the clusters.

Suppose that the dataset can be represented by $D = \{C_1, \ldots, C_K\}$ after the initial division, where K denotes the number of clusters and $C_i (i = 1, \ldots, K)$. While the neighbor clusters are merged to form new clusters continuously during the aggregation, C_p' is described by $C_p' = \left\{C_{m_1}, \ldots, C_{m_{K_p}}\right\}, 1 \le K_p, m_{K_p} \le K$. The set of border points in C_p' is

$$B\left(C_p'\right) = \left\{x_i \middle| x_i \in C_p' \text{ and } \left|N_{\text{eps}^{(t)}}(x_i)\right| < \text{minpts}\right\}, \tag{13}$$

where $\text{eps}^{(t)}$ denotes the neighborhood radius at the completion of division for x_i. The value changes dynamically due

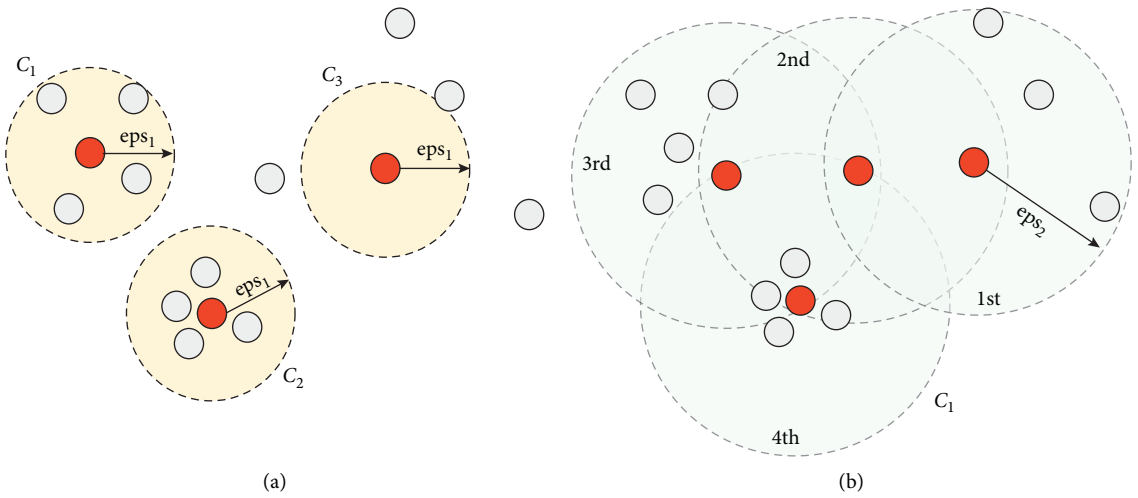


Figure 1: The expansion of clusters under different numerical values of eps: (a) expansion with eps_1; (b) expansion with eps_1.

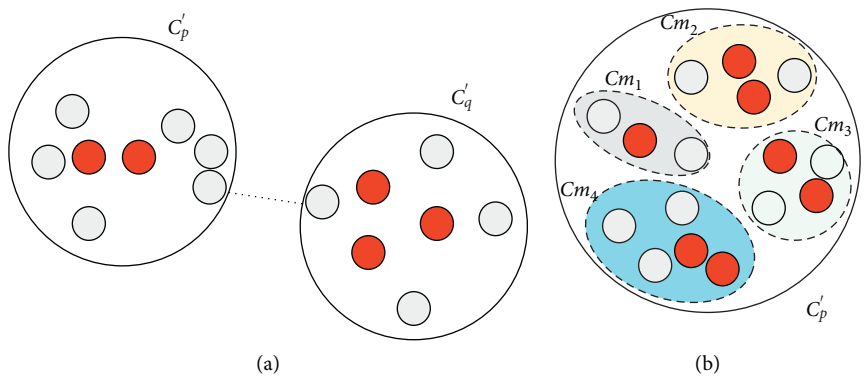


Figure 2: The border distance and the inner distance: (a) border distance; (b) inner distance.

to the adaptive adjustment of neighbor radius. According to Figure 2(a), the border distance between clusters C_p' and C_q' is the minimum distance between the border points of two clusters, namely,

$$O\left(C_p', C_q'\right) = \min\left\{\text{dist}\left(x_i, x_j\right) | x_i \in B\left(C_p'\right), x_j \in B\left(C_q'\right)\right\}. \tag{14}$$

As can be seen from Figure 2(b), the cluster C_p' consists of four initial clusters, and thus the inner distance of the cluster is defined as

$$I\left(C_p'\right) = \frac{2}{K_p\left(K_p - 1\right)} \sum_{C_{m_i} \in C_p'} \sum_{C_{m_j} \in C_p' C_{m_j} \neq C_{m_i}} O\left(C_{m_i}, C_{m_j}\right). \tag{15}$$

During the aggregation, the two clusters with the minimum border distance are considered as the neighbor clusters for further merging if their difference of inner distances and that of densities below certain limitations. Algorithm 1 is a simple implementation of aggregation for neighbor clusters. In the actual implementation of the algorithm, values such as border distances and inner distances will be restored to avoid repeated calculation. According to the 14th line of Algorithm 1, two clusters will be involved in calculating neighbor clusters only when their density difference, border distances, and inner distances satisfy certain conditions.

The proposed HDBSCAN clustering algorithm can capture clusters with different forms in the data space. The aggregation of neighbor clusters weakens the sensitivity of the algorithm to hyperparameters in the initial division. Besides, the result of the division in the DBSCAN algorithm depends on the selection sequence of initial core points. The proposed method can weaken the fluctuation caused by the selection sequence to some extent. The Algorithm 2 summarizes the whole process.

3. Experimental Results and Analysis

3.1. Experimental Design

3.1.1. Datasets. Three public artificial datasets and four real-world datasets are chosen to verify the effectiveness of the proposed clustering algorithm. The description of artificial datasets is listed in Table 1. The visualization of artificial datasets is shown in Figure 3.

Input: clusters after initial division $D = \{C_1, \ldots, C_K\}$; the threshold ε; $\{\text{den}(x_i) | i = 1, \ldots, n\}$
Output: final clusters after aggregation $D = \{C_1', \ldots, C_K'\}$
min_$O \longleftarrow +\infty$, combine$\longleftarrow$
While True
  Calculateaverage_den_diff by the averaged of density differences between clusters
  For each cluster C_p' in D
    For each cluster C_q' in $D - C_p'$
      Calculate $O(C_p', C_q'), I(C_p'), I(C_q')$
      Calculate $\text{den}(C_p'), \text{den}(C_q')$ by the averaged densities for samples in the clusters
den_diff $\longleftarrow |\text{den}(C_p') - \text{den}(C_q')| / \max(\text{den}(C_p'), \text{den}(C_q'))$
  tmp $= \max(O(C_p', C_q')), I(C_p'), I(C_q')$
dist_diff$\longleftarrow ((O(C_p', C_q') - \max(I(C_p'), I(C_q'))) / \text{tmp})$
    $O \longleftarrow O(C_p', C_q')$
    If den_diff < average_den_diff anddist_diff $< \varepsilon$ and $O <$ min_O
      min_$O \longleftarrow O$, combine $\longleftarrow \{C_p', C_q'\}$
    End For
  End For
  If combine $\neq \varnothing$
$D \longleftarrow D -$ combine[0] $-$ combine[1] $+ \{\text{combine}[0] \cup \text{combine}[1]\}$
    Else
      Break
End While

Algorithm 1: The aggregation of neighbor clusters.

Input: parameter k , clusters after initial division $D = \{C_1, \ldots, C_K\}$; the threshold ε;$\{\text{den}(x_i) | i = 1, \ldots, n\}$
Output: final clusters
min_$O \longleftarrow +\infty$, combine$\longleftarrow$
While True
  Calculate the local density
    $O = \left\{x_{m_1}, \ldots, x_{m_n} | \text{den}(x_{m_j}) > \text{den}(x_{m_{j+1}}), 1 \le j \le n-1, 1 \le m_j \le n\right\}$
  Select an unlabeled sample x_i from the sequence O
  For $C_q^{(1)} = \{x_i\}, Q_q^{(1)} = \{x_i\}$.
    Calculate the adaptive neighborhood radius
    Select a core point x_j from the sequence $Q_q^{(t)}$
    Calculate the set of neighbor samples
    For $Q_q^{(t+1)} = \varnothing$
    The expansion of the q-th cluster C_q is completed
    End For
  End For
  Calculateaverage_den_diff by the averaged of density differences between clusters
      If den_diff < average_den_diff anddist_diff $< \varepsilon$ and $O <$ min_O
        min_$O \longleftarrow O$, combine $\longleftarrow \{C_p', C_q'\}$
      End For
    End For
    If combine $\neq \varnothing$
$D \longleftarrow D -$ combine[0] $-$ combine[1] $+ \{\text{combine}[0] \cup \text{combine}[1]\}$
    Else
      Break
End While

Algorithm 2: The proposed cluster method.

The description of real-world datasets is listed in Table 2, where Banknote, Parkinson, Codon usage, HCV, and Planning relax are taken from UCI machine learning repository, and CFPS2016 is the dataset of poverty-stricken households in China. The CFPS2016 dataset comes from the China Family Panel Studies (CFPSs) released by the Institute of Social Science Survey of Peking University, China, in 2016. In the experiment, the CFPS2016 dataset consists of 14019 samples and 320 attributes, which covers the family economy as well as the states of adults and children in health,

Table 1: The description of artificial datasets.

Data sets	Size	Dimension	Cluster number
cluto-t5-8k	8000	2	6
cluto-t8-8k	8000	2	8
triangle2	8000	2	10

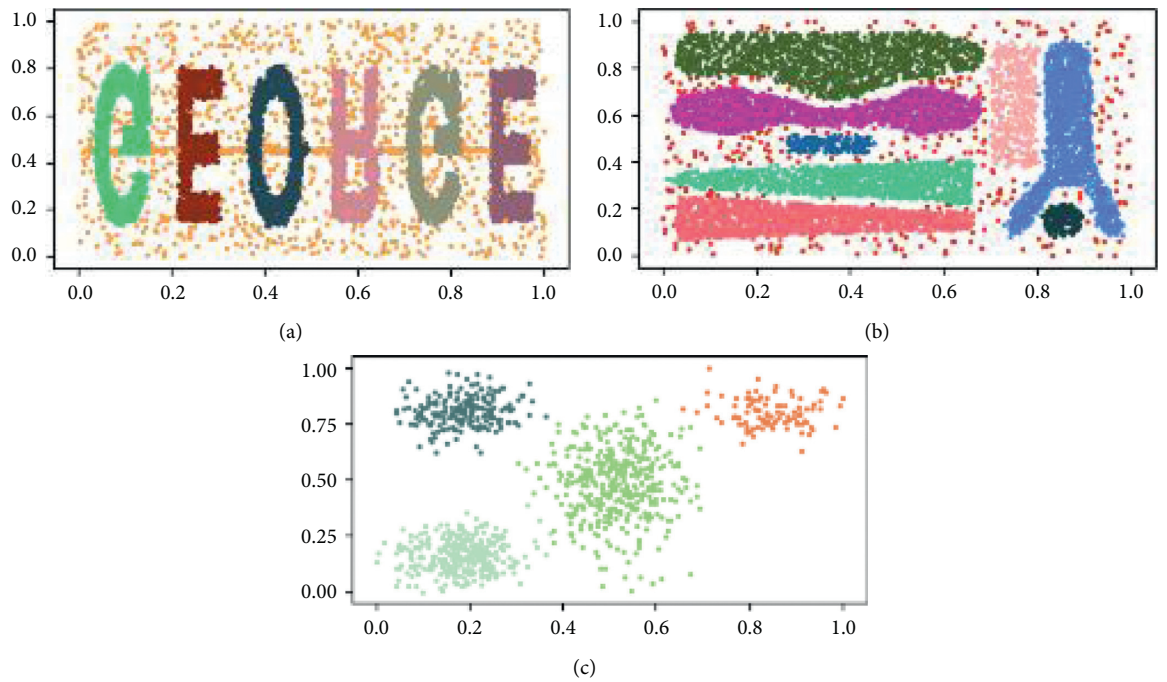


Figure 3: The visualization of three artificial datasets: (a) cluto-t5-8k, (b) cluto-t8-8k, and (c) triangle2.

Table 2: The description of real-world datasets.

Data sets	Records	Attributes
Banknote	1372	5
Parkinson	1040	26
Planning relax	182	13
Codon usage	13028	69
HCV	615	14
CFPS2016	1778	320

education, and psychology. Hence, the CFPS2016 dataset can reflect the status of each Chinese household objectively. During the data preprocessing, we fill in missing values with the K-nearest neighbor imputation method [30], and then 1778 poverty-stricken households are measured from 14019 Chinese households based on the Alkire–Foster method, the main measurement method of multidimensional poverty [31]. The parameters in this experiment are set the same as DBSCAN under the same experimental platform.

3.1.2. Evaluation Metrics. We take the silhouette coefficient (SC) [32], Davies–Bouldin index (DBI) [33], adjusted Rand index (ARI), and normalized mutual information (NMI) [34] to measure the performance of clustering. The silhouette coefficient is defined by

$$\mathrm{SC} = \frac{1}{n}\sum_{i=1}^{n}\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, \tag{16}$$

where n denotes the total number of samples; $a(i)$ denotes the average distance between the sample x_i and all other samples in its cluster, which reflects the cohesiveness of clustering; and $b(i)$ denotes the minimum value of average distances between the sample x_i and all samples in any other cluster, which reflects the dispersity of clustering. The larger SC represents the higher performance of clustering. Besides, the definition of the Davies–Bouldin index is

$$\mathrm{DBI} = \frac{1}{K'}\sum_{i,j=1}^{K'}\max_{i\neq j}\left(\frac{\overline{S}_i-\overline{S}_j}{\left\|w_i-w_j\right\|_2}\right), \tag{17}$$

where K' denotes the number of clusters; $\overline{S}_i$ and $\overline{S}_j$ denote the average distance between all the samples within the cluster and the centroid of the cluster; $\|w_i - w_j\|_2$ denotes the distance between cluster centroids. The smaller DBI denotes the higher performance of clustering.

With respect to performance, adjusted Rand index (ARI) and normalized mutual information (NMI) are also used for evaluation. ARI represents the similarity measure between two clusterings that is adjusted for chance and is related to accuracy, while NMI quantifies the amount of information obtained about one clustering, through the other clustering (i.e., the mutual dependence between the two). In the case of observations being identified as noise, each noise observation is treated as a distinct singleton cluster for both ARI and NMI.

3.1.3. Compared Methods. This paper compares the proposed method with three existing clustering algorithms which are described as follows:

(1) AHC: as described in Section 2.2, the method regards every sample as a separate cluster and then merges the two closest clusters continuously until the iteration ends.

(2) DBSCAN: as described in Section 2.1, the method performs the continuous expansion for each cluster based on core points and thus takes regions with high densities as clusters and those with low-densities as noises.

(3) EDBSCAN: the method calculates the density variation for each core points and specified that a core point is allowed to expand only when its density variation is below a specified threshold and its neighborhood satisfies the homogeneity index [35].

(4) NS-DBSCAN: the NS-DBSCAN algorithm used a strategy similar to the DBSCAN algorithm. Furthermore, it provided a new technique for visualizing the density distribution and indicating the intrinsic clustering structure [36].

(5) ADBSCAN: unlike many other algorithms that estimate the density of each samples using different kinds of density estimators and then choose core samples based on a threshold, ADBSCAN utilized the inherent properties of the nearest neighbor graph [37].

4. Results and Analysis

4.1. Artificial Datasets and Real-World Datasets from UCI. First, we conduct the effect experiments of ε on the local sensitivity as shown in Figure 4. Then, the selected ε is used for the following experiments to provide the equitable comparison. From Figure 4, we can know that when ε is 0.5, the local sensitivity is small. The effect of proposed method is better. Therefore, we select $\varepsilon = 0.5$ in this paper.

The clustering results of three artificial datasets based on the proposed method are shown in Figure 5, where regions with different colors can be regarded as one cluster. According to Figures 5(a), 5(c), and 5(e), the datasets are cut into several regions with different densities after the initial division. As can be seen from Figures 5(b), 5(d), and 5(f), the adjacent regions with similar densities aggregate continuously during the aggregation of neighbor clusters, which contributes to the ideal results of clustering. In Figure 5(f), some discrete points are distributed around four large clusters. The proposed method identifies these points as noises since there exist certain differences between the densities of discrete points and those of clusters around them.

The metric values for three UCI datasets obtained by four comparison methods are shown in Table 3, in which the optimal results have been bolded and the suboptimal results have been italicized.

According to Table 2, all the SC values obtained by the proposed method HDBSCAN are better than those obtained by other methods, and the method also has ideal DBI values. For instance, in respect of the Parkinson dataset, the SC value of HDBSCAN is 8.91% higher than that of the suboptimal method AHC. Although the DBI value of HDBSCAN is suboptimal, it is only 2.63% worse than that of EDBSCAN. The above results indicate that the proposed method HDBSCAN has the ideal performance of clustering. Table 2 shows the ARI performance with the different methods on the artificial datasets. From these results, HDBSCAN is shown to rank first in these datasets. More importantly, in each case HDBSCAN is able to identify the underlying classes of each dataset, whereas each of the other approaches fails at this task in at least one case.

4.2. The Dataset of Poverty-Stricken Households in China. We perform clustering on 1778 poverty-stricken households of CFPS2016 so as to identify different categories of poverty-stricken households. Table 4 shows the metric values for CFPS2016 obtained by four compared methods, where the optimal results have been bolded and the suboptimal results have been italicized. Table 4 also shows NMI performance results on the same set of artificial datasets and clustering approaches. Here, HDBSCAN ranking performance is identical to those discussed with respect to ARI.

We also make accuracy comparison with the other three methods. The results are the average values shown in Table 5.

It can be seen from Table 5 that the values of SC and DBI obtained by HDBSCAN are better than those obtained by other compared methods. Therefore, the proposed method has the ideal performance of clustering on the CFPS2016 dataset. The clustering result based on HDBSCAN is listed in Table 6.

According to Table 6, the proposed method divides CFPS2016 into 10 clusters and identifies 70 noises. Additionally, the numbers of households within different clusters are distributed unevenly. For instance, the number of households in Cluster 1 is 382 while those in Cluster 9 and Cluster 10 are 61 and 34, respectively. To evaluate the rationality of the clustering result, we adopt the random forest algorithm to calculate the importances of attributes in ten

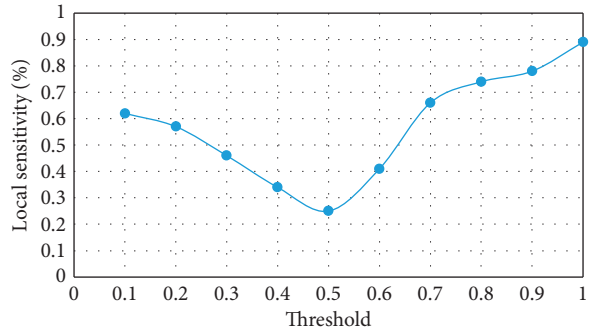


Figure 4: The relation between local sensitivity and threshold ε.

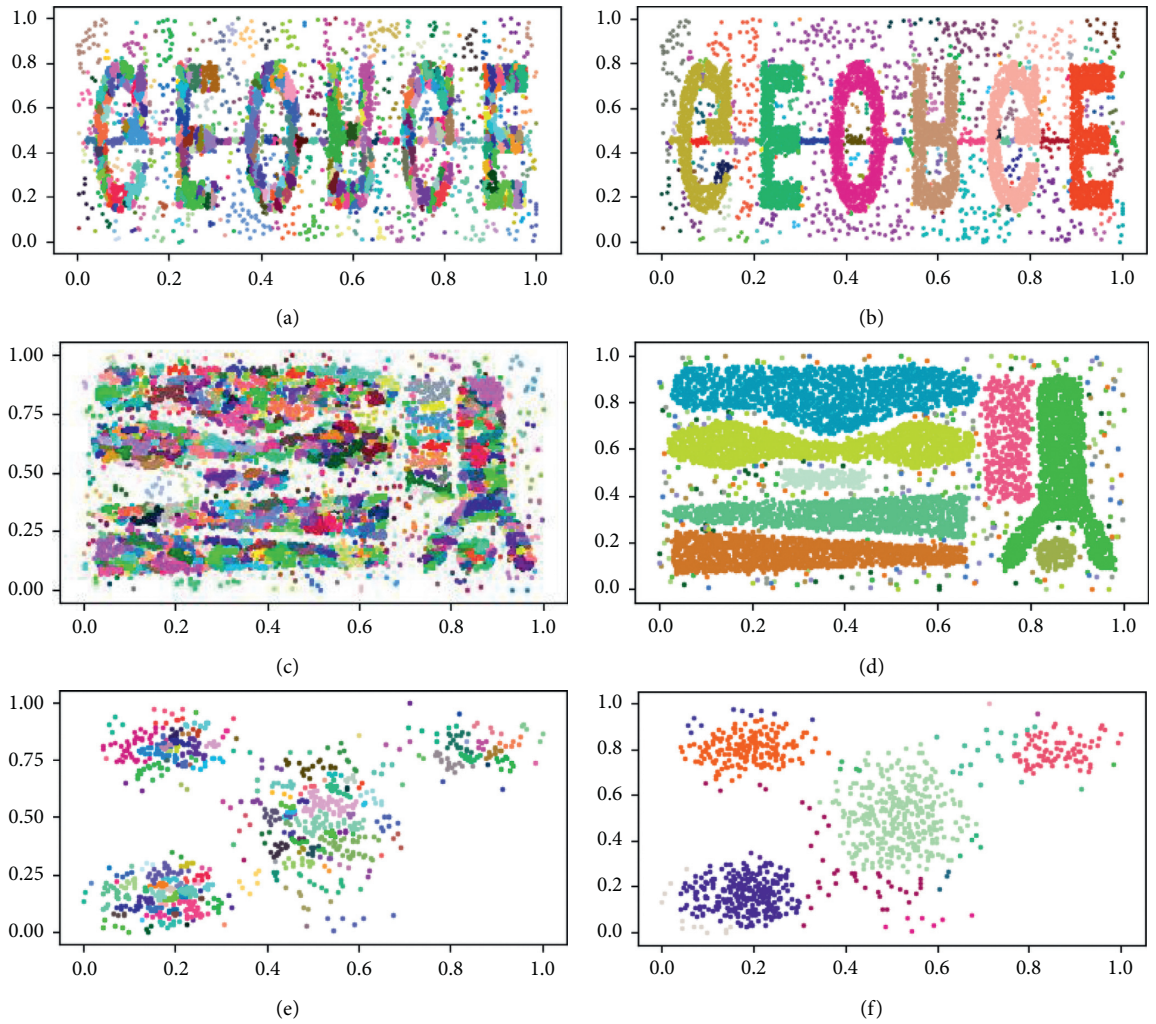


Figure 5: Clustering results on artificial datasets based on HDBSCAN. (a) Initial division for cluto-t5-8k. (b) Aggregation for cluto-t5-8k. (c) Initial division for cluto-t8-8k. (d) Aggregation for cluto-t8-8k. (e) Initial division for triangle2. (f) Aggregation for triangle2.

clusters and thus analyze the characteristics of each cluster. Specifically, based on the labels generated by HDBSCAN clustering, we take each cluster as the positive class and the other clusters as the negative class to construct multiple binary classification models, thereby mining the important attributes within each cluster.

Based on the important attributes within clusters, the characteristics of Cluster 1 are listed below. (1) The household has no children under the age of 16. (2) The annual net income of the household is higher than the average level. (3) Medical expenses are more prominent in the expenditure of the household. The characteristics of

Table 3: The metric values on four UCI datasets.

Dataset	Metric	AHC	DBSCAN	EDBSCAN	NS-DBSCAN	ADBSCAN	HDBSCAN
Banknote	SC	0.308	0.321	0.481	0.483	*0.484*	**0.485**
	DBI	1.239	1.492	1.226	*1.197*	1.875	**1.155**
	ARI	0.796	0.753	0.865	0.912	*0.987*	**0.991**
	NMI	0.758	0.865	0.897	0.953	*0.992*	**0.996**
Planning relax	SC	0.178	0.136	0.171	0.215	*0.231*	**0.271**
	DBI	2.355	11.834	**1.733**	2.322	*2.095*	2.368
	ARI	0.612	0.537	0.712	0.739	*0.865*	**0.898**
	NMI	0.821	0.854	0.882	0.913	*0.924*	**0.928**
Parkinson	SC	0.258	0.201	0.212	0.255	*0.276*	**0.281**
	DBI	1.731	1.785	**1.598**	1.679	1.717	*1.640*
	ARI	0.635	0.689	0.728	0.825	*0.877*	**0.969**
	NMI	0.589	0.721	0.737	0.862	*0.883*	**0.928**
Codon usage	SC	0.265	0.238	0.275	0.281	*0.283*	**0.296**
	DBI	3.856	8.954	2.917	2.805	*2.655*	**2.192**
	ARI	0.712	0.884	0.865	0.928	**0.944**	*0.939*
	NMI	0.822	0.851	0.874	0.912	*0.953*	**0.967**
HCV	SC	0.328	0.337	0.416	0.397	*0.419*	**0.511**
	DBI	2.754	2.663	2.841	*2.425*	2.538	**2.331**
	ARI	0.714	0.821	0.885	0.836	*0.918*	**0.986**
	NMI	0.689	0.774	0.796	0.825	*0.884*	**0.917**

Table 4: The metric values on the CFPS2016 datasets.

Metric	AHC	DBSCAN	EDBSCAN	NS-DBSCAN	ADBSCAN	HDBSCAN
SC	0.099	0.109	0.138	0.158	*0.177*	**0.237**
DBI	2.656	2.019	1.977	1.765	*1.528*	**1.003**
ARI	0.713	0.852	0.872	0.893	*0.948*	**0.995**
NMI	0.788	0.854	0.861	0.875	*0.942*	**0.993**

Table 5: The accuracy values on the CFPS2016 datasets.

Metric	AHC	DBSCAN	EDBSCAN	NS-DBSCAN	ADBSCAN	HDBSCAN
Accuracy	79.6%	82.7%	85.3%	85.8%	*87.1%*	**89.6%**

Table 6: The result of clustering based on HDBSCAN.

Label	1	2	3	4	5	6
Number	382	369	282	148	127	126
Label	7	8	9	10	Noise	
Number	90	89	61	34	70	

Cluster 9 are as follows: (1) the average age of adults in the household is 76. (2) Almost every household member has no pension insurance. Besides, the characteristics of Cluster 10 are as follows: (1) the annual per capita income of the household is 35, 914 yuan, 1.43 times higher than the average level. (2) More than half of the members use computers. The living standard of households in Cluster 10 is relatively high compared with other clusters, and Cluster 10 accounts for a small proportion of poverty-stricken households. According to the above analysis, the causes of poverty and characteristics for most households are similar so that the numbers of households in some clusters are large whereas the characteristics of a few poverty-stricken households are clearly different from others, which leads to small numbers of households in clusters such as Cluster 9 and Cluster 10.

Figure 6 shows the distribution of attribute importances in each cluster, where the abscissa values indicate the numbers of 320 attributes and the ordinate values indicate the attribute importances; the ten curves represent the distribution of attribute importances in ten clusters.

As can be seen from Figure 6, the distributions of attribute importances represented by ten curves nearly differ from each other. For instance, the attribute with the highest importance in Cluster 7 is the 165th-dimensional attribute which denotes the stage of schooling for household members at the last survey. And that in Cluster 8 is the 218th-dimensional attribute which denotes the total post-tax annual income from work. The phenomenon shows that poverty-stricken households within different categories differ in the characteristics and the causes of

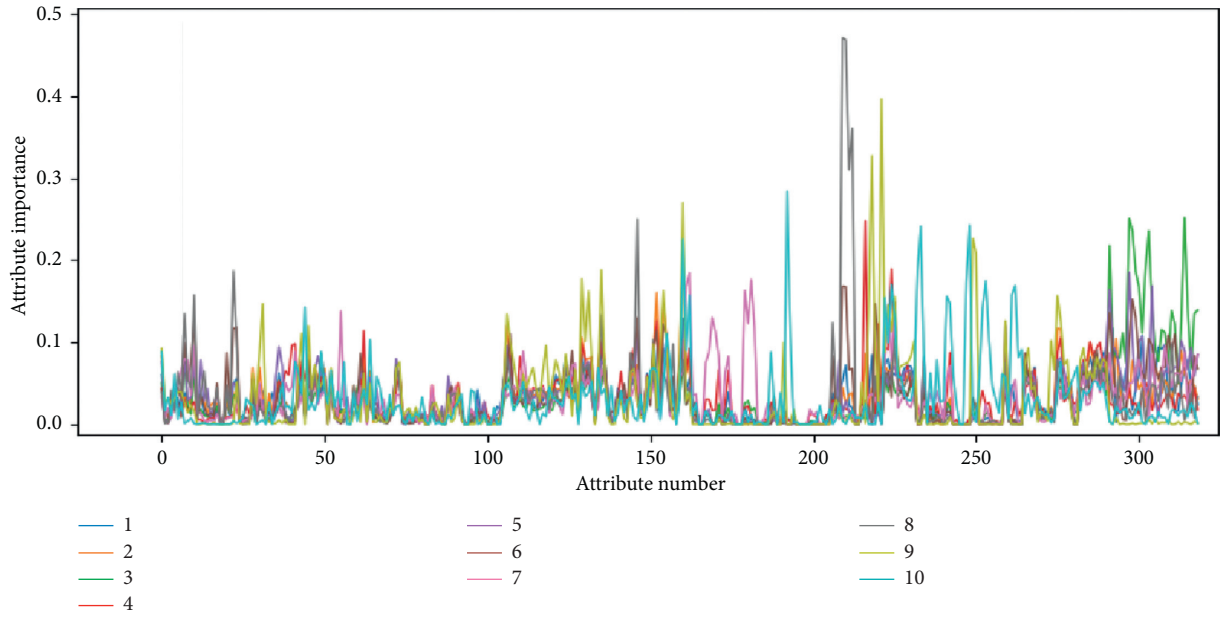


Figure 6: The attribute importances based on the clustering result by HDBSCAN.

Table 7: The metric values on four UCI datasets.

Dataset	AHC	DBSCAN	EDBSCAN	NS-DBSCAN	ADBSCAN	HDBSCAN
Banknote	*2.7*	**2.6**	2.8	2.9	2.8	3.1
Planning relax	*2.3*	**2.1**	2.5	2.6	2.6	*2.3*
Parkinson	*2.5*	**2.4**	2.7	2.6	2.7	*2.5*
Codon usage	2.9	**2.7**	2.9	2.8	2.9	*2.8*
HCV	2.6	**2.4**	2.7	2.8	2.6	*2.5*
CFPS2016	2.8	**2.4**	2.6	2.7	2.7	*2.5*

poverty. Therefore, the proposed method can identify the commonalities and differences in poverty effectively. Finally, for all the datasets, we conduct computational complexity experiments with the different methods. The results are shown in Table 7. Because the proposed method is the hierarchical DBSCAN algorithm based on the initial division and aggregation of neighbor clusters, the time is higher than traditional DBSCAN. However, the time is lower than other new methods.

5. Conclusions

This paper designs the hierarchical DBSCAN algorithm based on the initial division and aggregation of neighbor clusters. First, the proposed method HDBSCAN adopts the adaptive neighborhood radius to perceive regions with different densities and thus makes the initial division of the dataset. Then, iterative aggregation is performed on neighbor clusters according to the border and inner distances. Experiments on artificial datasets and UCI real-world datasets indicate that HDBSCAN has the ideal performance of clustering. Additionally, HDBSCAN divides the dataset of Chinese poverty-stricken household, namely, CFPS2016, into 10 clusters, and experimental results verify the rationality of the clustering result. The main reasons for the ideal performance of HDBSCAN lie in the following two aspects. First, the adaptive neighborhood radius helps to identify regions of different densities in the data space with imbalanced density distribution. Second, the aggregation further merges neighbor clusters with similar densities, which weakens the impact of the accuracy of initial partition on the clustering performance effectively. However, if the dimension of the datasets is very higher, the cluster effect is not better. In the future, more research studies will be conducted on the clustering result of the CFPS2016 dataset. To be specific, we will study the characteristics of poverty-stricken households in each category so as to support the formulation and implementation of antipoverty measures. The advanced clustering technology will be applied in targeted poverty alleviation of the poverty counties in China.

Data Availability

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

Conflicts of Interest

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

References

[1] H. Hsieh, S. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *Proceedings of the 21th ACM SIGKDD International Conference*, Sydney, Australia, August 2015.

[2] 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.

[3] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "Incremental deep computation model for wireless big data feature learning," *IEEE Transactions on Big Data*, vol. 6, no. 2, p. 248, 2020.

[4] Z. Lv, L. Qiao, K. Cai, and Q. Wang, "Big data analysis technology for electric vehicle networks in smart cities," *IEEE Transactions on Intelligent Transportation Systems*, p. 1, 2020.

[5] J. Chen, Z. Lv, and H. Song, "Design of personnel big data management system based on blockchain," *Future Generation Computer Systems*, vol. 101, pp. 1122–1129, 2019.

[6] Z. Lv, X. Li, H. Lv, and W. Xiu, "BIM big data storage in WebVRGIS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2566–2573, 2020.

[7] J. E. Blumenstock, "Fighting poverty with data," *Science*, vol. 353, no. 6301, pp. 753-754, 2016.

[8] Y. Zhou, Y. Guo, Y. Liu, W. Wu, and Y. Li, "Targeted poverty alleviation and land policy innovation: some practice and policy implications from China," *Land Use Policy*, vol. 74, pp. 53–65, 2018.

[9] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Vol. 344, John Wiley & Sons, Hoboken, NJ, USA, 2009.

[10] Y. Xie and J. Hu, "An introduction to the China family panel studies (CFPS)," *Chinese Sociological Review*, vol. 47, no. 1, pp. 3–29, 2014.

[11] A. Saxena, M. Prasad, A. Gupta, N. Bharill, and O. P. Patel, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[12] H. Tiwari, "Clustering algorithm and its application in data mining," *Wireless Personal Communications*, vol. 110, no. 1, pp. 21–30, 2020.

[13] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu, "Hierarchical clustering," *Journal of the ACM*, vol. 66, no. 4, pp. 1–42, 2019.

[14] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "DBSCAN: past, present and future," in *Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, pp. 232–238, Chennai, India, February 2014.

[15] X. Yu, D. Zhou, and Y. Zhou, "A new clustering algorithm based on distance and density," in *Proceedings of 2005 International Conference on Services Systems and Services Management*, vol. 2, pp. 1016–1021, Chongqing, China, June 2005.

[16] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A density based algorithm for discovering density varied clusters in large spatial databases," *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1–4, 2010.

[17] P. Liu, D. Zhou, and N. Wu, "VDBSCAN: varied density based spatial clustering of applications with noise," in *Proceedings of IEEE International Conference on Service Systems and Service Management*, pp. 1–4, Chengdu, China, June 2007.

[18] A. Fahim, A.-E. Salem, F. Torkey, M. Ramadan, and G. Saake, "Scalable varied density clustering algorithm for large datasets," *Journal of Software Engineering and Applications*, vol. 3, no. 6, pp. 593–602, 2010.

[19] M. N. Gaonkar and K. Sawant, "AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset," *International Journal on Advanced Computer Theory and Engineering*, vol. 2, no. 2, pp. 11–16, 2013.

[20] A. Ram, A. Sharma, A. S. Jalal, A. Agrawal, and R. Singh, "An enhanced density based spatial clustering of applications with noise," in *Proceedings of the 2009 IEEE International Advance Computing Conference*, pp. 1475–1478, Patiala, India, March 2009.

[21] A. Kouhi, H. Seyedarabi, and A. Aghagolzadeh, "Robust FCM clustering algorithm with combined spatial constraint and membership matrix local information for brain MRI segmentation," *Expert Systems with Applications*, vol. 146, 2019.

[22] D. Fan, Z. Song, S. Jon et al., "An improved quantum clustering algorithm with weighted distance based on PSO and research on the prediction of electrical power demand," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 2, pp. 1–9, 2020.

[23] M. Xu, J. Zhou, and P. Zhu, "An electronic nose system for the monitoring of water cane shoots quality with swarm clustering algorithm," *Journal of Food Safety*, vol. 41, no. 1, Article ID e12860, 2020.

[24] Y. Chen, L. Zhou, N. Bouguila et al., "BLOCK-DBSCAN: fast clustering for large scale data," *Pattern Recognition*, vol. 109, 2021.

[25] S. Chel, S. Gare, and L. Giri, "Detection of specific templates in calcium spiking in HeLa cells using hierarchical DBSCAN: clustering and visualization of CellDrug interaction at multiple doses∗," in *Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2425–2428, Montreal, Canada, July 2020.

[26] F. Znidi, H. Davarikia, M. Arani, and M. Barati, "Coherency detection and network partitioning based on hierarchical DBSCAN," in *Proceedings of the 2020 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–5, College Station, TX, USA, August 2020.

[27] M. Parmar, D. Wang, X. Zhang et al., "REDPC: a residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, 2019.

[28] M. D. Parmar, W. Pang, D. Hao et al., "FREDPC: a feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, pp. 89789–89804, 2019.

[29] M. Parmar, D. Wang, A. Tan, C. Miao, J. Jiang, and Y. Zhou, "A novel density peak clustering algorithm based on squared residual error," in *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp. 43–48, Shenzhen, China, November 2017.

[30] G. E. Batista and M. C. Monard, "A study of k-nearest neighbour as an imputation method," *Second International Conference on Hybrid Intelligent Systems*, vol. 87, pp. 251–260, 2002.

[31] S. Alkire and J. Foster, "Understandings and misunderstandings of multidimensional poverty measurement," *The Journal of Economic Inequality*, vol. 9, no. 2, pp. 289–314, 2011.

[32] P. J. Rousseeuw and " Silhouettes, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, 1987.

[33] A. Bryant and K. Cios, "RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density

estimates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1109–1121, 2018.

[34] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, Menlo Park, CA, USA, June 1996.

[36] T. Wang, C. Ren, Y. Luo, and J. Tian, "NS-DBSCAN: a density-based clustering algorithm in network space," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, p. 218, 2019.

[37] H. Li, X. Liu, T. Li et al., "A novel density-based clustering algorithm using nearest neighbor graph," *Pattern Recognition*, vol. 102, 2020.

Hindawi
Scientific Programming
Volume 2021, Article ID 6699313, 9 pages
https://doi.org/10.1155/2021/6699313

Research Article

Sensors Anomaly Detection of Industrial Internet of Things Based on Isolated Forest Algorithm and Data Compression

Desheng Liu, Hang Zhen, Dequan Kong, Xiaowei Chen, Lei Zhang, Mingrun Yuan, and Hui Wang

College of Information and Electronic Technology, Jiamusi University, Jiamusi 154007, China

Correspondence should be addressed to Dequan Kong; 893907285@qq.com and Hui Wang; 3120205463@bit.edu.cn

Received 27 October 2020; Revised 20 December 2020; Accepted 15 January 2021; Published 31 January 2021

Academic Editor: Liang Zhao

Aiming at solving network delay caused by large chunks of data in industrial Internet of Things, a data compression algorithm based on edge computing is creatively put forward in this paper. The data collected by sensors need to be handled in advance and are then processed by different single packet quantity K and error threshold e for multiple groups of comparative experiments, which greatly reduces the amount of data transmission under the premise of ensuring the instantaneity and effectiveness of data. On the basis of compression processing, an outlier detection algorithm based on isolated forest is proposed, which can accurately identify the anomaly caused by gradual change and sudden change and control and adjust the action of equipment, in order to meet the control requirement. As is shown by experimental simulation, the isolated forest algorithm based on partition outperforms box graph and K-means clustering algorithm based on distance in anomaly detection, which verifies the feasibility and advantages of the former in data compression and detection accuracy.

1. Introduction

With the rapid development and integration of the Internet of Things (IoT) and cloud computing technology, we have gradually entered the era of "Internet of Things, comprehensive perception" [1]. At the same time, a large number of sensor devices are widely used in various fields including biomedicine, petrochemical, public transportation, environmental protection, electric power, and industrial manufacturing. In spite of the excitement, IoT sensor-based technology still faces great challenges and uncertainties in its authenticity, timeliness, reliability, and security. With the extensive use of sensor devices, people's lifestyle changes a lot; meanwhile, massive time series data are generated during the process of application. According to the Internet Data Center (IDC) [2], by 2020, the global .data are expected to exceed 40 zb. Boeing 787 generates more than 5 GB of data per second, and the bandwidth between the aircraft and the satellite is not enough to support real-time transmission [3]. In order to capture road information in real time, sensors and cameras mounted on unmanned vehicles will generate about 1 GB of data per second. According to IHS, by 2035, there will be 54 million driverless vehicles in the world [4].

Usually, sensors collect data at a certain frequency and send the data to the cloud. The cloud then receives the observed data in strict-time sequence. These data known as "time series data" accurately record the real-time changes of certain parameters at some point, such as speed, power, and temperature. They can reflect the regulation of data changes under certain parameters, which is the premise of subsequent data analysis and mining. In practical scenarios, there are always some abnormal data that deviate from the normal perception in the process of data acquisition and transmission; thus, it is very difficult to obtain high-quality data through sensors. Furthermore, the occurrence of faults is always unpredictable. Nowadays, most of the anomaly detection algorithms are based on statistics, clustering, similarity measurement, constraint rules, and neural network

[5–9]. Statistical methods usually know the distribution of the sequence. By maintaining the sliding window and calculating the statistical characteristic indexes, abnormal parts can be detected accurately. This method is suitable for detecting discrete and abrupt value anomalies in the sequence, but it is difficult to effectively identify the continuous abnormal sequence interval. The clustering method quantifies the distance between outlier and normal cluster to judge outliers. Computational complexities of different clustering models vary tremendously, and the detection results depend on the quality of clustering. The method based on similarity measure can judge whether there is abnormal data by calculating the similarity between the standardized sequences. However, this method can take a long time. In the rule-based method, researchers have proposed sequence dependence and speed constraint, which can effectively use the characteristics in time series to repair highly abnormal data. However, this method can hardly meet the needs of sequence anomaly detection with variable patterns [10]. Yu et al. [11] proposed the framework of IoT monitoring system based on edge computing, and an anomaly detection approach using self-encoding neural network. According to the particularity of time series data and the difference of data composition, literature [12] proposes that, in the field of time series data, most anomaly detection methods are based on pattern recognition and clustering. In [13], a new anomaly detection algorithm for time series data is put forward, constructing a distributed recursive computing strategy and k-nearest neighbor fast selection strategy. Qi et al. [14] proposed a real-time anomaly detection algorithm for sensing data based on edge computing. By analyzing the continuity and correlation between sensing data in the form of time series, the algorithm establishes a distributed anomaly detection model of sensing data based on edge computing so as to effectively detect anomalies in real-time sensing data. At present, most of the existing time series anomaly detection methods focus on the abnormal recognition of single dimension periodic or simple pattern time series. A lot of misjudgments and omissions may occur in the process, which leads to the performance degradation of anomaly detection methods. Although various anomaly detection methods have been proposed in the literature, it is still difficult to accurately detect the abnormal data and patterns for the one-dimensional time series with variable patterns.

In addition, nowadays the IoT data is processed in the cloud, and cloud computing can provide an efficient computing platform for big data processing. However, with the growth rate of network bandwidth lagging well behind that of data, data transmission delay and energy consumption of cloud data center have increased significantly, which lead to the bottleneck of cloud computing. As a new computing mode, the core of edge computing is to migrate the decomposed computing tasks to the edge nodes for processing, so as to realize the preprocessing of data before entering the cloud server, and to reduce the computational load of cloud computing data center. It has been applied in many fields, such as online shopping, smart home, smart city, intelligent transportation, security monitoring, etc. [1, 15]. In order to provide a better computing platform for the Internet of Things, a cloud computing center with strong computing power and mass storage, this paper proposes an edge collaborative cloud architecture with the help of edge devices processing massive data and private data in edge computing. On this basis, an algorithm of data compression and anomaly detection based on edge computing comes into being. The data collected by sensors are preprocessed to reduce the amount of data transmission, so as to greatly reduce the cloud computing load. Analyzing from the perspective of time series data, anomalies in the sensor data can be effectively identified, and the normal data fluctuation in the sensor data is entirely retained.

The main content of this paper is as follows: Section 2 outlines the application of edge computing in the IoT and the advanced algorithms in sensor outlier detection. Sections 3 and 4 describe the basic principle and structure of isolated forest algorithm. In Section 5, the evaluation indicators of compression algorithm and anomaly detection algorithm are discussed, and the performance of the algorithm is evaluated experimentally using actual data. In Section 6, the whole research idea is summarized.

2. Related Work

2.1. Internet of Things. The Internet of Things, also known as the "Internet connecting goods," is an outstanding practical result of information network development during the third revolution of science and technology. IoT has now penetrated into various fields, including transportation, public safety, environmental protection, electric power, smart home, and medical health, and has received widespread attention from all walks of life. The Internet of Things refers to the connection of any object with the network through the information sensing equipment according to the agreed protocol. The objects exchange information through the media, so as to realize intelligent identification, positioning, tracking, supervision, and other functions. The Internet of Things, as the name suggests, is developed on the basis of the Internet. Put simply, it is an extension of the Internet. The information exchange and sharing of client extend the communication between things.. The Internet of Things is formed when everything is connected at any time, in any place, and between anyone.

Compared with the Internet, the Internet of Things covers a wider range. It does not necessarily require direct participation of people. Problems of objects are analyzed and managed by artificial intelligence. It contains a large number of sensor applications. Sensor is the source of massive data in the Internet of Things, which is more abundant in data types and processing diversification. It mainly uses wireless technology to connect. It can carry out real-time information interaction and data transmission, as well as information processing. It can integrate the storage, processing and analysis capabilities of things at one end of things, real-

time data processing, and feedback to improve user response efficiency and user experience [16].

With the development of information and communication technology, many items and devices can be connected to the network, for example, articles carrying radio frequency identification code, and most devices in industrial control, environmental control, and traffic control. Therefore, the IoT technology can make things more intelligent. The application of Internet of Things has covered the whole Internet field. The IoT architecture can be divided into perception layer, network layer, and application layer. The perception layer is the source of data and the source of identifying objects and collecting information in the Internet on Things. Mainly composed of a large number of sensors, RFID tags, cameras, and other sensing sensors, it is the basic layer supporting the whole IoT system. The network layer is the center of the Internet of Things, which is responsible for data transmission. It connects the application layer and the perception layer and realizes the relationship between things by wireless communication through the exchange equipment and transmission equipment. In this way, the user terminals distributed in different locations are connected to form a complete information transmission path. The application layer is a direct user-oriented interface, through which users interact with objects [17].

With the rapid development of IoT technology, a series of national strategies, including Made in China 2025, Advanced Manufacturing Partner program of the United States, and German Industry 4.0, are put forward and implemented. The Industrial Internet of Things (IIoT) emerges as the times require and has become an important driver of the intelligent transformation of global industrial system (originated from China Institute of Electronic Technology Standardization). IIoT, a cutting-edge industry of huge commercial value, is widely used in design, production, management, and service [16]. IIoT realizes flexible allocation of raw materials, execution of manufacturing process on demand, reasonable optimization of production process through network interconnection and rapid adaptation to the manufacturing environment, and data exchange and system interoperability of industrial resources to achieve efficient utilization of resources, in order to build a new service-driven industrial ecosystem [18, 19]. The Internet of Things (IoT) is equivalent to information about physical objects (sensors, machines, cars, buildings, and other objects), which makes possible the interaction and cooperation between these objects to achieve common goals. It helps realize remote monitoring and intelligent maintenance application scenarios of industrial equipment, and remote monitoring, preventive maintenance, and performance optimization analysis of equipment [20]. The so-called IIoT is an advanced production mode that uses cloud platform to upgrade traditional industry to intelligent industry.

2.2. Edge Calculation. As a key technology to realize the Internet of Things, edge computing is widely used in many fields, such as smart city, intelligent manufacturing, intelligent transportation, smart home, privacy protection [21], disaster relief [22–25], etc. In the aspect of smart city, edge computing can meet three requirements of large data volume, low latency, and real-time location identification in the construction of smart city. It can efficiently process the massive data in various fields including public safety, health data, public facilities, and transportation information. It can reduce the time for data transmission and process the private data of users and relevant institutions more safely. In the aspect of intelligent manufacturing, edge computing can effectively realize the interaction and cooperation of information in each part of the intelligent manufacturing system and ensure the real-time data processing in the intelligent process. It can upload the processing results to the cloud for compensation calculation and then download them to the controller for operation, so as to reduce the communication cost and improve the processing efficiency. In the aspect of intelligent transportation, the system analyzes the data collected by cameras and sensors in real time through edge calculation and makes corresponding decisions, which can solve bandwidth waste and delay, improve security of intelligent transportation, extend the applicability of it, and provide a better user experience. In the aspect of smart home, the edge computing system runs on the edge gateway inside the home, integrating smart home devices into the system. And the data generated by the devices can be processed and desensitized locally, which can effectively reduce the data transmission delay and better protect the privacy of users. In the aspect of disaster rescue, the key of intelligent fire protection is to process, analyze, and predict the data obtained from multiple data sources, and effectively transmit the results to rescuers, which require high computing power and timely response. Through edge computing, the data can be transmitted to the base station through the edge equipment and then to the cloud without infrastructure. In transmission, the edge computing and storage resources will be used nearby to realize the partial processing, analysis, and prediction of the data, reduce the number of data transmissions, and shorten the bandwidth and response time.

Cloud computing and edge computing are key technologies to realize the Internet of Things. As a computing model, cloud computing accesses computing resources, network resources, and storage resources of the data center through the network and provides scalable distributed computing capability for applications [26]. With the characteristics of large-scale servers, high reliability, strong extensibility, and virtualization, IT cloud computing is used by more and more enterprises and organizations to deploy their applications. But in cloud computing mode, computing tasks are handled by the cloud center. The service provider provides the data to be uploaded to the cloud center, and the client of the terminal sends the request to the cloud center. The cloud center responds to the relevant request and sends the relevant data to the terminal customer. The terminal customer always plays the role of consumer. Edge computing is a new computing mode to perform computing at the edge of the network, which places the data that should be processed in the cloud center near the data source. The comparison between edge computing and cloud computing is shown in Table 1.

Table 1: Comparison of edge computing and cloud computing.

Content	Edge computing	Cloud computing
Target application	Internet of Things or mobile application	General Internet
Service node location	Edge network	Data center
Communication network	WLAN 4 g/5 g	Wan
Number of devices available for service	Billions	Millions
Types of services provided	Local information	n global information

As can be seen from Table 1, compared with cloud computing, edge computing has the following obvious advantages: first, it can improve the security of data center; third, it can enhance the security of data. But edge computing cannot replace cloud computing. It is the extension of cloud computing, providing a better computing platform for the Internet of Things. Edge computing model requires the strong computing ability and mass storage support of cloud computing center. Cloud computing also needs the processing of massive data and private data by edge devices in edge computing to meet the real-time requirement and satisfy the needs of privacy protection. Therefore, the device edge cloud architecture model can provide a better configuration scheme.

3. Data Compression Preprocessing Based on Edge Computing

Aiming at the problem of cloud computing transmission and feedback delay caused by massive IoT data, an effective method is designed to better process a large amount of sensor time series data. Generally, increasing data redundancy can improve the stability of the system. In a sense, low data redundancy and high data reliability are contradictory, which means it is very difficult to find the optimal solution of minimum data redundancy and maximum data reliability. The shorter the processing time is, the better the compression processing is carried out on the premise that original data characteristics of the sensor and the true reflection of the data are not changed.

The method used in this paper needs to set the number k and error threshold e of each group of data packets in advance. When the time sequence data t is uploaded to the edge end, all the first k temperature data are uploaded. When the average value of the time sequence data $T[i+k]$ and its first k time series data is less than the error threshold e, the output will not be carried out, so as to cycle when $T[i+2k-1]$ and $T\ [i+2k-1]$ still meet the above conditions. We take the average value of $T\ [i+2k-1]$ and the first $k-1$ data as the uploaded data and store them in out2.txt, and $I+k$ in out1.txt. If the time series data $T[i+k]$ appears and the average value of the first k time series data in the group is no less than the error threshold e, then $T[i+k]$ is directly uploaded and stored in out2.txt, and $I+k$ is stored in out1.txt to reduce the amount of data transmission and subsequent data processing. Among them, $T[i]$ is the ith time series data collected, and out1.txt and out2.txt are edge storage files. The implementation of sensing data compression algorithm is shown in Algorithm 1.

4. Anomaly Detection Based on Isolated Forest Algorithm

Isolation forest algorithm is an unsupervised anomaly detection method based on random binary tree and suitable for continuous data [28]. In isolated forests, anomalies are defined as "outliers that are easily isolated," that is, points with sparse distribution and far away from high-density population. In the feature space, the sparsely distributed region indicates that the probability of events occurring in the region is very low, so it is judged that the data distributed in the sparse area is abnormal. It is suitable for anomaly detection of time series data.

The forest isolation algorithm is described in detail:

(i) Define 1 so that t is a binary tree and N is the node of T. if N is a leaf node, it is called an external node; if N is a node with two children, it is called an internal node.

Definition 2 in an iTree; the data of the edge from the root node to the outer node is called the path length, which is denoted as $H\ (s)$.

The construction process of a single iTree is as follows: select a point randomly from the data set $S = \{S1, S2, S3, \cdot\cdot, Sn\}$ to generate the cut point P randomly. The cutting point P is generated between the maximum value and the minimum value of the specified dimension in the current node data, and then each data is divided. The selection of the cutting point generates a hyperplane, which places the points smaller than P in the left branch of the current node and points greater than or equal to P in the right branch of the current node. The left and right branches are constructed recursively until only one data set or tree on the leaf node has grown to the set height. Traverse each iTree to find the final path length of S. Since the cutting process is completely random, we need to use the method of ensemble to make the result converge; that is, repeatedly start cutting from the beginning, and then calculate the average value of each segmentation result, namely, $H\ (s)$. The schematic diagram of data traversal iTree is shown in Figure 1.

```
Input: data.txt sensor data T number of packets processed in a single group K, error threshold E.
Output: out1.txt, out2.txt.
(1)  for i = 1 to N
(2)    read the data from "test.txt", and write them to "data.txt"
(3)  if e of the "test.txt"
(4)    break
(5)  end if
(6)  for i = 1 to N
(7)    read the data from "data.txt" to T[i + 1]
(8)  aver = sum(T)/i + 1;
(9)  end
(10) if (aver < 0)
(11)   for i = 1 to k
(12)     aver < aver + T[i]
(13)     aver < aver/k
(14)   end
(15) else
(16)   for i = 2 to n
(17)     temp < aver
(18)       for j = 0 to k − 1
(19)         if i + j ≥= n
(20)           temp < −1
(21)           aver < aver + T[i + j]
(22)         end if
(23)       end
(24)     end if
(25)   end
(26) end if
(27) aver < aver/k
(28) if |aver-temp| >= e
(29)   put i + j −1 to "out1.txt"
(30)   put T[i + j −1] to "out2.txt"
(31) end if
(32) return "out1.txt","out2.txt"
```

Algorithm 1: Sensor data compression algorithm.

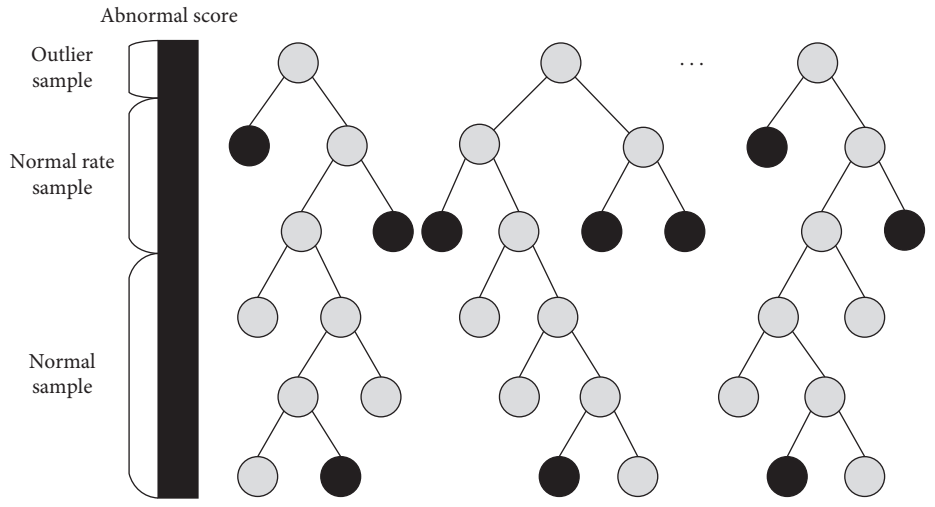


Figure 1: Orest anomaly detection process.

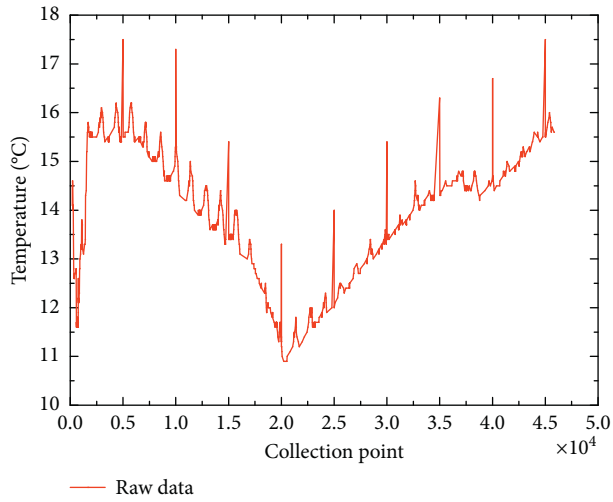


Figure 2: Scatter plot of time series variation trend of temperature data set.

$$H(k) = \ln(k) + \xi,$$

$$c(A) = \begin{cases} 2H(A-1) - \dfrac{2(A-1)}{n}, & A > 2, \\ 1, & A = 2, \\ 0, & A < 2, \end{cases} \tag{1}$$

$$E(h(S)) \longrightarrow 0, s \longrightarrow 1; E(h(S)) \longrightarrow A-1, s \longrightarrow 0; E(h(S)) \longrightarrow c(A), s \longrightarrow 0.5.$$

$H(s)$ is the node depth of S in iTree. $E[.]$ *is* the average of t iTrees. $c(A)$ is the average length of a point bisection search tree. $H(k) = \ln(k) + \xi$, ξ is Euler's constant. The closer $S(S)$ is to 1, the more likely it is to be abnormal data; and the closer it is to 0, the more likely it is to be a normal point. When the $S(S)$ of most data is 0.5, there is no abnormal value in the data.

Isolated forest algorithm is different from clustering, box graph, and other algorithms; it does not need to calculate the distance, density, and other indicators; it can greatly improve the calculation speed and reduce the system overhead. In the process of training, each iTree is randomly selected and generated independently. It accelerates the operation of the deployment of large-scale distributed systems. Based on the ensemble method, the more iTrees, the more stable the algorithm.

5. Experimental Simulation

The temperature data used in this paper is collected from the environmental data set uploaded from the experimental cloud platform of the Internet of Things. The time is intercepted from 8 : 00 on May 1, 2019, to 7 : 15, May 17, 2019. The data upload interval is 30 s, with a total of 45989 temperature sensing data, and the data accuracy is 0.1°C. Figure 2 shows a scatter diagram of time series variation trend of temperature data set, including 10 times of anomalies caused by gradual change or sudden change.

Hardware environment: all experiments are carried out with Windows 7 operating system, CPU is Intel Core i5 4200u, the graphics card is AMD Radeon HD 8670 m, memory is 4 GB, and python platform is used for simulation.

The isolated forest algorithm is used to detect the original temperature data set and four groups of compressed data sets to evaluate the performance of outlier detection. The parameters are as follows: the number of iTree $t = 100$; the number of test samples $a = 256$; the path length $H(s) = 15$. As shown in Figure 3, the test results of iForest algorithm in the original data set show that there are 10 abnormal data detected, all of which are detected without misjudgment. Figures 4–7, respectively, show the anomaly detection results of four groups of data based on iForest algorithm. In the first group, 10 abnormal data were detected, but one normal data was misjudged as abnormal data, and one abnormal data was not detected; nine abnormal data were detected in the second group without misjudgment, and one abnormal data was not detected; the third group detected 10 abnormal data, but there were 2 misjudgments, and 2 abnormal data were not detected; 10 abnormal data were detected in the fourth group, without misjudgment.

In order to verify the comparison and analysis of anomaly detection accuracy of the three algorithms, and to

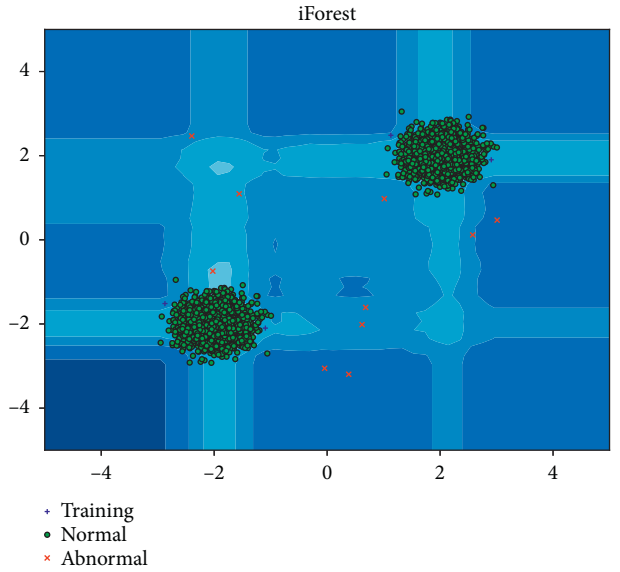


Figure 3: Anomaly detection results of iForest algorithm in the original data set.

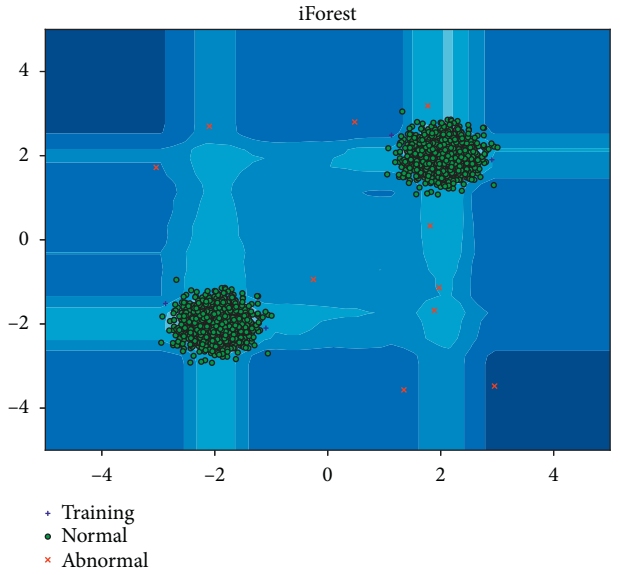


Figure 4: iForest algorithm anomaly detection results of the first group of compressed data.

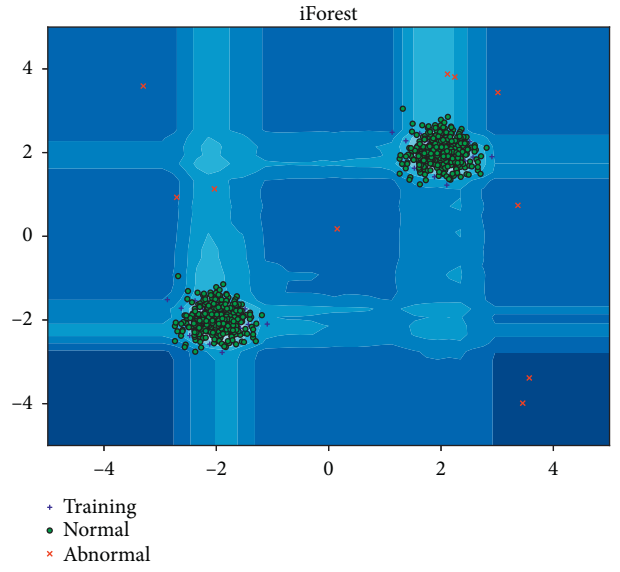


Figure 5: Anomaly detection results of iForest algorithm for the second group of compressed data.

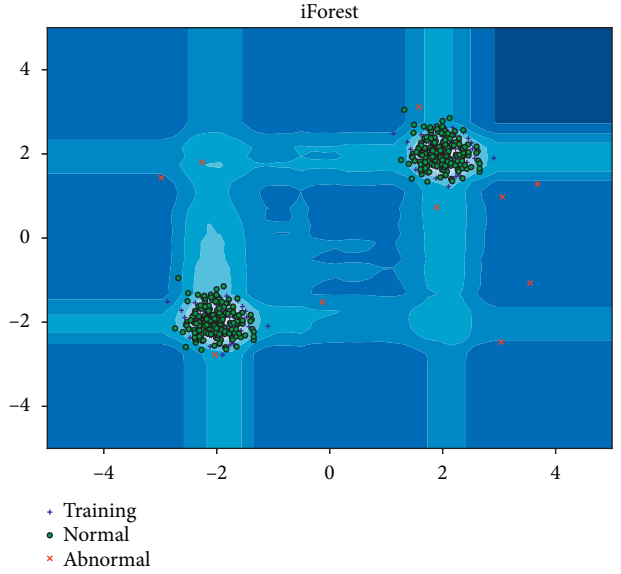


Figure 6: Anomaly detection results of iForest algorithm for the third group of compressed data.

assure the reliability and justness of results, the anomaly detection results of different algorithms for the original data and compressed data are listed in Table 2 (note: the original data is before the processing, and the data is after the processing).

In, Table 2 the accuracy is calculated by the ratio of the number of correctly classified samples to the total number of samples, the accuracy is the ratio of the correct prediction to the positive proportion of all the predicted samples, and the recall rate is the ratio of the correct prediction to the positive proportion of all the positive samples, which can be understood as the ratio between the number found and the total number to be found.

It is not difficult to find from Table 2 that the accuracy and recall rate of iForest algorithm are generally higher than those of the other two algorithms in the anomaly detection of the original data set and the compressed data set. In the comparison of different data sets of the same algorithm, due to the large amount of original data, the abnormal detection accuracy, accuracy, and recall rate of the original data are obviously higher than those of the compressed data, while the other compressed data does not deviate from changing the tracking of the original data. When the data is flat, the compressed data can replace the original data with fewer values; when the data becomes different, the original data can be replaced by the compressed data in normal time, and

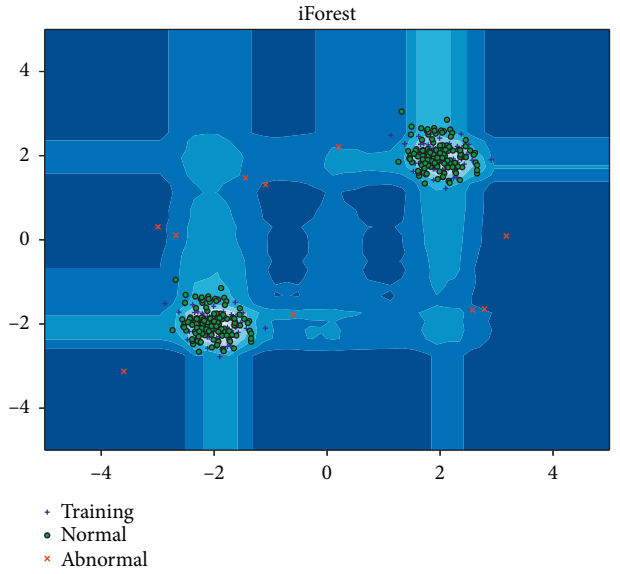


Figure 7: iForest algorithm anomaly detection results of the fourth group of compressed data.

Table 2: Comparison of detection results of three algorithms.

Algorithm category	Box diagram	K-means	iForest
Accuracy	99.98%, 96.95%	99.95%, 96.08%	100%, 99.52%
Precision	99.98%, 96.92%	99.99%, 99.3%	100%, 99.78%
Recall	100%, 99.25%	99.95%, 96.52%	100%, 99.72%
Execution time	15.33 s, 3.98 s	10.87 s, 2.85 s	14.21 s, 3.04 s

compressed data can keep the abnormal data for outlier detection, which can effectively prevent the abnormal data from being missed. In terms of algorithm execution time, the execution time of K-means clustering algorithm is always the shortest, but it is only 0.19 s shorter than iForest algorithm, which has no impact in practical application. Therefore, iForest algorithm based on partition outperforms box graph and K-means clustering algorithm based on distance in anomaly detection performance. From the aspect of the execution time of anomaly detection before and after compression processing, the box graph algorithm is shortened by 11.35 s, K-means clustering algorithm by 8.02 s, and iForest algorithm by 11.07 s. Data compression can significantly shorten the time of anomaly detection. Based on the time required for data compression, the time consumed in the whole data processing is still reduced to a certain extent. Therefore, the superiority of edge computing is finally verified.

6. Conclusions

In this paper, in order to solve the problem of cloud computing, transmission and feedback delay caused by the current massive IoT data, this paper proposes a cold chain monitoring management method based on edge computing through the research and analysis of the cold chain IoT monitoring system. The real-time sensing data is compressed to ensure that the original characteristics and true reflection of the sensing data remain unchanged, and the amount of data calculated in the cloud center can be reduced, as well as the transmission delay and response delay. Based on the data compression processing, the abnormal detection of the filtered data is carried out with high detection precision, which can timely detect anomalies and remind users of them.

In future work, we will take the lead in adjusting the compression conditions in data compression to achieve selective data compression. Secondly, in order to minimize the loss in the process of anomaly repair in the future, we try to add one or more prediction mechanisms in the follow-up work to reasonably optimize the anomaly detection method. Based on the consideration of the correlation between data files and the time interval of similar files being accessed, the cache replacement strategy will be improved.

Data Availability

According to the funding policy of this work, data cannot be shared or made publicly available during the funding contract.

Conflicts of Interest

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

Acknowledgments

This research was funded by Basic Research Project of Heilongjiang Province Department of Education (grant number: 2018-KYYWF-0942) and Heilongjiang Provincial Department of Education Science and Technology Innovation team construction project (2019-kyywf-1335)

References

[1] W. Shi, H. Sun, J. Cao, Q. Zhang, and W. Liu, "Edge computing: a new computing model in the Internet era," *Computer Research and Development*, vol. 54, no. 5, pp. 907–924, 2017.

[2] M. Zwolenski and L. Weatherill, "The digital universe rich data and the increasing value of the internet of things," *Australian Journal of Telecommunications and the Digital Economy*, vol. 2, no. 3, pp. 1–9, 2014.

[3] G. Pandian, M. Pecht, E. Zio, and M. Hodkiewicz, "Data-driven reliability analysis of Boeing 787 dreamliner," *Chinese Journal of Aeronautics*, vol. 33, no. 7, pp. 1969–1979, 2020.

[4] R. K. Runting, S. Phinn, Z. Xie, O. Venter, and J. E M. Watson, "Opportunities for big data in conservation and sustainability," *Nature Communications*, vol. 11, no. 1, 2020.

[5] Z. Li and Y. Zhang, "A new hyperspectral anomaly detection method based on higher order statistics and adaptive cosine estimator," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 4, pp. 661–665, 2020.

[6] H. Liu, Y. Wang, and W. Chen, "Anomaly detection for condition monitoring data using auxiliary feature vector and density-based clustering," *IET Generation, Transmission & Distribution*, vol. 14, no. 1, pp. 108–118, 2020.

[7] 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.

[8] M. Gowri and B. Paramasivan, "Rule-based anomaly detection technique using roaming honeypots for wireless sensor networks," *ETRI Journal*, vol. 38, no. 6, pp. 1145–1152, 2016.

[9] T. Nakazawa and D. V. Kulkarni, "Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 2, pp. 250–256, 2019.

[10] S. Li, "The approach of dynamic data fusion based on multi-sensor temperature data," *Chinese Science and Technology*, vol. 31, no. 1, pp. 146–149, 2015.

[11] T. Yu, Y. Zhu, and X. Wang, "Anomaly detection using self coding neural network in Internet of things monitoring system based on edge computing," *Journal of Internet of Things*, vol. 2, no. 4, pp. 14–21, 2018.

[12] 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.

[13] 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.

[14] Z. Qi, H. Y. Peng, J. Cun et al., "Application of edge computing: real time detection algorithm of sensor data anomaly," *Computer Research and Development*, vol. 55, no. 3, pp. 524–536, 2018.

[15] Y. Chen, "Application of edge computing in smart home," in *Proceedings of the 2019 National Symposium on Edge Computing*, pp. 73–79, China Communications Society, Beijing, China, 2019.

[16] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[17] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): an analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.

[18] Z. Huang, J. Chen, Y. Lin, P. You, and Y. Peng, "Minimizing data redundancy for high reliable cloud storage systems," *Computer Networks*, vol. 81, pp. 164–177, 2015.

[19] J. Zhang, X. Wu, Z. Yang et al., "Research and application of industrial data acquisition technology based on industrial Internet of things," *Telecommunication Science*, vol. 34, no. 10, pp. 130–135, 2018.

[20] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[21] A. W. Colomboaw, S. Karnouskoss, and T. Bangemannt, "Towards the next generation of industrial cyber-physical systems," *Industrial Cloud-Based Cyber-Physical Systems*, pp. 1–22, Springer-Verlag, Beilin, Germeny, 2014.

[22] Y. Xu, H. Liu, and Q. A. Zeng, "Resource management and QoS control in multiple traffic wireless and mobile Internet systems," *Wireless Communications & Mobile Computing*, vol. 5, no. 8, pp. 971–982, 2010.

[23] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y.-C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing," *IEEE Access*, vol. 6, pp. 55923–55936, 2018.

[24] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," *Wireless Networks*, vol. 26, no. 3, pp. 1905–1919, 2020.

[25] H. Wen and P.-H. Ho, "Physical layer technique to assist authentication based on PKI for vehicular communication networks," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 2, pp. 440–456, 2011.

[26] F. Huang, G. Zhou, H. Ding et al., "Electrical energy anomaly data detection based on isolated forest algorithm," *Journal of East China Normal University (Natural Science Edition)*, vol. 5, pp. 123–132, 2019.

Hindawi
Scientific Programming
Volume 2021, Article ID 6613452, 12 pages
https://doi.org/10.1155/2021/6613452

Research Article

Semisupervised Deep Embedded Clustering with Adaptive Labels

Zhikui Chen, Chaojie Li, Jing Gao, Jianing Zhang, and Peng Li

School of Software Technology, Dalian University of Technology, Dalian 116620, China

Correspondence should be addressed to Jing Gao; gaojinghit@gmail.com

Received 29 October 2020; Revised 14 December 2020; Accepted 8 January 2021; Published 16 January 2021

Academic Editor: Boxiang Dong

Deep embedding clustering (DEC) attracts much attention due to its outperforming performance attributed to the end-to-end clustering. However, DEC cannot make use of small amount of a priori knowledge contained in data of increasing volume. To tackle this challenge, a semisupervised deep embedded clustering algorithm with adaptive labels is proposed to cluster those data in a semisupervised end-to-end manner on the basis of a little priori knowledge. Specifically, a deep semisupervised clustering network is designed based on the autoencoder paradigm and deep clustering, which well mine the clustering representation and clustering assignment by preventing the shift of labels in DEC. Then, to train parameters of the deep semisupervised clustering network, a back-propagation-based algorithm with adaptive labels is introduced based on the pretrain and fine-tune strategies. Finally, extensive experiments on representative datasets are conducted to evaluate the performance of the proposed method in terms of clustering accuracy and normalized mutual information. Results show the proposed method outperforms the state-of-the-art methods of DEC.

1. Introduction

Clustering, as one of the most important basic research methods in data mining and machine learning, plays an important role in pattern recognition, image retrieval, computer vision, social network analysis, natural language processing, and knowledge discovery [1]. It divides data samples into different categories in the pattern space by exploring potential distribution structures of data. In the past decades, many classical clustering algorithms have been proposed, such as K-means, DBSCAN, Gaussian mixture model, spectral clustering, nonnegative matrix factorization-based clustering, and graph-based clustering [2–5]. Recently, deep clustering has attracted much attention with the increasing collection of high-dimensional data. It can well alleviate the degradation of traditional clustering in the face of high-dimensional input data by learning low-dimensional representations of data. For example, Lv et al. [6] proposed a deep feature-based clustering by using a stacked autoencoder to extract deep text features. To further improve clustering performance on high-dimensional data, some deep end-to-end clustering methods were proposed, which merged deep neural networks into clustering. For instance, Xie et al. [7] proposed deep embedded clustering (DEC), which learns clustering features of data and divides data in a self-learning manner. Hong et al. [8] proposed mini-GCN, which can combine CNN and GCN to extract more distinctive features and overcome the high computational cost of GCN. Zhao et al. [9] separated view-specific irrelevant information from common features, eliminating the influence of useless information in the view.

Those above methods can well mine data patterns in an unsupervised manner, neglecting some prior knowledge in real data, which is represented by a small number of labelled data or pairwise constraints given by experts. Lately, a number of semisupervised clustering methods were proposed [10–13], utilizing both enough unlabelled data and some prior knowledge to improve clustering performance. For example, Hong et al. [14] proposed a semisupervised deep learning framework that can learn more discriminative information from a small-scale hyperspectral image and transfer it to the classification task of large-scale data. However, most of the current semisupervised clustering cannot use a priori knowledge in a strong-supervision manner because they do not use label information to directly guide the learning of cluster centres. Also, they cannot

cluster samples in a data-driven way of learning clustering centroids and clustering-specific representations.

To address those challenges, a new semisupervised joint learning framework is proposed, which jointly learns the feature embedding space and cluster assignment by integrating a small amount of label information in a joint optimization function.

In addition, the previous semisupervised clustering strategies cannot directly use the strong-supervised knowledge of data labels in the deep embedded clustering due to the label shift problem that clustering results are inconsistent with the actual labels of samples. In other words, those labelled samples of the same class are often scattered to the incorrect classes, and this incorrect supervised information destroys pattern structures of data, causing the degradation of deep embedded clustering.

To solve this challenge, a label adaptive strategy is introduced in this paper based on a voting mechanism. Through the label adaptive strategy, the shifted labels generated in the clustering process are projected as the winner label, ensuring that the labelled samples of the same cluster are always in one cluster in the clustering process. So, the proposed strategy can directly use the label loss to guide the clustering process via adjusting the cluster centres and learning clustering-specific representations. The method in this paper is improved on the basis of DEC and expanded to a semisupervised deep clustering method. The contributions of this paper are summarized as follows:

(i) A new semisupervised joint learning framework is proposed, which integrates a small amount label information to jointly learn the feature embedding space and the cluster assignment with the help of a joint optimization function.

(ii) A label adaptive strategy is introduced to correct the label shift of the clustering process. It can not only improve the utilization of label information, but also effectively avoid the potential degradation that the centroid of traditional deep clustering algorithm is dominated by the code network.

(iii) Extensive experiments on two image datasets and one text dataset are conducted, where the results prove that the proposed method greatly outperforms the state-of-the-art clustering methods.

The rest of this paper is organized as follows: we briefly review the related work in Section 2. Section 3 introduces the details of the proposed method. Section 4 introduces the back-propagation-based algorithm with adaptive labels based on the pretraining and fine-tuning strategies. Section 5 introduces the experimental details of this paper. Finally, the conclusions are presented.

2. Related Work

2.1. Unsupervised Clustering. Clustering has attracted a lot of attention and has been greatly developed for a long time. Many excellent clustering algorithms were proposed [15, 16]. For example, K-means is a classical unsupervised clustering algorithm aiming to minimize the sum of the distance between data points and centroids [2]. Fuzzy expectation maximization combines clustering, cluster number detection, and feature selection into an estimation problem to perform the clustering process [17]. Feature clustering hashing (FCH) is a hashing method based on feature clustering, which can generate lower dimensional data with balanced variance on the premise of maintaining similarity in the Euclidean space [18]. The above methods can be regarded as the clustering algorithm based on features. Distance metric learning with side information learns a distance measure that incorporates the given similarity pairs. Learning a Mahalanobis distance metric designs a new distance measurement function that can learn the Mahalanobis distance metric by forcibly adjusting the distance of a given instance and applying it to new data [19]. Bayesian discriminative fuzzy clustering (BDFC) designs a probabilistic method for unsupervised distance metric learning which can maximize the separability between different clusters in the projection space [20]. The above methods can be regarded as the clustering algorithm based on the distance metric learning. Constrained Laplacian rank (CLR) learns graph with k connected components (where k is the number of clusters) and adjusts the data graph as part of the clustering process [21]. Structure doubly stochastic (SDS) learns structured double random matrices by applying low-rank constraints on Laplace matrices of graphs [22]. Multiview spectral clustering is a novel multiview Markov chain clustering method which can utilize complementary information embedded in different views [23]. The above methods can be regarded as the clustering algorithm based on a graph. With the rise of deep learning, the introduction of deep neural network in clustering has received much attention. Deep clustering network (DCN) finds K-means-friendly clustering space through synchronous deep learning and clustering process [24]. Deep embedded clustering (DEC) uses an automatic encoder to complete the transformation of feature space [7]. Ingeniously, it can perform feature extraction and cluster assignment tasks simultaneously. This algorithm achieves good results and becomes a reference for the performance of new deep clustering algorithm. Improved deep embedded clustering (IDEC) improves clustering performance by preserving the local structure of data [25]. Colearning nonnegative correlated and uncorrelated features (CoUFC) [26] recognizes view-specific features and eliminates the influence of irrelevant information to obtain useful interview feature correlation.

There exists some prior information in many actual data, but the above unsupervised methods do not consider the information. In order to make full use of the label information, this paper proposes a new semisupervised joint learning framework, which integrates label information into deep clustering to jointly learn the data representations and the clustering assignment.

2.2. Semisupervised Clustering. Semisupervised clustering is one of the important research directions in the field of data mining. It can guide the clustering process and improve the quality of clustering by using prior knowledge such as paired constraints or a small amount of labelled data. Recently, the semisupervised clustering method has achieved fruitful

results. For instance, semisupervised kernel mean shift clustering (SKMS) maps data points to a high-dimensional kernel space in which constraints are imposed by linear transformation of the mapped points [27]. Semisupervised linear discriminant clustering (SLDC) combines k-means and linear discriminant analysis (LDA) to consider both the clustering and dimensionality reduction and finds the appropriate feature space by using soft LDA with unlabelled examples [28]. Semisupervised nonnegative matrix factorization (CPSNMF) propagates limited constraint information to the entire data set to obtain more supervisory information and utilizes this supervisory information to maintain the geometry of the data space [29]. Semisupervised graph-based clustering (SSGC) uses a graph of k-nearest neighbours and the local density measure of the similarity between vertexes to integrate the seed into the process of building the cluster, improving the quality of the cluster [30]. The above methods can be regarded as an extension of the traditional clustering algorithms by using label information or pairwise constraints. Relevant component analysis (RCA) is an efficient algorithm for learning Mahalanobis metrics by using a version of the constrained Fisher's linear discriminant [31]. Discriminative component analysis (DCA) learns the linear data transformation of the best Mahalanobis distance measurement with context information [32]. Information theoretic metric learning (ITML) uses a relationship between multivariate Gaussian distribution and Mahalanobis distance set to learn a new Mahalanobis distance function [33]. Bregman distance function learning (BKM) presents a new method for learning nonlinear distance functions with edge information, which is to use a nonparametric method similar to support vector machines to learn Bregman distance functions [34]. The above methods can be considered as exploring a new distance metric function by using constraint information. Still some research work is used to explore an integrated framework for semisupervised clustering. For example, the double affinity propagation-based cluster ensemble (AP^2C) integrates affinity propagation (AP) algorithm and normalized cut (Ncut) algorithm into cluster integration framework [35]. It can capture the relationship between attributes, find a group of representative attributes, and eliminate noise attributes. Semisupervised clustering with sequential constraints (SCSC) proposes an efficient dynamic semisupervised clustering framework [36]. It transforms the dynamic clustering process into a search problem on a feasible clustering space, which is defined as a convex shell generated by partitioning multiple sets. Hybrid semisupervised clustering ensemble (HSCE) proposes a semisupervised clustering ensemble framework that uses pairwise constraints or labelled data to generate different basic partitions by using constraint-based semisupervised clustering algorithm and metric-based semisupervised clustering algorithm, respectively, and then integrates these basic partitions into integration functions to obtain target clustering [37].

Traditional semisupervised clustering algorithms are mostly executed in the original space and have poor performance in the face of high-dimensional data. Therefore, it is necessary to enhance its expressiveness by using deep neural network. MDL-RS designs a general multimodal deep learning framework, which can well embed multiple fusion modules and break the performance bottleneck under single modality [38]. Deep transductive semisupervised maximum margin clustering uses labelled and unlabelled data under a given pair of constraints to learn the nonlinear mapping under the maximum margin framework for clustering analysis [39]. This work proves that the deep representation of the original does contribute to the improvement of clustering results. Semisupervised deep embedded clustering (SDEC) incorporates pairwise constraints in the process of feature learning, forcing data samples in the same cluster to be close to each other, and data samples of different clusters are far apart from each other [40].

However, due to the label shift problem, these semisupervised methods cannot directly use label information to guide the learning of cluster centres. Therefore, this paper designs a label adaptive strategy based on the voting mechanism to correct the transfer of labels in the clustering process, directly using the label loss to guide the clustering process and improve the clustering performance.

3. Semisupervised Deep Clustering with Adaptive Labels

In this section, a semisupervised deep embedded clustering algorithm with adaptive labels (Semi-DEC) is introduced to make full use of prior knowledge of a small number of labels. Semi-DEC is composed of a deep code network and a semisupervised embedding network, as shown in Figure 1. The former uses the encoder-decoder paradigm, transferring high-dimensional data into low-dimensional features. It can well address the curse of dimensionality in data. The latter mines knowledge patterns by dividing data into several groups. It can better consider prior knowledge by solving the shift of labels in clustering. The details of those two networks are introduced as follows.

3.1. The Deep Code Network. The deep code network aims to learn latent features of data in a low-dimensional space on the basis of the encoder-decoder network [41]. That is, it computes the hidden representations of data samples, reconstructs data samples from those hidden representations, and minimizes the loss between raw data and reconstructed data. Specifically, given a dataset of n points $X = \{x_i \in R^{d_1}\}_{i=1}^{n}$, where d_1 is the dimension of data, the deep code network learns hidden representations of data in the following form:

$$\overline{x} = \text{Dropout}(x), \tag{1}$$

$$h = g_e(W_e\overline{x} + b_e), \tag{2}$$

where Dropout is the random mapping function that sets some elements of each input to be 0 based on a given probability. $\overline{x}$ is the result of the random mapping of the input x. W_e and b_e are the weight and bias vectors, respectively, which represent the parameters of the encoder

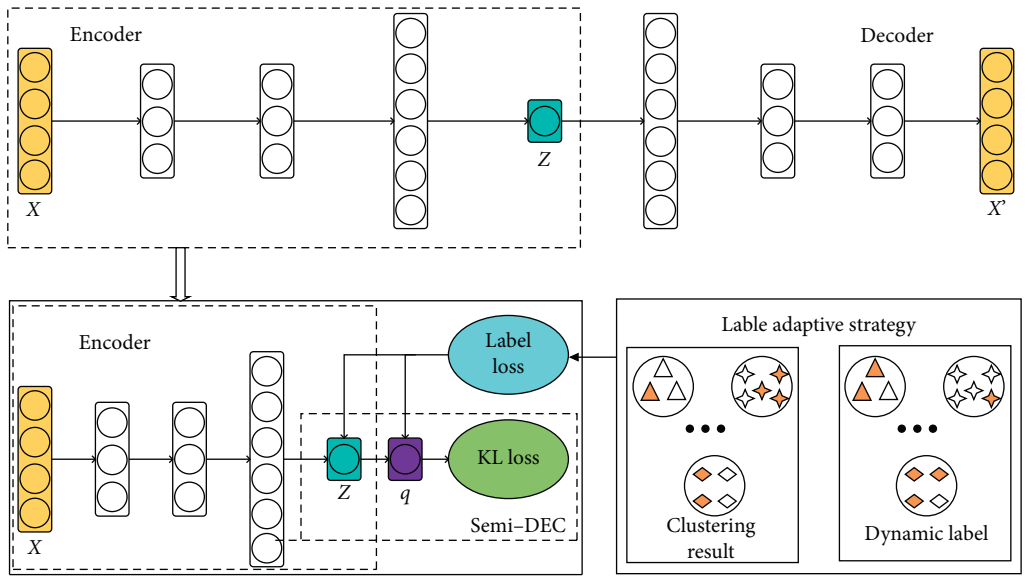


Figure 1: The architecture of the semisupervised deep embedding clustering algorithm with adaptive labels.

network. h is the hidden representation with g_e representing the encoder function.

After obtaining hidden representations of data samples, the deep code network decodes hidden representations by the reconstructing function as follows:

$$\overline{h} = \text{Dropout}(h), \tag{3}$$

$$t = g_d\left(W_d\overline{h} + b_d\right), \tag{4}$$

where $\overline{h}$ is the result of the random mapping of the hidden representation h. W_d and b_d are the weight and bias vectors of the decoder function. t is the reconstructed data, and g_d represents the decoder function.

Finally, the deep code network uses the mean squared error function to measure the loss between raw data and reconstructed data as follows:

$$\text{loss} = \|x - t\|_2^2, \tag{5}$$

where $\frac{2}{2}$ represents the mean squared error function. In Semi-DEC, the loss of the deep code network is used to pretrain parameters.

3.2. The Semisupervised Embedding Network. The semisupervised embedding network aims to divide data into several groups where the distances between samples of the same group are closer than those of different groups. The semisupervised embedding network consists of the unsupervised part that mines intrinsic patterns and the supervised part that uses the small amount of prior knowledge.

3.2.1. The Unsupervised Part. The unsupervised part of the semisupervised embedding network is measured by the KL divergence as follows:

$$L_1 = \text{KL}(P\|Q) = \sum_{i=1}^{n}\sum_{j=1}^{k} p_{ij}\log\frac{p_{ij}}{q_{ij}}, \tag{6}$$

where Q is the cluster assignment of the semisupervised embedding network and P is the target distribution. Given the hidden representations of n data samples $Z = \{z_i \in R^{d_2}\}_{i=1}^{n}$ (d_2 is the dimension of data in the embedding space) and k cluster centroids $\mu_j | j = 1, \ldots, k$, the cluster assignment of the semisupervised embedding network is expressed as

$$q_{ij} = \frac{\left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1}}{\sum_{j'=1}^{k}\left(1 + \left\|z_i - \mu_{j_\prime}\right\|^2\right)^{-1}}. \tag{7}$$

The target distribution is defined as follows:

$$p_{ij} = \frac{\left(q_{ij}^2 / \sum_{i=1}^{n} q_{ij}\right)}{\left(\sum_{j'=1}^{k} q_{ij}^2 / \sum_{i=1}^{n} q_{ij_\prime}\right)}, \tag{8}$$

where Q is measured by the student distribution and P is the square of Q, which strengthens the membership each sample.

3.2.2. The Supervised Part. The supervised part is introduced to address the shift of labels in the unsupervised part based on the small group of priori knowledge. It is measured by the soft-max loss function as follows:

$$L_2 = -\lambda\sum_{i=1}^{n} a_i y_i' \log q_i = -\lambda\sum_{i=1}^{n}\sum_{j=1}^{k} a_i y_{ij}' \log q_{ij}, \tag{9}$$

where y_i' represents the temporary correction label obtained through the label adaptive strategy, λ is a trade-off parameter to balance the influence of the label loss, q_i represents the label obtained by cluster assignment, and a_i is the sign that indicates whether there is a label of a certain sample and is expressed via

$$a_i = \begin{cases} 1, & y_i \text{ exists}, \\ 0, & \text{else}, \end{cases} \tag{10}$$

where y_i represents the true label of the sample.

Finally, the computation of the semisupervised embedding network is expressed as follows:

$$L = L_1 + L_2 = \sum_{i=1}^{n} \sum_{j=1}^{k} p_{ij} \log \frac{p_{ij}}{q_{ij}} - \lambda \sum_{i=1}^{n} \sum_{j=1}^{k} a_i y_{ij}' \log q_{ij}, \tag{11}$$

which can effectively merge the knowledge of a small number of labels into the unsupervised learning.

3.3. Optimization. We use the stochastic gradient descent (SGD) and back-propagation to optimize the loss function equation (11). It is worth noting that the parameters to be optimized have two parts: feature space embedded of each data point z_i and the cluster centres μ_j. The gradients of L with respect to embedded point z_i can be computed as

$$\frac{\partial L}{\partial z_i} = 2 \sum_{j=1}^{k} \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(p_{ij} - q_{ij}\right)\left(z_i - \mu_j\right) - 2\lambda a_i \sum_{j=1}^{k} y_{ij}' \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(1 - \frac{q_{ij}}{p_{ij}}\right)\left(z_i - \mu_j\right). \tag{12}$$

The gradients of L with respect to the cluster centre μ_j can be computed as

$$\frac{\partial L}{\partial \mu_j} = -2 \sum_{i=1}^{n} \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(p_{ij} - q_{ij}\right)\left(z_i - \mu_j\right) + 2\lambda a_i \sum_{i=1}^{n} y_{ij}' \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(1 - \frac{q_{ij}}{p_{ij}}\right)\left(z_i - \mu_j\right). \tag{13}$$

In the process of back propagation, the parameters $\{W_e, b_e\}$ in the deep code network are updated by passing down the gradient $(\partial L/\partial z_i)$. The cluster centre μ_j is updated by gradient $(\partial L/\partial \mu_j)$. The clustering process will be terminated when the cluster assignment between two consecutive iterations is less than tol % or the maximum number of training times is reached.

4. The Back-Propagation Algorithm of Semi-DEC

In this section, the back-propagation algorithm is introduced to train parameters of Semi-DEC. It is composed of two steps, i.e., the unsupervised pretraining step and the semisupervised fine-tuning step. The details of the back-propagation algorithm of Semi-DEC are introduced as follows.

4.1. The Unsupervised Pretraining Step. The unsupervised pretraining step uses the encoder-decoder paradigm to learn generalized features of data and adopts the K-means clustering to explore the centroids hidden in data.

Specifically, given a dataset of n points X and a deep encoder network of m layers, the unsupervised pretraining step models each layer of the deep encoder network as an autoencoder based on equations (1) to (4) to obtain the pretraining parameters of the deep code network. For example, each raw sample x_i in the dataset is input into the autoencoder of the 1st hidden layer, obtaining the hidden representation h_i which is input into the autoencoder of the 2nd hidden layer. After each hidden layer is initialized in the same way, the whole network is trained again in an end-to-end manner by minimizing the reconstruction loss.

Then, the raw data X are mapped into the latent feature space by the deep code network, getting the hidden representations Z. The K-means clustering is conducted on the hidden representations to get initial centroids.

4.2. The Semisupervised Fine-Tuning Step. After obtaining the pretrained deep code network and the initial centroids, Semi-DEC is trained in the semisupervised manner based on the loss function equation (11) to solve the shift of label in unsupervised learning. Specifically, given the raw data X, Semi-DEC constructs the label sign list of samples as defined in equation (10). Then, suppose the number of labelled samples is v, it gathers statistics of the distribution of data which have labels in each epoch as follows:

$$\begin{cases} R = [q_1, q_2, \ldots, q_v], \\ q_i = \arg\max_j q_{ij}, \quad j = 1, 2, \ldots, k, \end{cases} \tag{14}$$

where $q_1, q_2, \ldots, q_v$ represent the assigned labels of those labelled data and their values range from 1 to k. Finally, the temporal labels $q_1, q_2, \ldots, q_v$ are rectified to the label whose number is maximum.

Figure 2 is an example of the label adaptive strategy. For the subset of labelled data with category o, we assume that after cluster assignment, most of the samples are assigned to

category j and a few samples are assigned to other categories such as s and u. Here, o, j, s, and u, respectively, represent different categories. Through the voting mechanism, we believe that the category j with the largest number of samples is the correct result of this subset in cluster assignment. Then, we can rectify the samples that are clustered incorrectly in this round of calculation, that is, make them move closer to category j. The adaptive label algorithm is introduced as follows:

Step 1: Semi-DEC gathers the label distribution of each cluster $\{c_i|i = 1, \ldots, k\}$ based on the output of the semisupervised embedding network $R = [q_1, q_2, \ldots, q_v]$ in each epoch. At the same time, the label with the maximum number is dynamically treated as the correct label.

Step 2: Semi-DEC rectifies those wrong labels according to the statistics of the label distribution.

Step 3: Semi-DEC computes the loss of those samples that are wrongly labelled according to equation (9) to rectify the parameters of network.

Step 4: Semi-DEC fine-tunes the parameters of the deep code network and the semisupervised clustering network to find the final assignment strategy.

With the help of the proposed label adaptive strategy, the labelled data that were wrongly divided in the clustering process are corrected via the voting mechanism, which can effectively solve the label shift problem in a strong-supervision manner by forcing the data that have the same label to be in the same cluster. In other words, this label adaptive strategy preserves the data structure in clustering assignment and cluster-specific feature learning. The overall steps of the back-propagation algorithm of Semi-DEC are shown in Algorithm 1.

5. Experiments

In this section, extensive experiments are conducted on several representative datasets to evaluate the performance of Semi-DEC. The datasets used in our experiment are first introduced. Then, several state-of-the-art clustering algorithms and evaluation metrics are presented. Finally, the implementation and experimental results are illustrated in detail. The detailed information of the datasets is shown in Table 1.

5.1. Datasets

5.1.1. MNIST. The MNIST dataset is composed of 70000 handwritten digits of $28 * 28$ pixel size. In the experiment, each image is reshaped to a 784-dimensional vector.

5.1.2. USPS. The USPS dataset is composed of 9298 handwritten digits of $16 * 16$ pixel size. The images are divided into 10 categories, with a training set size of 7291 and a test set size of 2007.

5.1.3. REUTERS-10K. In the original Reuters data set, there are around 810000 English news stories labelled with a category. Four root categories are as follows: corporate/industrial, government/social, markets, and economics as labels are used, and all documents with multiple labels are further excluded. We computed TF-IDF features on the 2000 most frequent words to represent all documents. A subset of 10000 samples is randomly sampled, referred as REUTERS-10K.

5.2. Compared Methods. To verify the effectiveness of the proposed method, several state-of-the-art algorithms are used as the compared methods. The following is a summary of these algorithms.

5.2.1. K-Means. K-means is a traditional unsupervised clustering algorithm [2]. It guides the division of data sets into K classes based on the principle of minimizing the sum of the distances from the data points to the centroids.

5.2.2. DEC. The deep embedding clustering (DEC) is a deep unsupervised clustering algorithm [7]. It uses an automatic encoder to transform feature of the original data and then performs the clustering process in the feature space.

5.2.3. DCN. The deep clustering network (DCN) is a deep unsupervised clustering algorithm [24]. It combines autoencoder with the K-means and proposes an algorithm that jointly optimizes reconstruction loss and K-means loss.

5.2.4. IDEC. The improved deep embedding clustering (IDEC) is also a deep unsupervised clustering algorithm [25]. It is an improvement to DEC by adding the local structure preservation.

5.2.5. SMKL. The self-weighted multiple kernel learning (SMKL) is a traditional semisupervised clustering algorithm [13]. It constructs the best kernel and assigns an optimal weight for each kernel automatically.

5.2.6. SDEC. The semisupervised deep embedded clustering (SDEC) is a deep semisupervised clustering algorithm [40]. It incorporates pairwise constraints in the process of the feature learning.

5.3. Evaluation Metric. The clustering accuracy (ACC) and normalized mutual information (NMI) are used to evaluate the performance of the proposed method and other compared algorithms, which are widely used in clustering tasks. The values of both ACC and NMI range from 0 to 1. The larger values of both metrics indicate the better clustering results.

ACC is defined as follows:

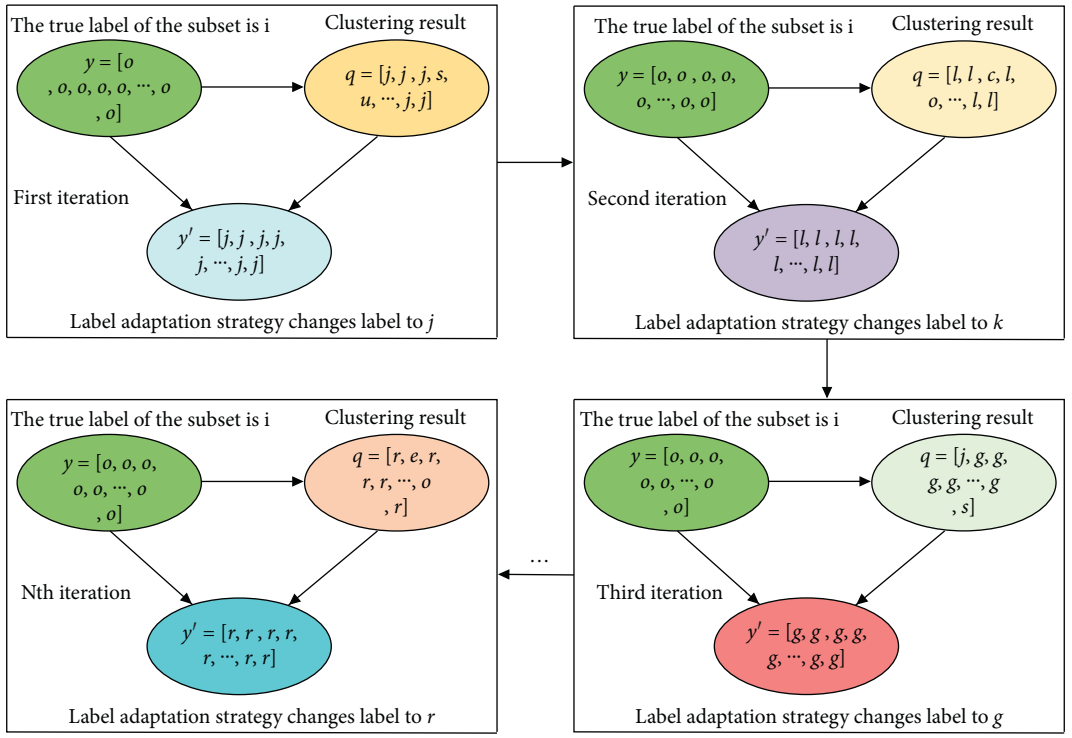


Figure 2: An example of the label adaptive strategy.

Input: the training dataset $\{x_i \in X\}_{i=1}^{n}$, the number of clusters k, the iteration maximum maxiter, and the training threshold.
Output: the cluster assignment Q, the cluster centroids $\{\mu_j\}_{j=1}^{k}$, and the nonlinear mapping f_θ.
Begin
Pretraining computing:
To construct the deep code network.
To initialize network parameters based on the normal distribution.
To train each layer of the deep code network based on the denoising autoencoder strategy.
To connect each pretrained layer and fine-tune network parameters in an end-to-end manner.
To use pretrained deep code network to map raw data into the latent space for obtaining feature z_i.
To use K-means to initialize centroids $\{\mu_j\}_{j=1}^{k}$ based on feature z_i.
Clustering computing with adaptive labels:
To use equations (7) and (8) to compute cluster assignment Q and target assignment P.
To compute $(\sum_{i=1}^{n} q_{\text{old}_i} \neq q_i) < \text{tol}\,\%$.
To use equation (10) for constructing the label list.
To dynamically rectify labels based on the adaptive label algorithm.
To compute the loss based on equation (11).
To update network parameters and centroids.
End

Algorithm 1: Deep semiclustering with adaptive labels.

$$\text{ACC} = \frac{1}{N} \max_{k} \sum_{i=1}^{n} 1\{l_i = k(c_i)\}, \tag{15}$$

where N is the number of samples, l_i is the true label, c_i is the cluster assignment label produced by the algorithm, and k ranges over all possible one-to-one mappings between clusters and labels.

NMI is defined as follows:

$$\text{NMI}(A, B) = \frac{\text{MI}(A, B)}{\sqrt{H(A)H(B)}}, \tag{16}$$

where A is the true cluster set and B is the predicted cluster set. MI (A, B) is the mutual information between A and B. H (A) and H (B) denote the entropies of A and B.

5.4. Parameters Setting. The encoder layer structure of deep code network is set to d-500-500-2000-10 for all data sets, where d is the dimension of the input data. All layers are fully connected, and all internal layers (except the input layer, embedding layer and output layer) are activated by the ReLU nonlinear function. During the pretraining and fine-tuning of the autoencoder network, we use the same parameter settings as in DEC to ensure that the improvement of the experimental results is the contribution of the method proposed in this paper.

For each dataset, the monitor information list A is dynamically generated based on the presence or absence of label information in the dataset. The length of the list is consistent with the size of the data batch taken each time, and its corresponding element value is 1 if the data point has a real label, or 0 if there is no label. The learning rate of SGD is 0.01. The convergence threshold to l % is set to 0.1%. After experimental testing, the trade-off parameter λ of label loss is set to 0.2 (this is determined by a grid search in {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0}). For all algorithms, we set the cluster number k as the number of ground truth categories. We independently run each algorithm 10 times and report average results.

5.5. Experiment Results. This section demonstrates the results of the compared methods on the three representative datasets. In detail, Tables 2 and 3 report the results in terms of ACC and NMI, respectively. The percentage of labelled data is 30%. In the two tables, the best-performance results are highlighted in bold. It can be seen the proposed method is superior to the state-of-the art methods.

Specifically, compared with the traditional K-means and SMKL methods, the proposed method can learn features of more representational capabilities by the deep code network. Also, the K-means is an unsupervised method, which cannot utilize label information in the clustering process, further leading to the degradation of the performance. Although DEC, DCN, and IDEC also take advantage of deep features of data, they ignore the information hidden in the small amount of label data, resulting that those deep methods produced lower performance than the proposed method. SDEC uses pairwise constraints to guide the process of clustering, which belongs to a weak utilization of supervisory information. Through the label adaptive strategy, we can directly use the label loss, which is a strong use of label information. This is also the key to our proposed approach.

To further illustrate the superiority of the proposed method, we also visualize the clustering results in the training process in Figure 3. We randomly select 1000 samples in each dataset and map the latent representations z into the 2D space. From the change trend of the clustering results, it can be seen that the samples in different clusters become easier to distinguish as the number of trainings increases, and the samples in the same cluster also become closer. This indicates that the learned feature space becomes more suitable for clustering tasks, and it is also a proof that the label adaptive strategy can effectively guide the learning of the feature space and cluster assignment.

Table 1: Datasets statistics.

Datasets	Samples	Dimension	Classes
MNIST	70000	784	10
USPS	9298	256	10
REUTERS-10K	10000	2000	4

Table 2: Clustering results measured by ACC.

Methods	MNIST	USPS	REUTERS-10K
K-means	0.5298	0.6567	0.5162
DEC	0.843	0.7408	0.7369
DCN	0.811	0.73	0.7505
IDEC	0.8806	0.7605	0.7564
SMKL	0.783	0.6819	0.7203
SDEC	0.8611	0.7639	0.6937
Semi-DEC	**0.9648**	**0.8609**	**0.9176**

Also, to evaluate the influence of the prior knowledge on the performance of Semi-DEC, the radio of labelled training samples is increased from 1% to 50%. Each experiment is carried out 10 times, and the average results are shown in Table 4. And Table 5 shows the classification accuracy results produced by the same network architecture with Semi-DEC.

As shown in Tables 4 and 5 and Figure 4, there are two observations. First, the ACC and NMI results become larger in all three datasets as the number of labelled samples increases. Especially, the ACC and NMI can reach 97.5% and 95.2%, respectively, on the MNIST dataset with 50% labelled training images. Second, the clustering ACC of Semi-DEC on datasets with 50% labelled data is approximately equal to the classification ACC on the three datasets. Those observations indicate the outperformance of Semi-DEC.

In order to further test the method in this paper, we conducted experiments in many aspects, including the impact of different proportions of labelled data on performance, the change process of loss function and accuracy, and the effect of trade-off parameter λ on clustering performance and running time analysis.

Specifically, about the impact of different proportions of labelled data on performance, Figure 4 shows the trend of the accuracy of the clustering results on the MNIST, USPS, and REUSTER-10K datasets. The dotted line represents the classification accuracy results obtained through multiple experiments under the same network architecture with Semi-DEC. It can be more intuitively shown that with the gradual increase of the proportion of labelled data, the effect of Semi-DEC can be close to the classification effect in the MNIST and REUSTER-10K datasets. Although the clustering effect on the USPS dataset still has a certain gap with the classification effect, it is not far away.

The change process of the loss function and accuracy with the increase of training times is recorded in Figure 5. It can be seen that after reaching a certain number of iterations,

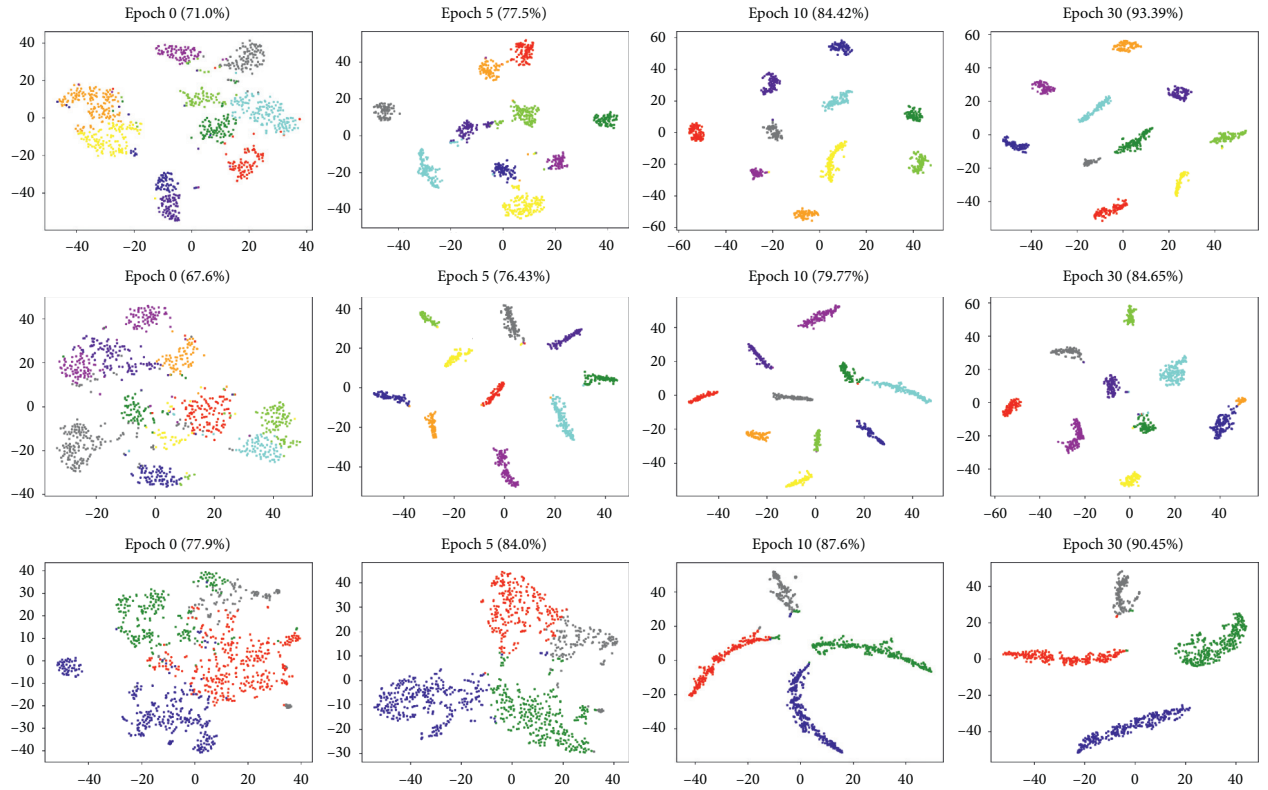


Figure 3: The visualization of clustering results during training on subset of MNIST, USPS, and REUTERS-10K from top to bottom. Different colours mark different clusters. The clustering accuracy of the corresponding epoch is given in parentheses. It can be seen that the data of the same class become more compact while the data of different classes are further away from each other as the number of epochs increases. This also shows that the learned feature embedding space is more and more suitable for clustering tasks.

Table 3: Clustering results measured by NMI.

Methods	MNIST	USPS	REUTERS-10K
K-means	0.4974	0.62	0.4932
DEC	0.8372	0.7529	0.4976
DCN	0.757	0.719	0.4106
IDEC	0.8672	0.7846	0.4981
SMKL	0.6842	0.7105	0.4076
SDEC	0.8289	0.7768	0.4762
Semi-DEC	**0.9457**	**0.8654**	**0.7642**

Table 4: Clustering results on datasets of various ratios of labelled data.

Datasets	1%		2%		5%		10%		20%		30%		40%		50%	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
MNIST	0.809	0.774	0.815	0.783	0.843	0.828	0.886	0.881	0.920	0.916	0.965	0.946	0.965	0.949	0.975	0.952
USPS	0.748	0.755	0.758	0.776	0.776	0.784	0.787	0.807	0.805	0.847	0.861	0.884	0.884	0.881	0.885	0.878
REUTERS-10K	0.751	0.506	0.758	0.519	0.769	0.554	0.795	0.586	0.863	0.68	0.918	0.764	0.954	0.829	0.956	0.831

the loss value and accuracy will tend to be stable, which is also a proof of the robustness of the method in this paper.

To see how the trade-off parameter λ of label loss affects the performance of the method in this paper, we conduct experiment on three datasets by sampling in range [0.01, 5.0]. Figure 6 gives the results. As shown in this figure, our method performs stably in a wide range of λ. The main reason is that the semisupervised loss dominates in this case. When λ is 0.2, the performance is asymptotically optimal.

About the running time, Figure 7 records the running time comparison between our method and DEC. Since the method in this paper is a further study on the basis of DEC, it only compares the running time with DEC. It can be seen

Table 5: Classification accuracy on the three datasets.

Datasets	MNIST	USPS	REUSTERS-10K
Average ACC	0.972	0.931	0.949

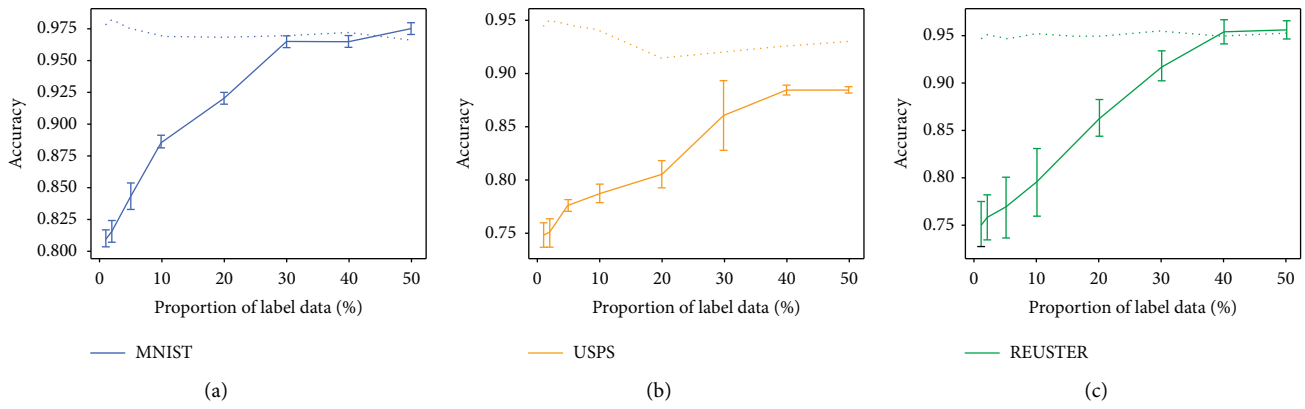


Figure 4: Accuracy of labelled data at different proportions on (a) MNIST, (b) USPS, and (c) REUTERS-10K.

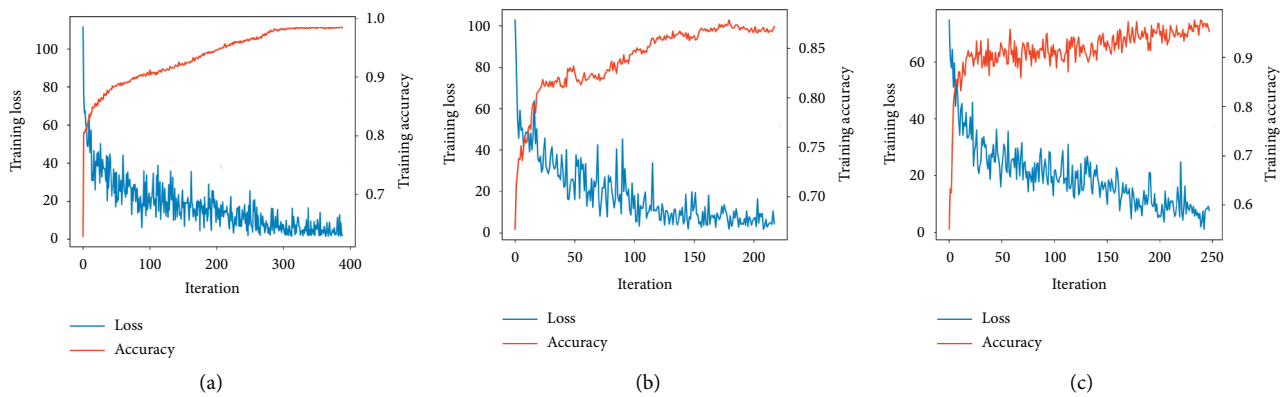


Figure 5: Trend of accuracy and loss with the number of iterations on (a) MNIST, (b) USPS, and (c) REUTERS-10K.

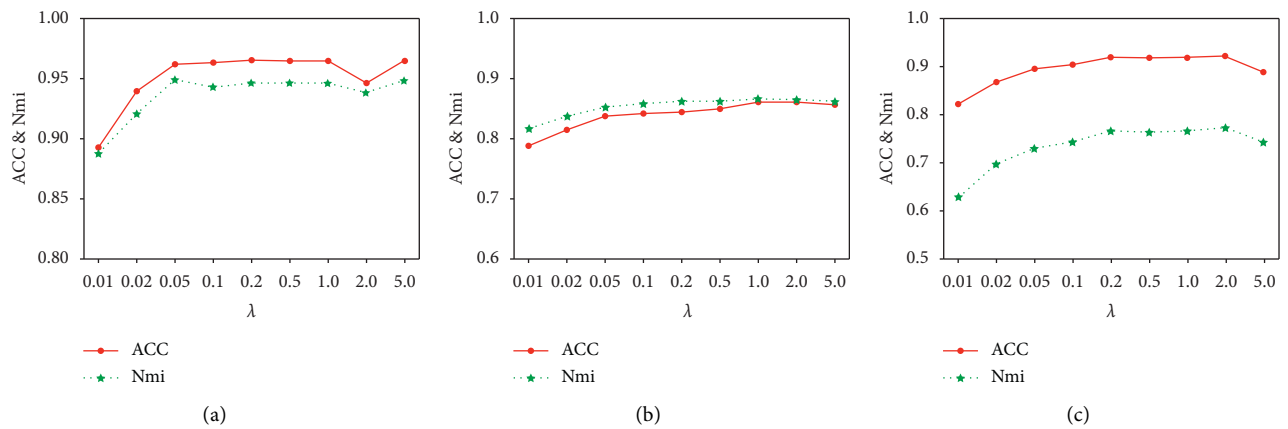


Figure 6: The effect of trade-off parameter λ on clustering performance on (a) MNIST, (b) USPS, and (c) REUTERS-10K.

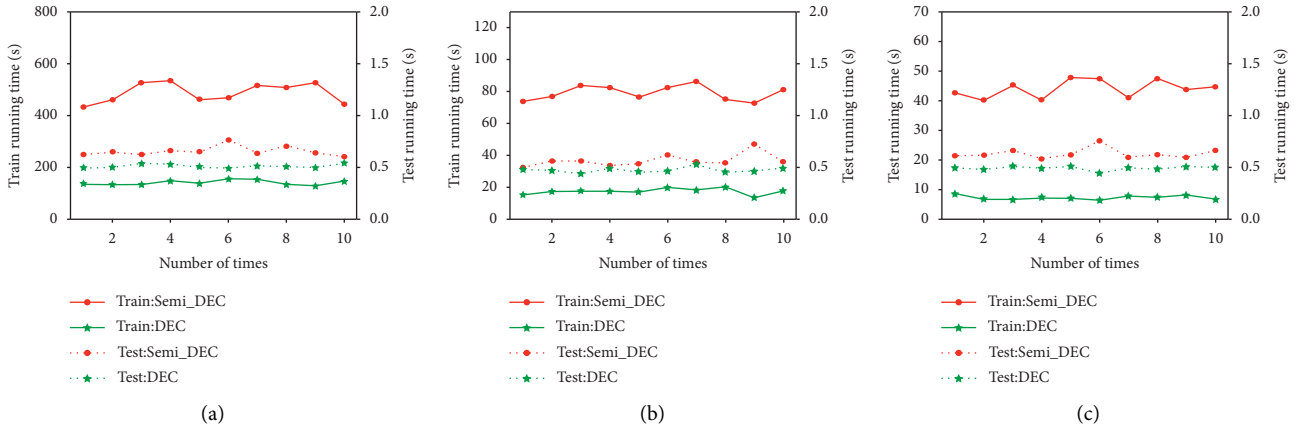


FIGURE 7: Running time statistics. The solid line represents the training process and dotted line represents the testing process. The circle represents the method in this paper, and the asterisk represents the DEC method. (a) MNIST, (b) USPS, and (c) REUTERS-10K.

that the method in this paper consumes more time in the training process than DEC. This is because the label adaptive strategy is added and the label loss needs to be calculated. But we think the limited time for training is worth it because we have got a big improvement in performance.

6. Conclusions

In this paper, a novel semisupervised deep embedded clustering method with adaptive labels is proposed to jointly learn cluster representation and assignment of data with the help of a priori knowledge. A deep semi-supervised clustering network is proposed, as well as a label adaptive strategy that can directly guide the clustering process by using the existing label information. Also, a joint optimization of the KL divergence loss and label loss in semisupervised deep clustering framework is designed to learn more powerful deep representation and more accurate cluster centres. Experimental results on MNIST, USPS, and REUSTER-10K show the method proposed in this paper has achieved significant performance improvement in both ACC and NMI, proving the effectiveness of the method. In the future, more efficient ways to use label information in the deep embedded clustering will be explored.

Data Availability

We perform experiment on two image datasets and one text dataset. The datasets used are commonly used public datasets, which are linked as follows: MNIST: http://yann.lecun.com/exdb/mnist/. USPS: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html. Reuters: http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61672123, Grant 61602083, and Grant 62002044, the Doctoral Scientific Research Foundation of Liaoning Province (20170520425), the Fundamental Research Funds for the Central Universities under Grant DUT20LAB136, Grant DUT20TD107, and Grant DUT15RC(3)100, and the China Scholarship Council.

References

[1] X. Li, H. Yin, K. Zhou, and X. Zhou, "Semi-supervised clustering with deep metric learning and graph embedding," *World Wide Web*, vol. 23, no. 2, pp. 781–798, 2020.

[2] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, "Next-generation big data analytics: state of the art, challenges, and future research topics," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1891–1899, 2017.

[3] W. Wang, Y. Wu, C. Tang, and M. Hor, "Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data," in *Proceedings of the 2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 445–451, Guangzhou, China, 2015.

[4] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

[5] V. Bureva, E. Sotirova, S. Popov, D. Mavrov, and V. Traneva, "Generalized net of cluster analysis process using STING: a statistical information grid approach to spatial data mining," in *Proceedings of the 12th International Conference Flexible Query Answering Systems (FQAS)*, London, UK, 2017.

[6] B. Lv, W. Hou, G. Liu et al., "A deep cfs model for text clustering," in *Proceedings of the 2018 International Conference on Internet of Things*, pp. 132–137, Halifax, Canada, 2018.

[7] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, vol. 48, pp. 478–487, New York, NY, USA, 2016.

[8] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image

classification," *IEEE Transactions on Geoscience and Remote Sensing*, p. 1, 2020.

[9] L. Zhao, T. Zhao, T. Sun, Z. Liu, and Z. Chen, "Multi-view robust feature learning for data clustering," *IEEE Signal Processing Letters*, vol. 27, pp. 1750–1754, 2020.

[10] W. Fan, C. Wang, and J. Lai, "SDenPeak: semi-supervised nonlinear clustering based on density and distance," in *Proceedings of the 2016 International Conference on Big Data Computing Service and Applications*, pp. 269–275, Oxford, UK, 2016.

[11] H. Li, J. Zhang, G. Shi, and J. Liu, "Graph-based discriminative nonnegative matrix factorization with label information," *Neurocomputing*, vol. 266, pp. 91–100, 2017.

[12] X. Li, Y. Wu, M. Ester et al., "Semi-supervised clustering in attributed heterogeneous information networks," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1621–1629, Perth, Australia, 2017.

[13] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2312–2318, Stockholm, Sweden, 2018.

[14] D. Hong, N. Yokoya, G.-S. Xia, J. Chanussot, and X. X. Zhu, "X-ModalNet: a semi-supervised deep cross-modal network for classification of remote sensing data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 12–23, 2020.

[15] P. Li, Z. Chen, J. Gao et al., "A deep fusion Gaussian mixture model for multiview land data clustering," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8880430, 9 pages, 2020.

[16] P. Li, Z. Chen, L. T. Yang, L. Zhao, and Q. Zhang, "A privacy-preserving high-order neuro-fuzzy c-means algorithm with cloud computing," *Neurocomputing*, vol. 256, pp. 82–89, 2017.

[17] A. Saha and S. Das, "Clustering of fuzzy data and simultaneous feature selection: a model selection approach," *Fuzzy Sets and Systems*, vol. 340, pp. 1–37, 2018.

[18] T. Yuan, W. Deng, J. Hu, Z. An, and Y. Tang, "Unsupervised adaptive hashing based on feature clustering," *Neurocomputing*, vol. 323, pp. 373–382, 2019.

[19] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[20] N. Heidari, Z. Moslehi, A. Mirzaei, and M. Safayani, "Bayesian distance metric learning for discriminative fuzzy c-means clustering," *Neurocomputing*, vol. 319, pp. 21–33, 2018.

[21] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1969–1976, Phoenix, AZ, USA, 2016.

[22] X. Wang, F. Nie, and H. Huang, "Structured doubly stochastic matrix for graph based clustering: structured doubly stochastic matrix," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1245–1254, San Francisco, CA, USA, 2016.

[23] D. Xie, Q. Gao, Q. Wang, and S. Xiao, "Multi-view spectral clustering via integrating global and local graphs," *IEEE Access*, vol. 7, pp. 31197–31206, 2019.

[24] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3861–3870, Sydney, Australia, 2017.

[25] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1753–1759, Melbourne, Australia, 2017.

[26] 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.

[27] S. Anand, S. Mittal, O. Tuzel, and P. Meer, "Semi-supervised kernel mean shift clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1201–1215, 2014.

[28] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, and F.-S. Gou, "Semi-supervised linear discriminant clustering," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 989–1000, 2014.

[29] D. Wang, X. Gao, and X. Wang, "Semi-supervised nonnegative matrix factorization via constraint propagation," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 233–244, 2016.

[30] V.-V. Vu, "An efficient semi-supervised graph based clustering," *Intelligent Data Analysis*, vol. 22, no. 2, pp. 297–307, 2018.

[31] A. Barhillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, vol. 6, pp. 937–965, 2005.

[32] S. C. H. Hoi, W. Liu, M. R. Lyu, and W. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2072–2078, New York, NY, USA, 2006.

[33] G. Niu, B. Dai, M. Yamada, and M. Sugiyama, "Information-theoretic semi-supervised metric learning via entropy regularization," *Neural Computation*, vol. 26, no. 8, pp. 1717–1762, 2014.

[34] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu, "Learning Bregman distance functions for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 478–491, 2012.

[35] Z. Yu, L. Li, J. Liu, J. Zhang, and G. Han, "Adaptive noise immune cluster ensemble using affinity propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3176–3189, 2015.

[36] J. Yi, L. Zhang, T. Yang, W. Liu, and J. Wang, "An efficient semi-supervised clustering algorithm with sequential constraints," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1405–1414, Sydney, Australia, 2015.

[37] S. Wei, Z. Li, and C. Zhang, "A semi-supervised clustering ensemble approach integrated constraint-based and metric-based," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service (ICIMCS)*, Zhangjiajie, China, 2015.

[38] D. Hong, L. Gao, N. Yokoya et al., "More diverse means better: multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, p. 1, 2020.

[39] G. Chen, "Deep transductive semi-supervised maximum margin clustering," 2015, https://arxiv.org/abs/1501.06237.

[40] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. H. Hoi, and Z. Xu, "Semi-supervised deep embedded clustering," *Neurocomputing*, vol. 325, pp. 121–130, 2019.

[41] 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.

Hindawi
Scientific Programming
Volume 2021, Article ID 6614873, 12 pages
https://doi.org/10.1155/2021/6614873

Research Article

Suspect Multifocus Image Fusion Based on Sparse Denoising Autoencoder Neural Network for Police Multimodal Big Data Analysis

Jin Wang and Yanfei Gao

Department of Public Security, Railway Police College, Zhengzhou 450000, China

Correspondence should be addressed to Jin Wang; wangjj815@163.com

Received 4 November 2020; Revised 11 December 2020; Accepted 28 December 2020; Published 7 January 2021

Academic Editor: Liang Zhao

In recent years, the success rate of solving major criminal cases through big data has been greatly improved. The analysis of multimodal big data plays a key role in the detection of suspects. However, the traditional multiexposure image fusion methods have low efficiency and are largely time-consuming due to the artifact effect in the image edge and other sensitive factors. Therefore, this paper focuses on the suspect multiexposure image fusion. The self-coding neural network based on deep learning has become a hotspot in the research of data dimension reduction, which can effectively eliminate the irrelevant and redundant learning data. In the case of limited field depth, due to the limited focusing depth of the camera, the focusing plane cannot obtain the global clear image of the target in the depth scene, which is prone to defocusing and blurring phenomena. Therefore, this paper proposes a multifocus image fusion based on a sparse denoising autoencoder neural network. To realize an unsupervised end-to-end fusion network, the sparse denoising autoencoder neural network is adopted to extract features and learn fusion rules and reconstruction rules simultaneously. The initial decision graph of the multifocus image is taken as a prior input to learn the rich detailed information of the image. The local strategy is added to the loss function to ensure that the image is restored accurately. The results show that this method is superior to the state-of-the-art fusion methods.

1. Introduction

Image fusion refers to the comprehensive processing of two or more complementary source images obtained from different sensors to obtain a new fused image, which enables the fused image to have higher credibility [1–4], clarity, and better understandability. In the case of limited field depth, due to the limited focusing depth of the camera, the focusing plane cannot obtain the global clear image of the target in the depth scene, which is prone to defocusing and blurring phenomena. Multifocus image fusion technology is to fuse multiple images with different focus positions in the same scene into a fully focused image with more information [5]. At present, multifocus image fusion algorithms can be divided into transform domain-based fusion method, space domain-based fusion method, and deep learning-based fusion method according to the fusion strategy.

The fusion method based on the transform domain generally uses a variety of decomposition tools to decompose the source image into multilevel coefficients and then designs different fusion rules according to the characteristics of each level coefficient [6, 7]. Finally, it performs the inverse multiscale transformation on the fused coefficients of each level to obtain the fused image. The design of transformation tools and the design of fusion rules play an important role in the fusion performance of transformation domain-based fusion methods.

Common transformation tools include curvelet transform (CVT) [8], nonsubsampled contourlet transform (NSCT) [9], Laplacian pyramid (LP) [10], low-pass pyramid, and gradient pyramid (GP) [11]. The fusion rules include maximization, weighted average, saliency, and active contour. The sparse representation (SR), higher-order singular value decomposition (HOSVD) [12], and other sparse

principal component analysis- (RPCA-) based multifocus image fusion methods [13] have attracted more attentions.

The fusion method based on the spatial domain can be divided into three types according to the different focus measurement objects: pixel-based, block-based, and region-based. The pixel-based multifocus image fusion method can extract the feature information from the source image and retain the original information to the greatest extent. It has the characteristics of high accuracy and strong robustness, which includes dense scale-invariant feature transform (DSIFT), guided filtering (GF), and image matting (IM). The multifocus image fusion method based on blocks and regions adopts some segmentation strategies to divide the source image into different blocks or regions and then selects more focus blocks or regions as part of the fused image by focus measurement [14]. The common focus measurement methods include image gradient and spatial frequency. The block size and segmentation algorithm can directly affect the visual effect of the fused image, which is prone to "block effect." Both transform domain-based fusion methods and spatial domain-based fusion methods require to manually design the fusion rules. However, complex image scenes limit the expressive ability of features and the robustness of fusion rules.

In order to improve the feature expression ability and the robustness of fusion rules, deep learning technology has been introduced into multifocus image fusion research [15–17]. Karim et al. [18] proposed a drone plane for monitoring and targeting street crime criminals based on real time image processing techniques. Liu et al. [19] used the multiscale Gaussian filter with different standard deviations to fuzzy process the random region on the gray image to simulate the multifocus image. By using supervised learning, the image was classified into focusing pixels and defocusing pixels, and the focus map with the same size as the input image was obtained. Then, the focus decision graph was generated by verifying the size and consistency of the focus map. Finally, based on the judging criteria, the weighted average strategy was used to obtain the fused images in the spatial domain. Tang et al. [20] proposed a multifocus image fusion method based on a pixel-wise convolutional neural network (P-CNN). This model used Cifar10 as the training set, and three kinds of pixels could be learned from adjacent pixel information: focusing pixel, defocusing pixel, and unknown pixel. After the source image was scored by PCNN, a scored matrix representing the focusing level of the pixel was formed. Then, by comparing the scores matrix of the two source images, then it obtained the decision graph. Finally, the weighted average value of the two input images was obtained according to the final decision graph filtered by a threshold. The model had excellent performance in real-time performance and fusion effect, but the limitation of supervised learning was that accurate label data could not be obtained for image fusion.

To further distinguish the private and public features in multifocus images, Luo et al. [21] proposed a joint convolution self-encoding network, which obtained the focus map based on the image features learned by the private branch and used the pixel-level weighted average rule to obtain the fully focused fused image. This method adopted unsupervised learning and did not need manually designed label and achieved ideal results on subjective evaluations and multiple objective evaluation. However, these methods only take advantage of CNN feature extraction and classification capability and still use the manually designed fusion rules, which makes the model unable to adjust the fusion strategy according to the application scenarios.

To further realize the self-learning of fusion rules and make full use of the feature extraction of CNN, combined with the prior knowledge of manual features, in this paper, a multifocus image fusion network with self-learning fusion rules is designed. The multifocus image and its initial decision graph are taken as the input of the network, so that the network can learn more accurate detailed information. The structural similarity index measure (SSIM) and local mean squared error (MSE) are used as loss functions to drive fusion rules.

The rest of this paper is organized as follows. Section 2 designs the proposed approach and, after that, Section 3 describes experimental results. Finally, Section 4 concludes the paper.

2. Proposed Multifocus Image Fusion

This paper first introduces the network structure of multifocus image fusion, then discusses the network fusion in detail, and finally discusses the loss function design.

2.1. Feature Extraction Network Based on Sparse Denoising Autoencoder Neural Network. Figure 1 shows the sparse denoising autoencoder neural network (SDNA-ENN).

The whole network is divided into the input layer, coding layer, fusion layer, decoding layer, and output layer. The input layer includes the initial decision graph of multifocus image A, multifocus image B, and multifocus image A. The coding layer includes 9 trainable convolutional layers with a convolution kernel size of 3×3, and each convolutional layer is followed by a ReLU layer. The coding layer can be divided into the private branch PriA, public branch ComA of multifocus image A, and the private branch PriB, and public branch ComB of multifocus image B, where PriA and PriB are used to extract the private features of the input images, respectively. ComA and ComB share weights to extract the common features from multiple input images. The fusion layer cascades the feature map output by PriA and PriB along the channel and then connects the cascaded feature map to the next trainable convolution layer with a convolution kernel size of 1×1. The output feature map of ComA and ComB is treated in the same way as PriA and PriB. The decoding layer consists of four trainable convolution layers with a convolution kernel size of 3×3, and the last convolutional layer is used to reconstruct the fully focused image. In this paper, a short connection is added to the public branch to solve the problem of gradient disappearance during the training process. Compared with the previous networks, this new network adds fusion units and uses short connections to improve the robustness of feature learning.

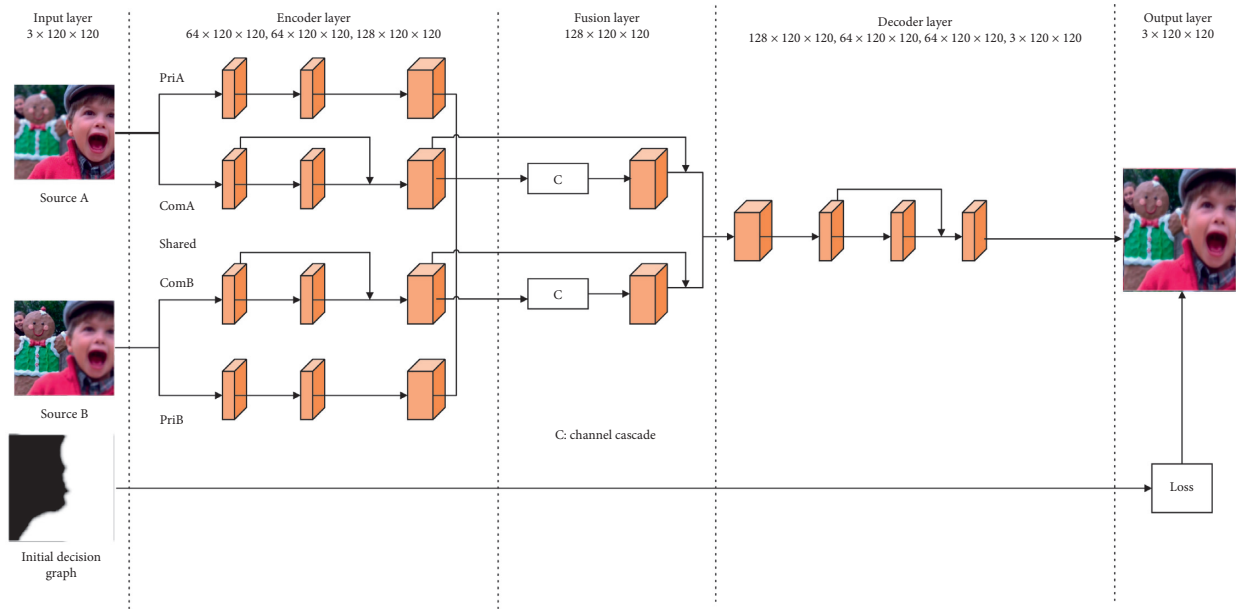


Figure 1: Structure of SDNA-ENN.

2.2. Fusion Layer Design. In the study of multifocus image fusion based on deep learning, the network fusion layer usually contains two methods that can be used to fuse the convolution features of multiple inputs:

(1) Cascade the convolution features of multiple inputs along the channel, and then fuse them with the next convolutional layer

(2) The multiple input convolution features are fused by the pixel-level fusion rule

The cascade fusion method stacks multiple inputs, so that the network can learn sufficient feature information.

The pixel-level fusion rule includes summation, taking large and mean value [22]. The fusion strategy can be selected according to the features of the data set. In multifocus images, because the pixel value of the image represents the information saliency, the proposed method in this paper introduces the mean rule on the basis of cascading fusion to ensure the diversity and accuracy of feature learning. The concrete realization of the fusion layer design includes weight initialization and weight constraint.

2.2.1. Weight Initialization. The weight initialization is to simulate the weighted average fusion rule, and the features extracted from the coding layer can be accurately fused by the reasonable weight assignment in the fusion layer. The output feature graphs of PriA, PriB, ComA, and ComB coding layers are splicing along the channel, followed by a trainable convolutional layer of 1×1. The first and $1 + p$ weight value of the k-th channel in the 1×1 convolutional layer is initialized to 0.5; that is,

$$W_k^l = W_k^{I+p} = 0.5, \quad I = 1, \ldots, 127; k = 1, 2 \ldots, 127, \quad (1)$$

where k is the channel number after the convolution operation. I is the filter number of the k-th channel. $P = 128$, which can be adjusted according to actual requirements. W_k^l is the I-th weight value of the k-th channel.

2.2.2. Weight Constraint. Because the weight value may appear numerical over-bounds phenomenon in the process of network iteration, the constraints are added to each weight value to realize the weight value fluctuation in the effective range. According to the mean value rule in the image fusion method, the sum of fusion coefficients of the two images is 1. However, the activation function of the training network adopts ReLU, for the k-th channel, $\sum_{I=0}^{p-1} W_k^I + \sum_{I=p}^{2p-1} W_k^{I+p} > 1$. Therefore, we make two improvements in this process. One is to improve the activation cost function, the second is to apply the minimum/maximum norm weight constraint to the $2p$ weights of the k-th channel in the fusion layer.

In order to make the activation units with fewer hidden layers represent the most effective features, through the traditional autoencoder neural network research, this paper proposes to add sparse restriction to the hidden neurons in the denoising autoencoder neural network (DAE), which can suppress most of the output neurons and use fewer activation units to represent features.

The sparse denoising autoencoder network structure consists of a sparse denoising autoencoder and a softmax classifier as shown in Figure 2. X represents the original data layer, $\tilde{X}$ represents the data layer with disturbing noise, and $\tilde{H}$ represents the hidden layer.

Specifically, assuming that the number of input samples is m. x represents the input. y represents the output. l represents the layer number of the neural network. s^l represents the neuron number in hidden layer l. Then the

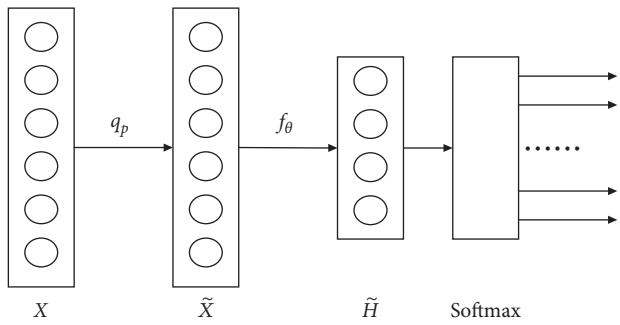


Figure 2: Structure of sparse denoising autoencoder neural network.

activation cost function of the sparse denoising autoencoder neural network is defined as follows:

$$J_{\mathrm{SDAE}}(w,b) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{w,b}(\widetilde{x})^{(i)} - y^{(i)}\right\|^2\right) + \beta\sum_{j=1}^{s^l}\mathrm{KL}\left(\rho\|\widehat{\rho}_j\right). \tag{2}$$

The residual of each neuron in the hidden layer is

$$\begin{cases} \delta_i^l = -(y_i - a_i)f'(z_i), & \text{when } l \text{ is output layer,} \\ \delta_i^l = \left[\left(\sum_{j=1}^{s_2} w_{ji}\delta_i^{l+1}\right) + \beta\left(-\dfrac{\rho}{\widehat{\rho}_j} + \dfrac{1-\rho}{1-\widehat{\rho}_j}\right)\right]f'(z_i), & \text{when } l \text{ is hidden layer.} \end{cases} \tag{3}$$

Then, the partial derivatives of weight and bias items are calculated as follows:

$$\begin{aligned} \nabla_{W^l}J(W,b) &= \frac{\partial}{\partial W_i^l}J_{\mathrm{SDAE}}(w,b) = a_j^l\delta_i^{l+1}, \\ \nabla_{b^l}J(W,b) &= \frac{\partial}{\partial b_i^l}J_{\mathrm{SDAE}}(w,b) = \delta_i^{l+1}. \end{aligned} \tag{4}$$

Then, we calculate the L2-norm of $2p$ weights in the k-th channel.

$$S_k = \sqrt{\sum_{I=0}^{p-1}\left(W_k^I\right)^2 + \sum_{I=p}^{2p-1}\left(W_k^{I+p}\right)^2}. \tag{5}$$

S_k is truncated in the range $(S_{\min}, S_{\max})$; that is,

$$S_t = \begin{cases} S_{\min}, & S_k < S_{\min}, \\ S_k, & S_{\min} < S_k < S_{\max}, \\ S_{\max}, & S_k > S_{\max}, \end{cases} \tag{6}$$

where $S_{\min}$ is the minimum L2-norm of input weight value. $S_{\max}$ is the maximum L2-norm of input weight value.

Finally, each weight value of the k-th channel is readjusted.

$$\begin{aligned} W_k^m &= W_k^m \times Z_k, \quad m = 0, 1, 2, \ldots, 2p-1, \\ Z_k &= \frac{\alpha \times S_t + (1-\alpha)\times S_k}{\gamma + S_k}, \end{aligned} \tag{7}$$

where W_k^m is the m-th weight value of the k-th channel and Z_k is the constraint range of the weight value. α is the proportion of constraint; when $\alpha = 1$, the constraint is strictly enforced, and when $\alpha < 1$, the weight must be adjusted for each step. In order to avoid gradient explosion, $\gamma = e^{-3}$. After weight initialization and constraint, the rules of the fusion layer are finally converted to

$$\widehat{f}_k(x,y) = W_k^I f_I(x,y) + W_k^{I+p} f_{I+p}(x,y). \tag{8}$$

2.3. The Design of Loss Function. In order to ensure that the network can learn the features of the input image accurately and effectively, the local strategy is added into the loss function, including local structure similarity and local mean square error.

2.3.1. Local Structure Similarity. Human visual system is more sensitive to structural loss and deformation. Therefore, the structural similarity index measure (SSIM) [23] can be used to intuitively compare the structural information of distorted images and original images. SSIM is mainly composed of three parts: relevancy, brightness, and contrast as shown in the following:

$$\mathrm{SSIM}(X,F) = \sum_{x,f}\frac{\left(2\mu_x\mu_f + C_1\right)\left(2\mu_x\mu_f + C_1\right)\left(2\mu_x\mu_f + C_1\right)}{\left(\mu_x^2 + \mu_f^2 + C_1\right)\left(\sigma_x^2 + \sigma_f^2 + C_2\right)\left(\sigma_x\sigma_f + C_3\right)}, \tag{9}$$

where $\mathrm{SSIM}(X, F)$ represents the structural similarity of source image X and fused image F. x and f represent the image blocks in the source image and the fused image, respectively. μ_x and σ_x represent the mean and standard deviation of the image X, respectively. μ_f and σ_f represent the mean and standard deviation of fused image F respectively. σ_{xf} represents the covariance of the source image and the fused image. C_1, C_2, and C_3 are the parameters used to stabilize the algorithm.

On the basis of SSIM, the corresponded region of image X is extracted by combining the initial decision graph X_m of the input image X.

$$\overline{X} = \min\left(X_m, X\right). \tag{10}$$

The initial decision graph corresponding to the input images A and B are X_A and X_B, respectively. According to (10), corresponding regions $\overline{A}$, $\overline{B}$, and $\overline{F}$ of images A and B and fused image F can be obtained, respectively. According to (9), $\mathrm{SSIM}(\overline{A}, t\overline{F})$ and $\mathrm{SSIM}(\overline{B}, t\overline{F})$ can be calculated.

2.3.2. Local Mean Square Error. Mean square error is used to measure the difference degree between the source image and the fused image. The mean square error is inversely proportional to the quality of the fused image. The smaller value denotes higher fusion quality. Its calculation formula is

$$\mathrm{MSE}(X, F) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(X(i, j) - F(i, j)\right)^2, \tag{11}$$

where $\mathrm{MSE}(X, F)$ represents the difference between the input image X and the fused image F.

According to (11), $\mathrm{MSE}(\overline{A}, t\overline{F})$ and $\mathrm{MSE}(\overline{B}, t\overline{F})$ can be obtained.

The final loss function of the proposed network is

$$L = \lambda_1 \left(\mathrm{SSIM}(\overline{A}, \overline{F}) + \mathrm{SSIM}(\overline{B}, \overline{F})\right) + \lambda_2 \left(\mathrm{MSE}(\overline{A}, \overline{F}) + \mathrm{MSE}(\overline{B}, \overline{F})\right), \tag{12}$$

where λ_1 and λ_2 represent the weights of local structure similarity and local mean square error, respectively. In this paper, λ_1 is used to adjust the similarity between the fused image and the source image. The larger λ_1 denotes the higher similarity between the fused image and the source image. λ_2 is used to enhance the focus area of the source image in the fused image. The larger λ_2 denotes the significant focus area of the source image. Based on the extensive experiments, this paper sets $\lambda_1 = 5$, $\lambda_2 = 5$, respectively.

3. Experiment and Analysis

In order to verify the performance of the proposed fusion method, we conduct comparison experiments with seven state-of-the-art fusion methods, namely, DE [24], NFBD [25], GDMC [26], LRRW [27], NNSR [28], CFM [29], and FRL-PCNN [30]. The experiment environment is MATLAB7a, Windows10, GPU TX1060, Memory 16G, and Intel(R) Core(TM) i7-67001. The Keras framework of Tensorflow is used for network training in this paper. All the comparison methods use the same parameters [31, 32]. Then, the detailed subjective and objective comparison and analysis are carried out on multiple multifocus images.

Because suspects are classified as the country secret data, this paper tests suspects and open datasets in the laboratory. The results are only from the public datasets. This paper conducts experiments on 60 pairs of multifocus images. 20 pairs are from the open-source dataset Lytro [33], the other 20 pairs have been widely used in the study of multifocus image fusion, and another 20 pairs are from actual suspect images. The sliding window method is adopted to take blocks with a stride length of 14. Each image in the dataset is divided into M image blocks with 224×224pixel. The initial decision graph acquisition in this paper consists of three parts: segmentation, mapping, and reprocessing. First, each image in the dataset is segmented into blocks with 4×4 pixel, and the spatial frequency is calculated. Then, the spatial frequency matrix is mapped to the original size of the source image, and the overlap part is processed with the mean value to obtain the spatial frequency map. The binary map is obtained by comparing the size. Finally, the initial decision graph of the network is obtained through consistency verification and guided filtering. The fusion results with different methods are shown in Figures 3–8.

To compare the fusion methods more intuitively, this paper selects a smaller region at a certain contour in each fused image, marks it with rectangular box, and gives an enlarged region. We give an analysis for image "disk." It can be seen from Figure 7 that the above methods can obtain fully focused images with good subjective vision. DE and NFBD present false information such as "artifact" in the edge of alarm clock. The fusion effect of IM is good, but there is a certain "Gibbs" phenomenon in the disk area, and some details are lost. GDMC shows fuzzy distortion in the local amplification region due to the emphasis on looking for boundaries and the focus metric is performed within a single block. The fusion results from LRRW, NNSR, CFM, and FRL-PCNN are good, but there is a slight "sag" on the left edge of the alarm clock.

Comparatively, the visual effect of the proposed method in this paper is similar to the subjective visual effect of other methods. It can be seen from the enlarged area in Figure 7 that the proposed method in this paper handles the details well, especially the edge area of the alarm clock is smooth and natural. A better fusion result is obtained. Since the initial decision graph of the focused image and the local strategy of the loss function are added into the network, the obtained fused image by the proposed method in this paper performs well in the retention of key information and is suitable for human visual perception. Figures 3–6 and 8 show the fusion results of the other 5 pairs of multifocus images in various fusion methods. As can be seen from the figures, all the methods can better fuse the multifocus image to some extent. Compared with other methods, the proposed method achieves better fusion results.

To objectively evaluate the results of each fusion method, this paper uses the evaluation index: entropy (EN), Q_W proposed by Piella and Heijmans, correlation-coefficient (CC), and Visual Information Fidelity (VIFF) to verify the

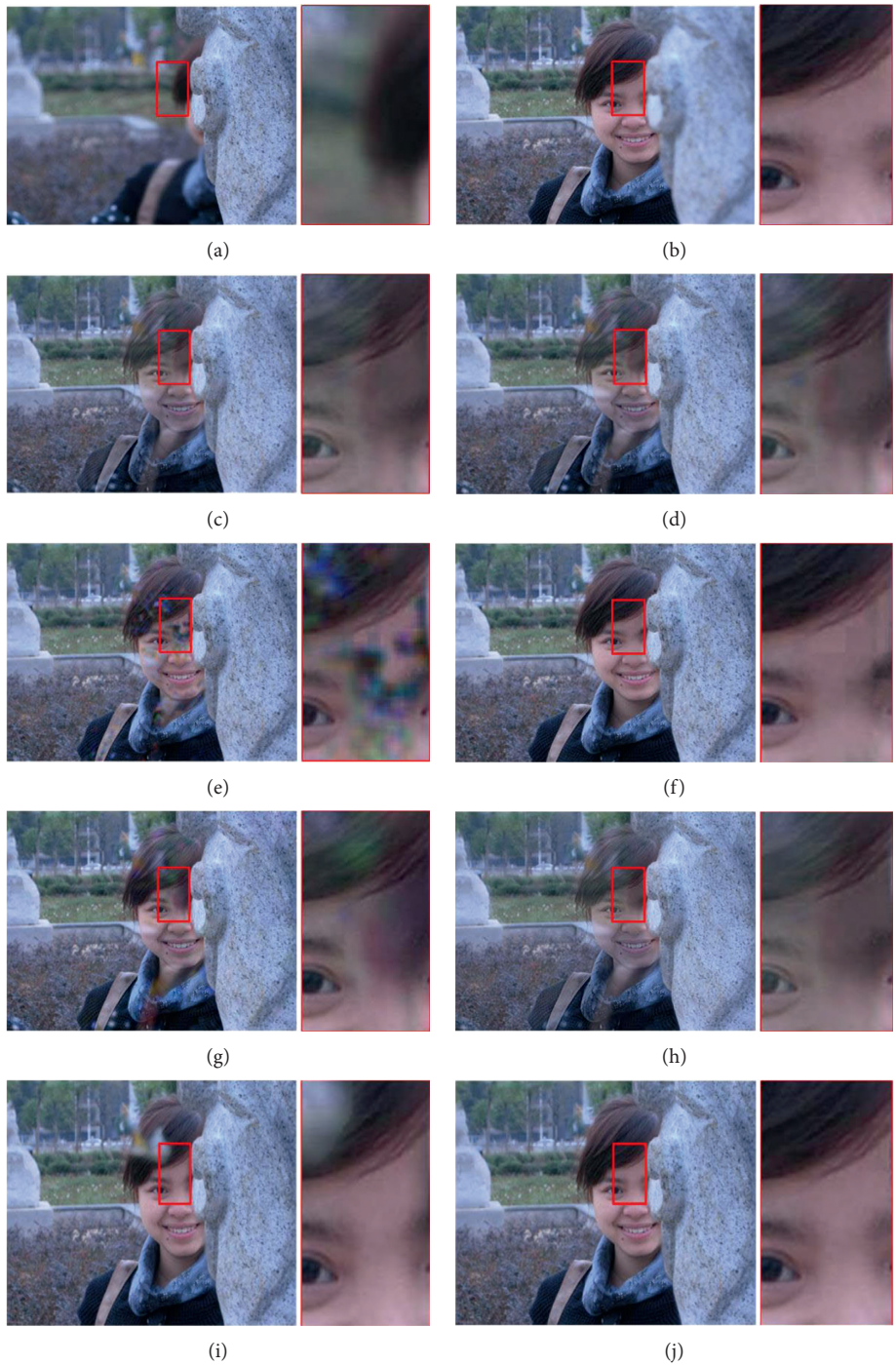


Figure 3: The "girl" source images and result images with different algorithms. (a, b) Source images; (c–j) the fusion images of DE, NFBD, GDMC, LRRW, NNSR, CFM, FRL-PCNN, and proposed method.

effectiveness of the proposed method. Entropy is an index based on information theory, which is used to reflect the amount of information in an image. If the entropy value is relatively large, it indicates that the fused image contains relatively more information. Q_W is a variant of the universal image quality index, which explores the position and size of distorted pixels by assigning high weights to visual saliency areas. The greater Q_W denotes the better fusion effect. The correlation coefficient measures the correlation between the source image and the fused image. The correlation value is positively correlated with the fusion effect. The VIFF is an index that simulates the subjective vision of human eyes to measure the fidelity of fused image. It includes four steps: partitioning, evaluation, calculating the fidelity of subband, and calculating the total fidelity. The higher VIFF presents the lower the distortion between the fused image and the source image. In order to ensure the fairness of objective evaluation, all indexes use the same parameters.

Figure 4: The "tree" source images and result images with different algorithms. (a, b) Source images; (c–j) the fusion images of DE, NFBD, GDMC, LRRW, NNSR, CFM, FRL-PCNN, and proposed method.

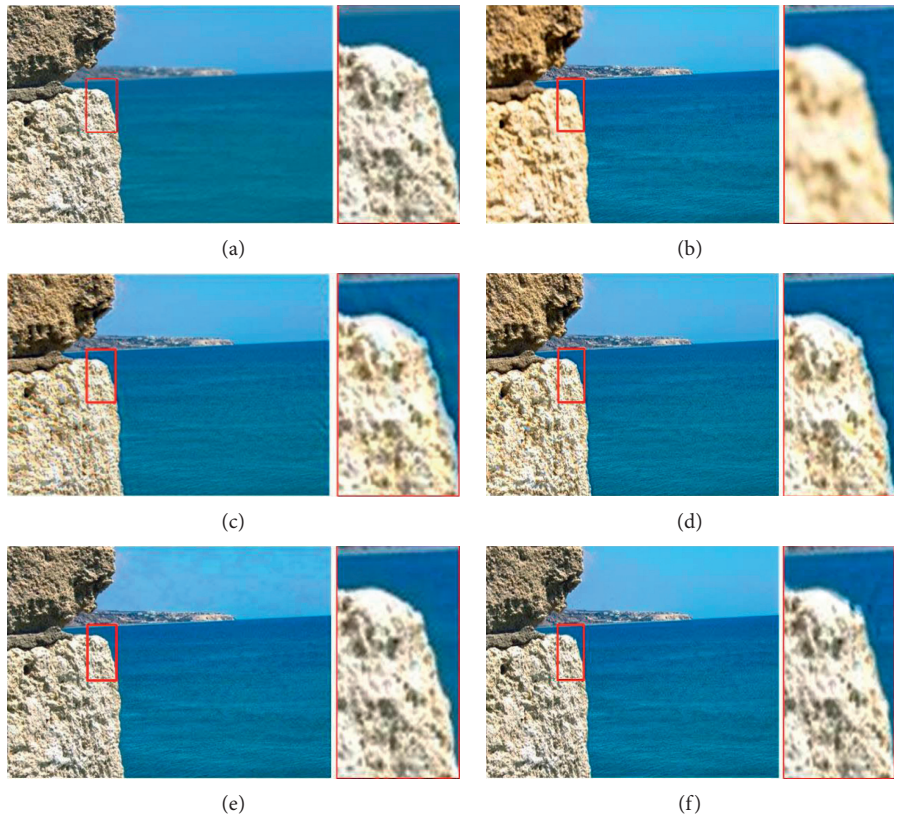
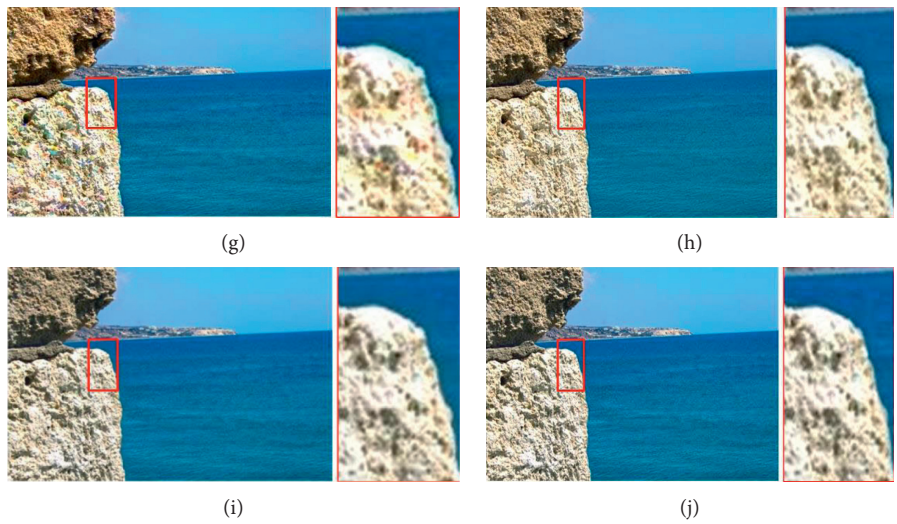


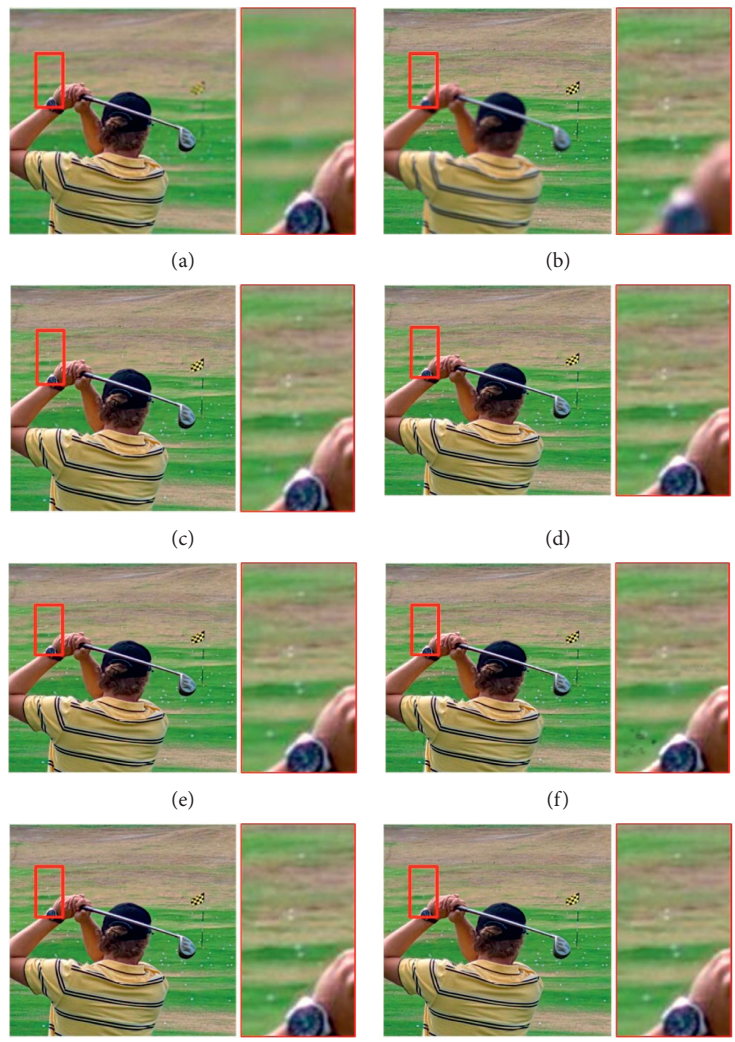
Figure 5: Continued.



(g) (h)

(i) (j)

Figure 5: The "sea" source images and result images with different algorithms. (a, b) Source images; (c–j) the fusion images of DE, NFBD, GDMC, LRRW, NNSR, CFM, FRL-PCNN, and proposed method.



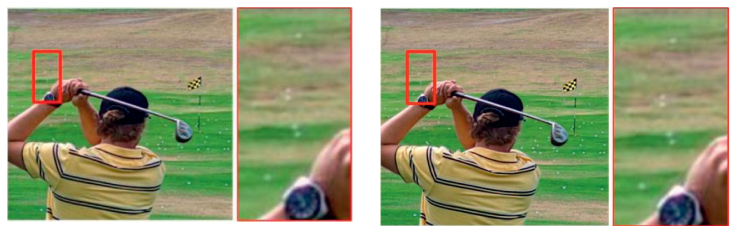
(a) (b)

(c) (d)

(e) (f)

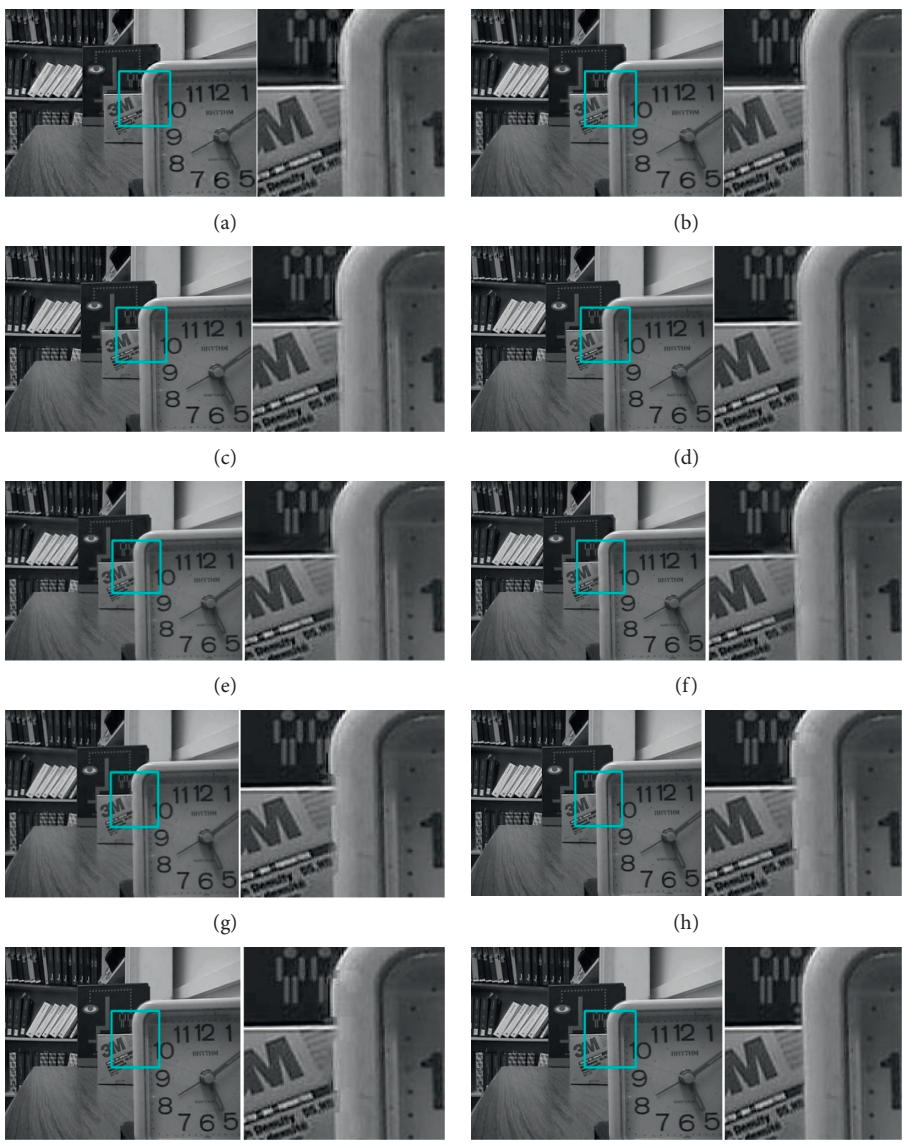
(g) (h)

Figure 6: Continued.



(i) (j)

Figure 6: The "golf" source images and result images with different algorithms. (a , b) source images; (c–j) the fusion images of DE, NFBD, GDMC, LRRW, NNSR, CFM, FRL-PCNN, and proposed method.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 7: The "disk" source images and result images with different algorithms. (a, b) Source images; (c–j) the fusion images of DE, NFBD, GDMC, LRRW, NNSR, CFM, FRL-PCNN, and proposed method.

Figure 8: The "temple" source images and result images with different algorithms. (a, b) Source images; (c–j) the fusion images of DE, NFBD, GDMC, LRRW, NNSR, CFM, FRL-PCNN, and proposed method.

Table 1: The objective metrics of different fusion approaches for image "girl".

Method	EN	Q_W	CC	VIFF
DE	7.8668	0.8654	0.9811	0.7298
NFBD	7.8651	0.8552	0.9824	0.7398
GDMC	7.8662	0.8741	0.9825	0.7399
LRRW	7.8674	0.8742	0.9822	0.7435
NNSR	7.8662	0.8746	0.9827	0.7471
CFM	7.8673	0.8755	0.9836	0.7488
FRL-PCNN	7.8695	0.8768	0.9839	0.7527
Proposed	**7.8698**	**0.8879**	**0.9857**	**0.7879**

Table 2: The objective metrics of different fusion approaches for image "tree".

Method	EN	Q_W	CC	VIFF
DE	7.4493	0.8894	0.9721	0.8625
NFBD	7.4512	0.8942	0.9738	0.8756
GDMC	7.4528	0.8953	0.9742	0.8747
LRRW	7.4538	0.8974	0.9758	0.8827
NNSR	7.4688	0.8995	0.9763	0.8875
CFM	7.4848	0.9027	0.9778	0.8957
FRL-PCNN	7.5128	0.9037	0.9781	0.8998
Proposed	**7.6456**	**0.9122**	**0.9829**	0.9025

Table 3: The objective metrics of different fusion approaches for image "sea".

Method	EN	Q_W	CC	VIFF
DE	7.1803	0.9237	0.9617	0.9187
NFBD	7.1809	0.9242	0.9618	0.9238
GDMC	7.1814	0.9248	0.9624	0.9247
LRRW	7.1816	0.9257	0.9627	0.9355
NNSR	7.1825	0.9259	0.9633	0.9371
CFM	7.1897	0.9265	0.9638	0.9382
FRL-PCNN	7.1907	0.9277	0.9688	0.9407
Proposed	**7.1938**	**0.9359**	**0.9729**	**0.9477**

Table 4: The objective metrics of different fusion approaches for image "golf".

Method	EN	Q_W	CC	VIFF
DE	7.2714	0.9286	0.9824	0.9367
NFBD	7.2728	0.9288	0.9827	0.9382
GDMC	7.2734	0.9297	0.9828	0.9341
LRRW	7.2741	0.9314	0.9831	0.9345
NNSR	7.2832	0.9319	0.9833	0.9346
CFM	7.2835	0.9324	0.9837	0.9452
FRL-PCNN	7.2839	0.9328	0.9841	0.9557
Proposed	**7.2847**	**0.9342**	**0.9852**	**0.9722**

Table 5: The objective metrics of different fusion approaches for image "disk".

Method	EN	Q_W	CC	VIFF
DE	7.2654	0.9271	0.9762	0.8692
NFBD	7.2693	0.9317	0.9768	0.8714
GDMC	7.2754	0.9345	0.9769	0.8725
LRRW	7.2768	0.9368	0.9701	0.8736
NNSR	7.2791	0.9372	0.9715	0.8745
CFM	7.2836	0.9408	0.9718	0.8767
FRL-PCNN	7.2914	0.9412	0.9724	0.8771
Proposed	**7.2988**	**0.9477**	**0.9783**	**0.8859**

Table 6: The objective metrics of different fusion approaches for image "temple".

Method	EN	Q_W	CC	VIFF
DE	7.2563	0.9157	0.9687	0.8774
NFBD	7.2566	0.9188	0.9688	0.8793
GDMC	7.2569	0.9236	0.9702	0.8817
LRRW	7.2572	0.9237	0.9711	0.8824
NNSR	7.2574	0.9239	0.9714	0.8829
CFM	7.2579	0.9385	0.9718	0.8836
FRL-PCNN	7.2583	0.9427	0.9724	0.8867
Proposed	**7.2594**	**0.9507**	**0.9791**	**0.8958**

Tables 1–6 display the fusion objective evaluation results on 6 pairs of multifocus images with the eight fusion methods. As can be seen from the tables, the proposed fusion method has obvious advantages over other fusion methods in terms of the fusion indexes. In general, the proposed method achieves the best results in terms of Q_W, CC, EN, VIFF, and average accuracy index, indicating that this new algorithm is an effective fusion method.

4. Conclusions

In this paper, an end-to-end unsupervised multifocus image fusion algorithm based on sparse denoising autoencoder neural network is proposed. Combined with the prior knowledge of multifocus image, the network can learn accurate image details. Reasonable weight initialization and weight constraint are designed in the fusion layer. Local structure similarity and local mean square error strategies are used in the loss function to drive the fusion unit to learn the fusion rules effectively. Experimental results show that the proposed method not only can realize the fusion rules in the fusion process of self-learning. In addition, good results can be obtained in subjective vision and objective evaluation. It is of great significance to further understand the multifocus image fusion mechanism based on deep learning and to study the general multi-modal image fusion framework. In the future, more newest deep learning methods will be utilized to analyze the multifocus image fusion.

Data Availability

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

Conflicts of Interest

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

References

[1] W. Zhao, H. Lu, and D. Wang, "Multisensor image fusion and enhancement in spectral total variation domain," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 866–879, 2018.

[2] S. Yin and Y. Zhang, "Singular value decomposition-based anisotropic diffusion for fusion of infrared and visible images," *International Journal of Image and Data Fusion*, vol. 10, no. 2, pp. 146–163, 2019.

[3] Z. Zhu, H. Yin, Y. Chai, Y. Li, and G. Qi, "A novel multimodality image fusion method based on image decomposition and sparse representation," *Information Sciences*, vol. 432, pp. 516–529, 2018.

[4] L. Bungert, D. Coomes, M. Ehrhardt et al., "Blind image fusion for hyperspectral imaging with the directional total variation," *Inverse Problems*, vol. 34, no. 4, 2018.

[5] 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*, p. 1, 2020.

[6] J. Li, B. Li, and Y. Jiang, "An infrared and visible image fusion algorithm based on LSWT-NSST," *IEEE Access*, vol. 8, pp. 179857–179880, 2020.

[7] J. Wang, C. Qin, and X. Zhang, "A multi-source image fusion algorithm based on gradient regularized convolution sparse representation," *Systems Engineering and Electronics*, vol. 31, no. 3, pp. 447–459, 2020.

[8] L. Dong, Q. Yang, H. Wu, H. Xiao, and M. Xu, "High quality multi-spectral and panchromatic image fusion technologies based on curvelet transform," *Neurocomputing*, vol. 159, pp. 268–274, 2015.

[9] M. Nazrudeen and M. Rajalakshmi, "CT and MRI image fusion using non-subsampled contourlet transform," in *Proceedings of the International Conference on Communication and Computer Networks of the Future*, Tokyo, Japan, June 2014.

[10] X. Wang, S. Yin, K. Sun et al., "KFC-CNN: modified Gaussian kernel fuzzy C-means and convolutional neural network for apple segmentation and recognition," *Journal of Applied Science and Engineering*, vol. 23, no. 3, pp. 555–561, 2020.

[11] M. J. Li, Y. B. Dong, and X. L. Wang, "Image fusion algorithm based on gradient pyramid and its performance evaluation," *Applied Mechanics and Materials*, vol. 525, pp. 715–718, 2014.

[12] J. Liang, Y. He, D. Liu, and X. Zeng, "Image fusion using higher order singular value decomposition," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 21, no. 5, pp. 2898–2909, 2012.

[13] T. Wan, C. Zhu, and Z. Qin, "Multifocus image fusion based on robust principal component analysis," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1001–1008, 2013.

[14] L. Zhao, T. Zhao, T. Sun, Z. Liu, and Z. Chen, "Multi-view robust feature learning for data clustering," *IEEE Signal Processing Letters*, vol. 27, pp. 1750–1754, 2020.

[15] S. Yin, H. Li, and L. Teng, "Airport detection based on improved faster RCNN in large scale remote sensing images," *Sensing and Imaging*, vol. 21, no. 1, 2020.

[16] S. Yin and H. Li, "Hot region selection based on selective search and modified fuzzy C-means in remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5862–5871, 2020.

[17] 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.

[18] S. Karim, Y. Zhang, A. A. Laghari, and M. R. Asif, "Image processing based proposed drone for detecting and controlling street crimes," in *Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pp. 1725–1730, Chengdu, China, October 2017.

[19] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Information Fusion*, vol. 36, pp. 191–207, 2017.

[20] H. Tang, B. Xiao, W. Li, and G. Wang, "Pixel convolutional neural network for multi-focus image fusion," *Information Sciences*, vol. 433-434, pp. 125–141, 2018.

[21] X. Luo, J. Zhang, and Q. Dai, "A regional image fusion based on similarity characteristics," *Signal Processing*, vol. 92, no. 5, pp. 1268–1280, 2012.

[22] 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.

[23] L. Zhao, C. Mo, T. Sun, and W. Huang, "Aero engine gas-path fault diagnose based on multimodal deep neural networks," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8891595, 10 pages, 2020.

[24] L. Zhang, G. Zeng, and J. Wei, "Adaptive region-segmentation multi-focus image fusion based on differential evolution," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 3, 2019.

[25] Y. Yang, Y. Zhang, J. Wu, L. Li, and S. Huang, "Multi-focus image fusion based on a non-fixed-base dictionary and multi-measure optimization," *IEEE Access*, vol. 7, pp. 46376–46388, 2019.

[26] X. Bai, M. Liu, Z. Chen, P. Wang, and Y. Zhang, "Multi-focus image fusion through gradient- based decision map construction and mathematical morphology," *IEEE Access*, vol. 4, pp. 4749–4760, 2016.

[27] Z. Ji, X. Kang, K. Zhang, P. Duan, and Q. Hao, "A two-stage multi-focus image fusion framework robust to image misregistration," *IEEE Access*, vol. 7, pp. 123231–123243, 2019.

[28] Q. Zhang, G. Li, Y. Cao et al., "Multi-focus image fusion based on non-negative sparse representation and patch-level consistency rectification," *Pattern Recognition*, vol. 104, Article ID 107325, 2020.

[29] L. He, X. Yang, L. Lu et al., "A novel multi-focus image fusion method for improving imaging systems by using cascade-forest model," *EURASIP Journal on Image and Video Processing*, vol. 5, no. 1, p. 2020, 2020.

[30] K. He, D. Zhou, X. Zhang et al., "Multi-focus image fusion combining focus-region-level partition and pulse-coupled neural network," *Soft Computing—A Fusion of Foundations, Methodologies and Applications*, vol. 23, no. 13, 2019.

[31] A. Laghari, H. He, A. Khan, and S. Karim, "Impact of video file format on quality of experience (QoE) of multimedia content," *3D Research*, vol. 9, no. 3, p. 39, 2018.

[32] K. Shahid, Y. Zhang, S. Yin, A. Laghari, and 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.

[33] H. T. Mustafa, J. Yang, and M. Zareapoor, "Multi-scale convolutional neural network for multi-focus image fusion," *Image and Vision Computing*, vol. 85, pp. 26–35, 2019.

Hindawi
Scientific Programming
Volume 2020, Article ID 8899957, 11 pages
https://doi.org/10.1155/2020/8899957

Research Article

A Crossover Comparison of the Sensitivity and the Specificity between BIS and AEP in Predicting Unconsciousness in General Anesthesia

Haitao Yang, Guan Wang, Jinxia Gao, and Jie Liu

Department of Anesthesiology, The Second Hospital of Dalian Medical University, Dalian 116027, China

Correspondence should be addressed to Jinxia Gao; gaojinxia2006@163.com and Jie Liu; liujaye@hotmail.com

Received 30 September 2020; Revised 27 November 2020; Accepted 18 December 2020; Published 29 December 2020

Academic Editor: Liang Zhao

Background. There is an increasing concern of awareness and recall during general anesthesia for both the patient and the anesthetist. The bispectral index (BIS) is used to assess the level of sedation and depth of anesthesia and detect consciousness in different anesthetic drugs. Middle-latency auditory evoked potentials (AEPs) also quantify action of anesthetic drugs and detect the transition from consciousness to unconsciousness. We aim to compare the sensitivity and specificity between BIS and AEP in predicting unconsciousness in inhalational sevoflurane anesthesia and intravenous propofol anesthesia. *Methods*. Totally, 40 patients were randomly allocated into two groups: propofol or sevoflurane group. In the propofol group, anesthesia was induced with target-controlled infusion propofol. In the sevoflurane group, anesthesia was induced by increasing concentrations of sevoflurane. There were 3 end points during induction: sedation, unconsciousness, and anesthesia. Target and effect-site concentrations of propofol, end-tidal concentration of sevoflurane, and BIS and AEP were recorded at each stage. *Results*. We obtained good EC50 with both monitors, at which there is a 50% chance that the patient has reached the end point, but the index variation was affected by the anesthetic technique. Propofol had higher correlations with stage of anesthesia, BIS, and AEP than sevoflurane. BIS had higher correlations with depth of anesthesia than AEP, but we did not find an anesthetic depth monitor that had high sensitivity and specificity and is not affected by the anesthetic technique. *Conclusions*. The prediction powers of BIS and AEP do not seem as good as some papers mentioned.

1. Introduction

Awareness and recall during general anesthesia, which are unintended accidental, represent failure of successful anesthesia and cause a serious complication of general anesthesia that is feared by patients and anesthetists alike [1–3]. It is difficult to describe and identify return to consciousness, so the reported incidence rates vary widely. Evidence suggests that the overall risk of awareness during anesthesia is between 0.1 and 0.5% [2, 4–6], and awareness has been considered as a potentially important factor for the occurrence of some diseases in patients, such as severe emotional distress and posttraumatic stress disorder [4, 6–8]. It also has important professional, personal, and financial consequences for the anesthetists [8–11].

The bispectral index (BIS), derived from electroencephalogram, is the most commonly used and accepted monitor for assessing the level of sedation and depth of anesthesia [10, 12–15]. BIS predicts movement in response to surgery and detects consciousness under different anesthetic drugs [15–18]. It is also a tool that may reduce the incidence of unexpected recall [10, 12, 13, 18].

Middle-latency auditory evoked potentials (AEPs) also quantify the action of anesthetic drugs and detect the transition from consciousness to unconsciousness [19–23]. The AEP index (AEP) is a dimensionless number scaled from 100 (awake) to 0 and a mathematically derived variable measuring the amplitude and latency of the cortical mid-latency auditory evoked potential that occurs in response to sound (a "click") [21, 23, 24].

Sevoflurane inhalational and propofol intravenous anesthesia are two widely used anesthetic techniques. However, there are no reports about the comparison of ability of predicting the awareness by BIS or AEP during these two anesthesia techniques. Herein, this study is designed to compare the sensitivity and the specificity between BIS and AEP in predicting unconsciousness with sevoflurane inhalational and propofol intravenous anesthesia.

2. Materials and Methods

The study was approved by the institutional review board. Unpremedicated patients who had given informed consent were recruited into the study. Demographic data and ASA classification were recorded. Routine monitoring plus monitoring for BIS and AEPi was established before the induction of anesthesia. Awake values for BIS and AEPi were recorded before the induction of anesthesia. Patients breathed oxygen through a standard anesthetic breathing circuit during induction. Patients were randomised into propofol or sevoflurane groups. There were 3 end points during induction:

(1) Sedation: patent was asleep and responded to gentle shaking or loud auditory stimulus (stage 4 of Ramsay scale).

(2) Loss of consciousness: patient showed no response to verbal command and loss of eyelash reflex.

(3) Anesthesia: patient gave no purposeful movement on tetanic stimulation to the ulnar nerve (50 Hz, 80 mA, 0.25 ms pulses) at the wrist using a constant current peripheral nerve stimulator.

The BIS and AEPi were recorded at each stage. In the propofol group, anesthesia was induced with target-controlled infusion (TCI) propofol. The TCI was initially set at $1\,\mu g{\cdot}l^{-1}$ and increased by $0.5\,\mu g{\cdot}l^{-1}$ every 2 minutes until anesthesia. Target and effect-site concentrations of propofol were recorded at each end point. In the sevoflurane group, anesthesia was induced by increasing concentrations of sevoflurane. End-tidal concentration of sevoflurane was recorded at each end point.

2.1. Anesthesia Induction. Routine monitoring plus monitoring for BIS and AEP was established before the induction of anesthesia. Awake values for BIS and AEP were recorded before the induction of anesthesia. Patients breathed oxygen through a standard anesthetic breathing circuit during induction. Patients were randomised into propofol or sevoflurane groups. There were 3 end points during induction: (1) sedation: patient was asleep and responded to gentle shaking or loud auditory stimulus (stage 4 of Ramsay scale); (2) unconsciousness: patient showed no response to verbal command and loss of eyelash reflex; (3) anesthesia: patient gave no purposeful movement on tetanic stimulation to the ulnar nerve (50 Hz, 80 mA, 0.25 ms pulses) at the wrist using a constant current peripheral nerve stimulator. The BIS and AEP were recorded at each stage. In the propofol group, anesthesia was induced with target-controlled infusion (TCI) of propofol. The TCI was initially set at $1\,\mu g{\cdot}l^{-1}$ and increased by $0.5\,\mu g{\cdot}l^{-1}$ every 2 minutes until anesthesia. Target and effect-site concentrations of propofol were recorded at each end point. In the sevoflurane group, anesthesia was induced by increasing concentrations of sevoflurane. End-tidal concentration of sevoflurane was recorded at each end point.

2.2. Statistical Analysis. GraphPad Prism version 5 (GraphPad Software, Inc) was used for data analysis. Demographic data were analyzed by the chi-square test and *t*-test. Haemodynamic data were analyzed by repeated measures analysis of variance and post hoc pair-wise comparison for difference stages of anesthesia. Spearman correlation analysis, logistic regression analysis, receiver operating characteristic (ROC) analysis, sensitivity and specificity, and prediction probability (P_K) were used for analyzing the depth of anesthesia, drug concentration, BIS, and AEP. $P < 0.05$ was considered to have statistically significant difference.

3. Results

3.1. Patient Characteristics. Forty-two patients were assessable for intraoperative BIS and AEP data, including 22 patients with sevoflurane anesthesia and 20 patients with propofol anesthesia. Two patients of the sevoflurane group were censored because of the unreasonable high BIS and AEP in the anesthesia stage. One patient of the sevoflurane group swapped the effect-site concentrations on sedation and unconsciousness because of the unreasonable high concentrations (4 and 4.6 mcg/ml) in the sedation stage. No significant difference in gender, height, weight, smoking history, alcohol intake, pain in sedation stage, or American society of Anesthesiologists status was found between two groups (Table 1).

3.2. Haemodynamic Data. Systolic blood pressure (SBP), heart rate (HR), and respiratory data (RR) in two groups were analyzed in four stages: base, sedation, unconsciousness, and anesthesia (Tables 2 and 3 and Figure 1).For SBP, time effect was significantly different at the 0.05 level of significance. Time and group interaction effect was significantly different at the 0.01 level of significance. On the propofol group, the SBP on all the other stages was significantly different from the baseline ($P = 0.0003$, <0.0001, and <0.0001 in sedation, unconsciousness, and anesthesia stages, respectively). The SBP on the sedation stage was also significantly different from the SBP on the unconsciousness and anesthesia stages. On the sevoflurane group, the SBP on all the other stages was significantly different from the baseline ($P = 0.0354$, 0.0053, and 0.0031 in sedation, unconsciousness, and anesthesia stages, respectively). The SBP on the sedation stage was significantly different from the SBP on unconsciousness and anesthesia stages.

On HR, both time effect and group interaction effects were significantly different at the 0.05 level of significance. On the propofol group, the heart rates on all the other stages were significantly different from the baseline heart rate

Table 1: Patient characteristics.

	Mean (SD) [range]/counts		
	Propofol group	Sevoflurane group	P value
Patient no.	20	20	N.A.
Age (years old)	27 (8.6) [17–46]	28 (11.3) [18–49]	0.7422
Weight (Kg)	60 (13.7) [41–85]	56 (7.1) [46–69]	0.3466
Sex ratio (male :female)	10 : 10	12 : 8	0.5250
ASA grading (I: II)	18 : 2	18 : 2	1.0000
Smoker	2	2	1.0000
Alcohol (no: occasional)	19 : 1	19 : 1	1.0000
Pain in sedation stage (none: mild: moderate)	11 : 8 : 1	No data	N.A.

ASA, American society of Anesthesiologists; N.A., not applicable.

Table 2: Haemodynamic data.

	Stage	Mean ± SD		P values
		Propofol	Sevoflurane	
Systolic blood pressure (SBP) in HHmg	Base	117.6 ± 12.7	117.1 ± 15.3	Group: 0.4542 Time: <0.0001** Interact: 0.0542
	Sed	112.2 ± 11.8	112.5 ± 11.1	
	Uncon	105.8 ± 8.8	109.4 ± 10.4	
	Anes	102.0 ± 7.4	108.4 ± 11.2	
Heart rate (HR) in beat $\min^{-1}$	Base	80.5 ± 10.7	79.6 ± 10.4	Group: 0.0784 Time: 0.0596 Interact: 0.0118**
	Sed	76.7 ± 8.6	81.0 ± 11.4	
	Uncon	72.2 ± 10.1	79.2 ± 13.1	
	Anes	72.9 ± 11.7	84.4 ± 14.9	
Respiratory rate (RR) in breaths $\min^{-1}$	Base	18.3 ± 3.2	17.0 ± 3.1	Group: 0.0964 Time: 0.4993 Interact: 0.9212
	Sed	17.9 ± 2.6	16.6 ± 2.6	
	Uncon	18.2 ± 2.4	16.7 ± 2.7	
	Anes	18.2 ± 2.7	17.1 ± 2.7	

**Significant at 0.05. Notes: repeated measures analysis of variance was applied. Interact = group*time interaction effect.

Table 3: P values for post hoc pair-wise comparisons.

	P values		
	Baseline	Sedation	Unconsciousness
SBP			
Time effect in propofol group			
Sedation	0.0003**	—	—
Unconsciousness	<.0001**	<.0001**	—
Anesthesia	<.0001**	0.0001**	0.0167
Time effect in sevoflurane group			
Sedation	0.0354	—	—
Unconsciousness	0.0053**	0.0493	—
Anesthesia	0.0031**	0.0355	0.4102
HR			
Time effect in propofol group			
Sedation	0.0266	—	—
Unconsciousness	0.0034**	0.0133	—
Anesthesia	0.0188	0.0610	0.6775
Time effect in sevoflurane group			
Sedation	0.5275	—	—
Unconsciousness	0.8083	0.1995	—
Anesthesia	0.2306	0.1517	0.0122**

Adjusted $\alpha' = 0.0125$ (=0.05/4) for post hoc comparisons ** $P < 0.01$.

($P = 0.0266$, 0.0034, and 0.0188 in sedation, unconsciousness, and anesthesia stages, respectively). The heart rate on the sedation stage was also significantly different from the heart rate on the unconsciousness stage ($P = 0.0133$). On the sevoflurane group, there were not significantly different in all different stages.

3.3. Drug Concentrations, BIS, and AEP

3.3.1. Descriptive Statistics of Drug Concentrations, BIS, and AEP at Different Stages of Induction. redicted blood and effect-site propofol concentrations and inspired and end-tidal sevoflurane concentrations during different stages of induction are shown in Table 4. Both BIS and AEP showed a trend of diminishing level of consciousness with both anesthetic techniques.

3.3.2. Correlation Analysis. On correlation analysis of BIS and AEP vs. propofol and sevoflurane concentration, only 6 correlation coefficients in the propofol group were significant at the 0.05 level of significance. $r = -0.50$, -0.49, and -0.45 when BIS is in the unconsciousness stage vs. predicted blood concentration of propofol in sedation, anesthesia stages, and effect-site concentration of propofol in the anesthesia stage, respectively. $r = 0.56$ when AEP is in the anesthesia stage vs. predicted blood concentration and effect-site concentration of propofol in the unconsciousness stages, and $r = -0.53$ when AEP of baseline vs. effect-site concentration of propofol is in the sedation stage. They were around 0.5, just fair correlated. All the others are not significantly correlated (Tables 5 and 6). On correlation analysis

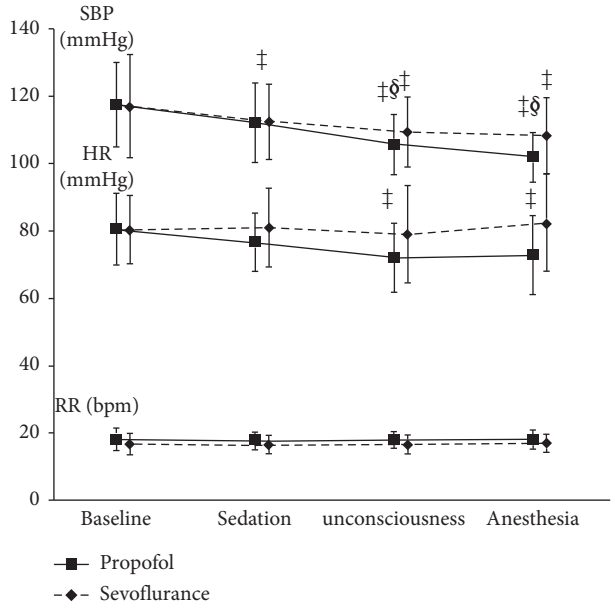


Figure 1: Systolic blood pressure (SBP), heart rate (HR), and respiratory data (RR) of two groups in four stages: base, sedation, unconsciousness, and anesthesia. On SBP, time effect was significantly different at the 0.05 level of significance; time and group interaction effect was significantly different at the 0.01 level of significance. On HR, both time effect and group interaction effects were significantly different at the 0.05 level of significance. On RR, no difference was found in both time effect and interaction effect. ‡ indicates significant difference from baseline. § indicates significant difference from sedation. † indicates significant difference from unconsciousness. On RR, no difference was found in both time effect and interaction effect.

of BIS vs. AEP, only two correlation coefficients in the sevoflurane group were significant at the 0.05 level of significance. $R = 0.66$ when BIS in baseline vs. AEP in the baseline stage, and $r = 0.52$ when BIS in the unconsciousness stage vs. AEP in the anesthesia stage. All the others are not significantly correlated (Tables 7 and 8).

Both monitors showed a trend of diminishing level of consciousness with both anesthetic techniques. The index showed good correlation with stage of induction (Table 9), except with AEP when used with sevoflurane which gave a low correlation coefficient of −0.61 and a 95% CI crossing −0.5. Results showed that propofol had higher correlations between stage of anesthesia and BIS and AEP than sevoflurane. BIS had higher correlations with depth of anesthesia than AEP (given the prior probabilities for the sedation/unconsciousness/anesthesia to be 0.5).

Effective concentrations EC5, EC50, and EC95 referred to drug concentration at which 5%, 50%, and 95% of the patients, respectively, reached the predefined end point. EC5, EC50, and EC95 of predicted blood and effect-site propofol and inspired and end-tidal sevoflurane as well as BIS and AEP values at sedation, unconsciousness, and anesthesia and their sensitivity, specificity, and P_K values are shown in Tables 10–12, respectively. Effect-site propofol concentration had a smaller 95% CI of EC50 than that of blood at all stages from sedation to anesthesia. This difference is not noticed between inspired and end-tidal sevoflurane. BIS gave similar 95% CI of EC50 with propofol and sevoflurane at sedation and unconsciousness, but a range of smaller values with sevoflurane at anesthesia. For AEP, propofol always showed a range of smaller values from sedation to anesthesia.

P_K is the probability that the indicator values of the data points predict correctly which of the data points are the lighter (or deeper). A value of $P_K = 0.5$ means that the indicator correctly predicts the anesthetic depths only 50% of the time, i.e., no better than a 50:50 chance. A value of $P_K = 1$ means that the indicator predicts the anesthetic depths correctly 100% of the time.

3.3.3. Predicting Power. Table 13 shows the P_K for depth of anesthesia measured by the two monitors, BIS and AEP with different anesthetic techniques, propofol and sevoflurane. BIS had good P_K values with both propofol and sevoflurane with both above 0.8, and the P_K with propofol was better than that with sevoflurane. On the contrary, AEP did not have good P_K values, especially with sevoflurane. The P_K values with BIS are significantly better than those with sevoflurane. The prediction powers of BIS and AEP do not seem as good as some papers mentioned ($P_K > 0.9$).

4. Discussion

In this study, we investigated the usefulness and consistency of two anesthetic depth monitors, BIS and AEP with different anesthetic techniques, propofol intravenous anesthesia and sevoflurane inhalational anesthesia. BIS and AEP are two popular anesthetic depth monitors. It is important for them to perform consistently with different anesthetic techniques.

EC50 is analogous to the concept of minimum alveolar concentration for volatile anesthetics and is defined as the concentration of an i.v. anesthetic agent at which 50% of the

Table 4: Descriptive statistics of drug concentrations, BIS, and AEP at different stages of induction.

Mean ± SD, median (range)	Predicted blood concentration $\mu g \cdot ml^{-1}$		Effect-site concentration $\mu g \cdot ml^{-1}$		BIS		AEP	
Stage	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane
Baseline	—	—	—	—	96.5 ± 3.19 97.5 [84–98]	96.6 ± 2.01 97 [90–98]	83.0 ± 10.6 82 [58–99]	79.4 ± 12.64 78.5 [48–99]
Sedation	2.0 ± 0.44 2 [1.5–3.0]	2.7 ± 0.72 2.6 [1.2–3.8]	1.2 ± 0.29 1.2 [0.7–1.7]	1.6 ± 0.53 1.55 [0.6–2.45]	81.2 ± 6.36 82 [64–93]	83.8 ± 14.63 86.5 [41–98]	58.1 ± 23.10 60 [20–95]	63.1 ± 19.45 65.5 [25–99]
Unconsciousness	3.5 ± 0.83 3.5 [2.0–5.5]	3.4 ± 0.62 3.4 [2.4–4.6]	2.4 ± 0.62 2.4 [1.2–3.7]	2.3 ± 0.68 2.25 [1.3–3.95]	58.0 ± 10.33 56.5 [40–74]	59.2 ± 19.18 60 [19–86]	34.1 ± 14.45 30 [10–55]	51.7 ± 20.02 46 [22–89]
Anesthesia	4.2 ± 0.98 4 [2.5–6.0]	4.1 ± 0.87 3.95 [2.6–5.95]	3.2 ± 0.81 3.25 [1.7–4.9]	3.0 ± 0.79 3.1 [1.7–4.25]	47.8 ± 8.67 46.5 [36–66]	41.05 ± 16.05 40 [14–72]	17.78 ± 5.97 17.5 [8–29]	38.7 ± 22.97 30.5 [12–83]

Table 5: BIS and AEP vs. propofol.

Propofol		Correlation coefficient, r							
		BIS				AEPi			
		Base	Sedation	Unconc	Anes	Base	Sedation	Unconc	Anes
Pred	Sedat	0.21	−0.06	−0.50**	−0.02	−0.22	0.08	0.24	0.34
	Unconc	0.40	0.27	−0.34	−0.12	−0.17	0.19	0.29	0.56**
	Anesthesia	0.44	0.30	−0.49**	−0.35	−0.14	0.15	0.33	0.43
Eff	Sedat	0.22	0.09	−0.30	−0.31	−0.53**	0.08	0.10	0.35
	Unconc	0.42	0.27	−0.26	−0.16	−0.23	0.09	0.22	0.56**
	Anesthesia	0.43	0.33	−0.45**	−0.35	−0.25	0.05	0.20	0.39

**Significantly different from $r = 0$ at the 0.05 level. $r = 1$ or -1 means perfect correlation. $r = 0$ means no correlation.

Table 6: BIS and AEP vs. sevoflurane.

Sevoflurane		Correlation coefficient, r							
		BIS				AEPi			
		Base	Sedation	Unconc	Anes	Base	Sedation	Unconc	Anes
Pred	Sedat	−0.11	−0.19	−0.14	−0.17	0.01	−0.02	0.24	0.26
	Unconc	0.04	0.15	−0.003	−0.07	−0.03	−0.10	0.05	0.04
	Anes	0.07	0.05	−0.10	−0.25	−0.12	−0.11	−0.01	−0.03
Eff	Sedat	−0.21	−0.25	−0.16	−0.33	−0.24	−0.23	0.10	0.08
	Unconc	−0.19	−0.03	−0.13	−0.09	−0.22	−0.22	−0.05	−0.10
	Anes	−0.05	0.04	−0.01	−0.18	−0.09	−0.01	0.09	−0.00

Table 7: Correlation analysis of BIS vs. AEPi in the propofol group.

Propofol		Correlation coefficient, r			
		BIS			
		Base	Sedation	Unconc	Anes
AEPi	Base	−0.10	0.04	−0.10	−0.001
	Sedation	0.22	0.40	0.32	−0.01
	Unconc	0.23	0.25	0.37	0.27
	Anesthesia	0.17	0.14	0.30	0.29

Table 8: Correlation analysis of BIS vs. AEPi in the sevoflurane group.

Sevoflurane		Correlation coefficient, r BIS			
		Base	Sedation	Unconc	Anes
AEPi	Base	0.66**	0.27	0.18	0.14
	Sedation	0.29	0.23	0.17	0.18
	Unconc	0.33	0.26	0.38	0.17
	Anesthesia	0.16	0.12	0.52**	0.21

**Significant different from $r = 0$ at the 0.05 level. $r = 1$ or -1 means perfect correlation. $r = 0$ means no correlation.

Table 9: Correlation of depth of anesthesia (4 stages, including baseline) vs. BIS and AEP.

	Spearman correlation coefficient, r [95% CI]		P value (H_0: $s_{sevo} = s_{prop}$)
	Propofol	Sevoflurane	
BIS	−0.92** [−0.87, −0.95]	−0.86** [−0.79, −0.91]	<0.05**
AEP	−0.80** [−0.71, −0.87]	−0.61** [−0.45, −0.73]	<0.05**
P value (H_0: $s_{BIS} = s_{AEP}$)	<0.05**	<0.05**	

Standard errors (SE) for the above s are 0.11. **Significantly different from $s = 0$ at 0.05.

patients will not move or respond to skin incision. This clinically useful concept allows prediction of propofol concentration in the blood and at the effect site [25, 26].

We defined the anesthesia stage as when the patient showed no gross purposeful movement to tetanic stimulation of the ulnar nerve, which was easy to perform and had the advantage over skin incision as a repeatable stimulus. A study showed no significant difference between the effective concentration of propofol which prevented half of the patients to move (EC_{50}) at tetanic stimulation and that at skin incision in somatic response, but significant differences in haemodynamic response [25, 26]. Tetanic stimulation was useful in this study as a reproducible and repeatable stimulus at different propofol and sevoflurane concentrations. Similar to the results from Milne's group, the range of effect-site concentrations to include 90% of patients (EC5–EC95) was smaller than the predicted blood concentration range and hence a more useful figure to guide propofol administration [27]. Similarly, in the sevoflurane group, the range of end-tidal concentrations was smaller than the inspired, but to a lesser extent. Both monitors had distinctly different EC50s with small 95% CI. BIS had similar EC50s with both propofol and sevoflurane, but AEP showed different values between the two anesthetic techniques.

In this study, 90% of the patients were sedated at a BIS value between 90 and 71 with propofol or between 100 and 60 with sevoflurane. This indicates that BIS is therefore better at predicting sedation with propofol. AEP showed very wide range of values in order to induce 90% of the patients at sedation with both propofol (AEP value range 100-11) and sevoflurane (AEP value range 96-27), and therefore, AEP did not seem to be useful in guiding sedation. At unconsciousness, BIS showed a smaller range with propofol (BIS value range 77-37) than with sevoflurane (BIS value range 93-23), which might again indicate that BIS performs better with propofol. AEP showed a wide range with both propofol (AEP value range 61-4) and sevoflurane (AEP value range 85-12) at unconsciousness. At anesthesia, BIS again had a smaller range with propofol (BIS value range 61-31) than with sevoflurane (BIS value range 67-11). AEP showed a narrow range with propofol (AEP value range 28-6) but a wide one with sevoflurane (AEP value range 75-0). BIS appeared to be a good indicator of depth of anesthesia with propofol, which was reflected by the high P_K value of 0.91. Anesthetic seemed to have an effect on performance of the monitors, particularly with AEP monitor. BIS overall performed well with both anesthetic techniques, i.v. propofol and inhalational sevoflurane, but with a higher P_K with propofol. AEP showed poorer performance than BIS in our study. With a P_K of 0.56 with sevoflurane, AEP became doubtful as an anesthetic depth monitor which means the prediction powers of BIS and AEP do not seem as good as some papers mentioned [21, 22, 28, 29]. Considering the difference results between this study and previous ones, different protocols of studies might be the reason [22, 28–30]. We use detected more drug concentrations at more time points with more accurate statistical methods, but we still think we need more studies to verify the results. And this result might remind the clinicians that both BIS and AEP are not as reliable as they thought.

In summary, we obtained good EC50 with both monitors, but the index variation was affected by the anesthetic technique. The performance of the anesthetic depth monitors was better when propofol was used. Very wide variation was found in the combination of AEP and sevoflurane [22, 25, 31–33]. It seems the monitors are at best at giving the EC50, at which there is a 50% chance that the patient has reached the end point, and we have not yet found an anesthetic depth monitor that has high sensitivity and specificity and not affected by the anesthetic technique.

Table 10: Propofol/sevoflurane concentration, BIS/AEP values, and prediction probabilities at sedation (mean (95% CI).

	EC5		EC50		EC95		Sensitivity (%)		Specificity (%)		Prediction probability (P_K)	
	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane
Blood concentration (g/ml)/Fi (%)	0.9	1.2	1.7 (1.5–1.9)	2.5 (2.4–2.7)	2.6	3.9	92	82	64	72	0.78	0.75
Effect-site concentration (g/ml)/ET (%)	0.6	0.6	1.1 (1.1–1.2)	1.6 (1.5–1.7)	1.6	2.5	73	76	77	75	0.77	0.76
BIS	90	100	81 (80–82)	85 (82–87)	71	60	80	63	77	84	0.78	0.73
AEP	100	96	56 (52–60)	62 (58–65)	11	27	76	82	75	69	0.76	0.76

Table 11: Propofol/sevoflurane concentration, BIS/AEP values, and prediction probabilities at unconsciousness (mean (95% CI).

	EC5		EC50		EC95		Sensitivity (%)		Specificity (%)		Prediction probability (P_K)	
	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane
Blood concentration (g/ml)/Fi (%)	1.9	2.3	3.2 (3–3.4)	3.3 (3.2–3.4)	4.5	4.3	85	79	81	66	0.83	0.72
Effect-site concentration (g/ml)/ET (%)	1.3	1.1	2.3 (2.2–2.4)	2.2 (2.1–2.3)	3.3	3.3	83	79	73	75	0.78	0.77
BIS	77	93	57 (55–59)	58 (54–61)	37	23	79	77	72	77	0.76	0.77
AEP	61	85	32 (30–35)	48 (45–52)	4	12	71	68	79	82	0.75	0.75

Table 12: Propofol/sevoflurane concentration, BIS/AEP values, and prediction probabilities at anesthesia (mean (95% CI).

	EC5		EC50		EC95		Sensitivity (%)		Specificity (%)		Prediction probability (P_K)	
	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane	Propofol	Sevoflurane
Blood concentration (g/ml)/Fi (%)	2.3	2.4	4 (3.7–4.2)	3.9 (3.8–4.1)	5.6	5.4	89	79	73	73	0.81	0.76
Effect-site concentration (g/ml)/ET (%)	1.7	1.6	3.1 (3–3.1)	3 (2.8–3.1)	4.5	4.4	76	78	76	75	0.76	0.76
BIS	61	67	46 (45–48)	39 (37–42)	31	11	73	76	76	75	0.74	0.76
AEP	28	75	17 (16–18)	34 (30–37)	6	0	81	66	72	83	0.77	0.75

Table 13: Prediction probability P_K for the depth of anesthesia (considering 4 stages including baseline).

	P_K (SE) [P values for testing $P_K = 0.5$]		P value [H_0: $P_{KBIS} = P_{KAEP}$]
	BIS	AEP	
Propofol	0.9117 (0.0147) [$P < 0.0001^{**}$]	0.7504 (0.0238) [$P < 0.0001^{**}$]	$<0.0001^{**}$
Sevoflurane	0.8233 (0.0182) [$P < 0.0001^{**}$]	0.5629 (0.0391) [$P < 0.0001^{**}$]	$<0.0001^{**}$

**Significantly different from $s = 0$ at 0.05.

Data Availability

All the underlying data supporting the results of this study can be found in IRB of Second Affiliated Hospital of Dalian Medical University.

Disclosure

Haitao Yang and Guan Wang are the co-first authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Haitao Yang and Guan Wang contributed equally to this work.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. H81471373).

References

[1] S. R. Tasbihgou, M. F. Vogels, and A. R. Absalom, "Accidental awareness during general anaesthesia-a narrative review," *Anaesthesia*, vol. 73, no. 1, pp. 112–122, 2018.

[2] M. Graham, A. M. Owen, K. Çipi, C. Weijer, and L. Naci, "Minimizing the harm of accidental awareness under general anesthesia," *Anesthesia & Analgesia*, vol. 126, no. 3, pp. 1073–1076, 2018.

[3] J. W. Sleigh, K. Leslie, A. J. Davidson et al., "Genetic analysis of patients who experienced awareness with recall while under general anesthesia," *Anesthesiology*, vol. 131, no. 5, pp. 974–982, 2019.

[4] H. Yu and D. Wu, "Effects of different methods of general anesthesia on intraoperative awareness in surgical patients," *Medicine*, vol. 96, no. 42, p. e6428, 2017.

[5] S. Devroe, M. Van De Velde, and S. Rex, "General anesthesia for caesarean section," *Current Opinion in Anaesthesiology*, vol. 28, no. 3, pp. 240–246, 2015.

[6] P. Bischoff, I. Rundshagen, and G. Schneider, "Unerwünschte wachphänomene ("awareness") während allgemeinanästhesie," *Der Anaesthesist*, vol. 64, no. 10, pp. 732–739, 2015.

[7] M. A. Earley, L. T. Pham, and M. M. April, "Scoping review: awareness of neurotoxicity from anesthesia in children in otolaryngology literature," *The Laryngoscope*, vol. 127, no. 8, pp. 1930–1937, 2017.

[8] M. Cascella, R. Fusco, D. Caliendo et al., "Anesthetic dreaming, anesthesia awareness and patient satisfaction after deep sedation with propofol target controlled infusion: a prospective cohort study of patients undergoing day case breast surgery," *Oncotarget*, vol. 8, no. 45, pp. 79248–79256, 2017.

[9] G. Ratnayake and V. Patil, "General anaesthesia during caesarean sections," *Current Opinion in Obstetrics and Gynecology*, vol. 31, no. 6, pp. 393–402, 2019.

[10] S. D. Bergese, A. A. Uribe, E. G. Puente et al., "A prospective, multicenter, single-blind study assessing indices of SNAP II versus BIS VISTA on surgical patients undergoing general anesthesia," *JMIR Research Protocols*, vol. 6, no. 2, p. e15, 2017.

[11] C. M. Schulz, V. Krautheim, A. Hackemann, M. Kreuzer, E. F. Kochs, and K. J. Wagner, "Situation awareness errors in anesthesia and critical care in 200 cases of a critical incident reporting system," *BMC Anesthesiology*, vol. 16, no. 4, 2016.

[12] M. H. Chiang, S. C. Wu, S. W. Hsu, and J. C Chin, "Bispectral Index and non-bispectral Index anesthetic protocols on postoperative recovery outcomes," *Minerva Anestesiologica*, vol. 84, no. 2, pp. 216–228, 2018.

[13] J. Lee, C. Park, and S. Kim, "Awareness during general anesthesia despite simultaneous bispectral index and end-tidal anesthetic gas concentration monitoring," *Yeungnam University Journal of Medicine*, vol. 36, no. 1, pp. 50–53, 2019.

[14] B. Altiparmak, N. Celebi, O. Canbay, M. K. Toker, B. Kılıçarslan, and Ü. Aypar, "Effect of magnesium sulfate on anesthesia depth, awareness incidence, and postoperative pain scores in obstetric patients. a double-blind randomized controlled trial," *Saudi Medical Journal*, vol. 39, no. 6, pp. 579–585, 2018.

[15] S. R. Lewis, M. W. Pritchard, L. J. Fawcett, and Y. Punjasawadwong, "Bispectral index for improving intraoperative awareness and early postoperative recovery in adults," *Cochrane Database of Systematic Reviews*, vol. 9, no. 9, Article ID 9CD003843, 2019.

[16] T. Lahtinen, J. Seppälä, T. Viren, and K. Johansson, "Experimental and analytical comparisons of tissue dielectric constant (TDC) and bioimpedance spectroscopy (BIS) in assessment of early arm lymphedema in breast cancer patients after axillary surgery and radiotherapy," *Lymphatic Research and Biology*, vol. 13, no. 3, pp. 176–185, 2015.

[17] I. Karaca, F. Eren Akcil, O. Korkmaz Dilmen, G. Meyanci Koksal, and Y. Tunalı, "The effect of BIS usage on anaesthetic agent consumption, haemodynamics and recovery time in supratentorial mass surgery," *Turkish Journal of Anesthesia and Reanimation*, vol. 42, no. 3, pp. 117–122, 2014.

[18] J. Shepherd, J. Jones, G. Frampton, J. Bryant, L. Baxter, and K. Cooper, "Clinical effectiveness and cost-effectiveness of depth of anaesthesia monitoring (E-entropy, bispectral index and narcotrend): a systematic review and economic evaluation," *Health Technology Assessment*, vol. 17, no. 34, pp. 1–264, 2013.

[19] M. Cornella, A. Bendixen, S. Grimm et al., "Spatial auditory regularity encoding and prediction: human middle-latency and long-latency auditory evoked potentials," *Brain Research*, vol. 1626, pp. 162621–162630, 2015.

[20] L. Li and Q. Gong, "The early component of middle latency auditory-evoked potentials in the process of deviance detection," *Neuroreport*, vol. 27, no. 10, pp. 769–773, 2016.

[21] Y. Punjasawadwong, W. Chau-In, M. Laopaiboon, S. Punjasawadwong, and P. Pin-On, "Processed electroencephalogram and evoked potential techniques for amelioration of postoperative delirium and cognitive dysfunction

following non-cardiac and non-neurosurgical procedures in adults," *Cochrane Database of Systematics Review*, vol. 5, no. 5, Article ID 5CD011283, 2018.

[22] K. Szostakiewicz, Z. Rybicki, and D. Tomaszewski, "Non-instrumental clinical monitoring does not guarantee an adequate course of general anesthesia. a prospective clinical study," *Biomedical Papers*, vol. 162, no. 3, pp. 198–205, 2018.

[23] M. Tacke, E. F. Kochs, M. Mueller et al., "Machine learning for a combined electroencephalographic anesthesia index to detect awareness under anesthesia," *PLoS One*, vol. 15, no. 8, Article ID e0238249, 2020.

[24] B. Huang, F. Liang, L. Zhong et al., "Latency of auditory evoked potential monitoring the effects of general anesthetics on nerve fibers and synapses," *Scientific Reports*, vol. 5, no. 1, Article ID 12730, 2015.

[25] S. Li, F. Yu, H. Zhu, Y. Yang, L. Yang, and J. Lian, "The median effective concentration (EC50) of propofol with different doses of fentanyl during colonoscopy in elderly patients," *BMC Anesthesiology*, vol. 16, p. 24, 2016.

[26] Y.-C. Shang and B.-Z. Chen, "Propofol EC50: an effect of luteal phase core temperature differences?," *British Journal of Anaesthesia*, vol. 114, no. 3, p. 526, 2015.

[27] S. E. Milne, A. Troy, M. G. Irwin, and G. N. C. Kenny, "Relationship between bispectral index, auditory evoked potential index and effect-site EC50 for propofol at two clinical end-points †," *British Journal of Anaesthesia*, vol. 90, no. 2, pp. 127–131, 2003.

[28] B. Horn, S. Pilge, E. F. Kochs, G. Stockmanns, A. Hock, and G. Schneider, "A combination of electroencephalogram and auditory evoked potentials separates different levels of anesthesia in volunteers," *Anesthesia & Analgesia*, vol. 108, no. 5, pp. 1512–1521, 2009.

[29] C. Luo and W. Zou, "Cerebral monitoring of anaesthesia on reducing cognitive dysfunction and postoperative delirium: a systematic review," *Journal of International Medical Research*, vol. 46, no. 10, pp. 4100–4110, 2018.

[30] C. Jeleazcov, G. Schneider, M. Daunderer, B. Scheller, J. r. Sch??ttler, and H. Schwilden, "The discriminant power of simultaneous monitoring of spontaneous electroencephalogram and evoked potentials as a predictor of different clinical states of general anesthesia," *Anesthesia & Analgesia*, vol. 103, no. 4, pp. 894–901, 2006.

[31] M. Zaballos, E. Bastida, S. Agusti, M. Portas, C. Jiménez, and M. López-Gil, "Effect-site concentration of propofol required for LMA-supreme insertion with and without remifentanil: a randomized controlled trial," *BMC Anesthesiology*, vol. 15, p. 131, 2015.

[32] J. Y. Yoo, H. J. Kwak, K. C. Lee, G. W. Kim, and J. Y. Kim, "Predicted EC50 and EC95 of remifentanil for smooth removal of a laryngeal mask airway under propofol anesthesia," *Yonsei Medical Journal*, vol. 56, no. 4, pp. 1128–1133, 2015.

[33] L. Laaksonen, M. Kallioinen, J. Långsjö et al., "Comparative effects of dexmedetomidine, propofol, sevoflurane, and S-ketamine on regional cerebral glucose metabolism in humans: a positron emission tomography study," *British Journal of Anaesthesia*, vol. 121, no. 1, pp. 281–290, 2018.