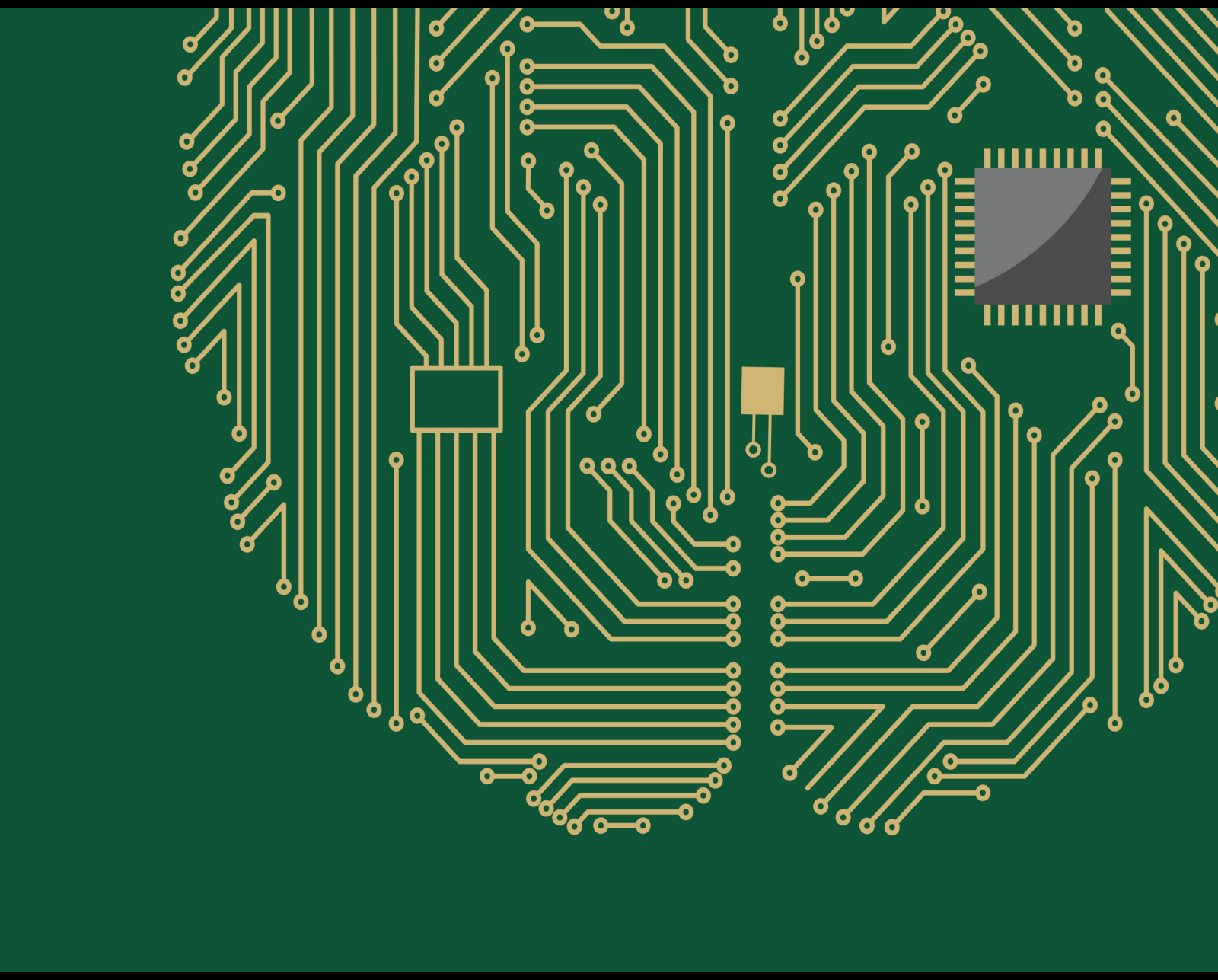
Computational Intelligence and Neuroscience

Explainable and Reliable Machine Learning by Exploiting Large-Scale and Heterogeneous Data

Lead Guest Editor: Nian Zhang
Guest Editors: Qingshan Liu and Zhishan Guo



Explainable and Reliable Machine Learning by Exploiting Large-Scale and Heterogeneous Data

Computational Intelligence and Neuroscience

Explainable and Reliable Machine Learning by Exploiting Large-Scale and Heterogeneous Data

Lead Guest Editor: Nian Zhang
Guest Editors: Qingshan Liu and Zhishan Guo

Contents

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8863847, 8 pages
https://doi.org/10.1155/2020/8863847

Research Article

A Study on Differences between Simplified and Traditional Chinese Based on Complex Network Analysis of the Word Co-Occurrence Networks

Zhongqiang Jiang,[1] Dongmei Zhao,[1] Jiangbin Zheng,[2,3] and Yidong Chen[2,3]

[1]*China Mobile (Suzhou) Software Technology Co., Ltd., Suzhou, China*
[2]*Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, China*
[3]*Xiamen Key Laboratory of Language and Culture Computation, Xiamen University, Xiamen 361005, China*

Correspondence should be addressed to Jiangbin Zheng; jiangbinzheng@stu.xmu.edu.cn and Yidong Chen; ydchen@xmu.edu.cn

Received 22 July 2020; Revised 15 September 2020; Accepted 19 October 2020; Published 3 December 2020

Academic Editor: Nian Zhang

Currently, most work on comparing differences between simplified and traditional Chinese only focuses on the character or lexical level, without taking the global differences into consideration. In order to solve this problem, this paper proposes to use complex network analysis of word co-occurrence networks, which have been successfully applied to the language analysis research and can tackle global characters and explore the differences between simplified and traditional Chinese. Specially, we first constructed a word co-occurrence network for simplified and traditional Chinese using selected news corpora. Then, the complex network analysis methods were performed, including network statistics analysis, kernel lexicon comparison, and motif analysis, to gain a global understanding of these networks. After that, the networks were compared based on the properties obtained. Through comparison, we can obtain three interesting results: first, the co-occurrence networks of simplified Chinese and traditional Chinese are both small-world and scale-free networks. However, given the same corpus size, the co-occurrence networks of traditional Chinese tend to have more nodes, which may be due to a large number of one-to-many character/word mappings from simplified Chinese to traditional Chinese; second, since traditional Chinese retains more ancient Chinese words and uses fewer weak verbs, the traditional Chinese kernel lexicons have more entries than the simplified Chinese kernel lexicons; third, motif analysis shows that there is no difference between the simplified Chinese network and the corresponding traditional Chinese network, which means that simplified and traditional Chinese are semantically consistent.

1. Introduction

Chinese is usually written in two forms: simplified Chinese (mainly used in Mainland China and Singapore) and traditional Chinese (mainly used in Hong Kong, Macao, and Taiwan). Although simplified Chinese is derived from traditional Chinese, the two systems are quite different on various levels, such as character set, encoding method, orthography, vocabulary, and semantics, which create barriers to communication between different areas where Chinese is spoken. This linguistic phenomenon is due to the independent development of these two homologous systems in the past half century, and they will continue to evolve in their respective cultural environments. However, in the past few decades, with the increase in exchange activities between four cross-strait regions, the problem of conversion between simplified Chinese and traditional Chinese as well as the comparison of the differences between simplified Chinese and traditional Chinese has attracted the attention of more and more researchers [1–4]. In short, the comparison between Simplified Chinese and Traditional Chinese has important reference value for the study of language evolution.

So far, research on comparing differences between these two forms of Chinese still focuses on the character or lexical levels [1, 3, 5]. For example, Fei [6] made a systematic comparison of the similarities and differences of the current

Chinese characters in simplified and traditional Chinese characters; Li [7] made an in-depth analysis of the reasons for the differences in the form of simplified and traditional Chinese characters from the aspects of politics, history and culture, and the principles of character selection; Liu [8] conducted a comprehensive analysis mainly from the perspective of eliminating the differences in form; Jiang [9] mainly compared and analyzed simplified and traditional Chinese vocabulary from two aspects: homographs with different meanings and different forms with synonymous meanings; Li and Qiu [10] discussed the causes, types, and processing methods of differences in dictionaries across the Taiwan Strait.

On the other hand, as an important methodology for linguistic research, complex networks-based approaches show their advantage in revealing the global features of language which have been successfully applied to analyse languages at various levels, e.g., lexical [11–13], word co-occurrence [14–18], syntax [19–21], and semantic [22–24]. This is because language is a typical hierarchical system which has a highly complex network structure, and complex network analysis methods have the advantage of revealing the laws of language as a whole. Hence, in this paper, we apply complex network analysis methods to explore the differences between simplified and traditional Chinese character systems from a holistic perspective. Specially, according to the construction method of the word co-occurrence network, this paper proposed to construct simplified Chinese and traditional Chinese word co-occurrence networks with different numbers of nodes and different corpus sizes and then make corresponding research on the complex characteristics of these networks. Through the obtained simplified and traditional Chinese core dictionary, we explored the differences between the two languages. In addition, this paper proposed to use primitives representing language semantics to analyze the semantic differences between simplified and traditional languages.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 puts forward a brief introduction to some basic concepts related to complex network analysis. Then, in Section 4, we constructed networks with different text scales and carried out corresponding studies on the characteristics of complex networks, e.g., cumulative degree distribution, clustering coefficient, kernel lexicon, and motif analysis. Finally, Section 5 concludes the paper.

2. Related Work

At present, the comparison and analysis of the differences between simplified and traditional Chinese mainly remain at the level of character shapes or words. The main reason why readers find it difficult to read unfamiliar written materials in simplified or traditional characters is due to the difference in glyphs. Studies have shown that the actual number of characters that can be compared in the simplified and traditional Chinese character lists is 4,786 [6]. Among them, 41% of the simplified and traditional characters used in mainland China and Taiwan have the same glyph, totaling 1,947 characters; 24% of the similar glyphs, totaling 1,170 characters; and 35% of different glyphs, totaling 1,669 characters. Simplified and traditional Chinese belong to the same ancestor and developed from the same ancient Chinese. Therefore, the differences between simplified and traditional Chinese need to be compared and analyzed systematically and comprehensively from the perspective of the language as a whole, which explores the differences between the two written forms of Chinese development status and law. However, the current comparative work of simplified and traditional Chinese characters has only achieved outstanding achievements on the level of character form and word, while other language levels (such as semantics and syntax) have not been involved.

As a typical hierarchical system, language exhibits a highly complex network structure at all levels (phonetics, morphology, syntax, and semantics) [25]. At present, a lot of research studies have been carried out on the complex characteristics of language networks on different levels, including lexical or vocabulary networks, word or character co-occurrence networks, and syntactic networks, the semantic networks. These research studies are important for identifying and understanding the topological structure of language. Among them, the research studies of Chinese network mainly include the following: in terms of morphology or vocabulary network, Li et al. [13] used Chinese characters as nodes based on the principle that two Chinese characters can form words and constructed a Chinese phrase network and studied the dynamic characteristics of the phrase network; in terms of syntactic network, Liu [20] used the syntactic labeling tree bank to connect the words with syntactic relations and finally established the Chinese syntactic dependency network and explored the complex network characteristics of the syntactic network; in the semantic network (current research studies on Chinese semantic networks are still relatively small), Liu et al. [24] constructed a small semantic network to explore the complex characteristics of the Chinese semantic network; and Cancho and Solé [14] used the English-speaking country corpus to construct an English word co-occurrence network and found that the English language network has a small world and scale-free features. Liu and Sun [15] used the same construction method to construct a simplified Chinese word co-occurrence network. The experiment proved that the simplified Chinese word co-occurrence network has complex network characteristics similar to the English word co-occurrence network. Other works [12, 26, 27] used different construction strategies to construct a Chinese word, word co-occurrence network, and English word co-occurrence network based on different themes of Chinese and English (prose, novels, popular science articles, and news reports) corpora.

3. Foundations

In this section, some basic concepts are put forward. Section 3.1 describes the basic definitions of the complex network. Then, Section 3.2 describes small-world networks and scale-free networks. Finally, Section 3.3 gives a brief introduction of motif analysis.

3.1. Basic Definitions. In general, a network G can be denoted as a two-tuples (V, E), where V is the set of vertices and E is the set of edges. In a language network, a vertex $v_i\,(1 \le i \le |V|)$ may represent a radical, character, or word; and an edge $e_{ij}\,(1 \le i, j \le |V|)$ can characterize the relationship between v_i and v_j.

Given a network, the conventional indicators, such as average path length, clustering coefficient, degree distribution, and cumulative degree distribution, are used to specify its statistical characteristics. These indicators could be defined, respectively, as follows:

Average Path Length ($\overline{d}$): the average distance between two reachable vertices:

$$\overline{d} = \frac{2}{N(N-2)} \sum_{i>j} d_{ij}, \tag{1}$$

where N is the number of vertices in the network, d_{ij} is the distance between vertex v_i and vertex v_j which also means the number of edges in the shortest path linking them.

Clustering Coefficient (C): the percentage of the neighbours that two vertices share. The clustering coefficient of vertices i could be defined as follows [23]:

$$C_i = \frac{2E_i}{k_i(k_i-1)}, \quad k_i \ne 0, 1, \tag{2}$$

where k_i is the degree of vertex i and E_i is the number of edges among the vertices in the nearest neighbourhood of vertex i. Moreover, the clustering coefficient of the whole network is the average of all individual C_i, as follows:

$$C = \frac{1}{N} \sum_{i=1} C_i. \tag{3}$$

3.2. Small-World Networks and Scare-Free Networks. A complex network is called a small-world network, in which the average number of edges lying between any two vertices is very small, while the clustering coefficient remains large. Specifically, for an ER random network in a small-world network, d_{ER} and C_{ER} represent the average shortest path and clustering coefficient, respectively, and d is similar to d_{ER}, but $C \gg C_{ER}$ [28].

The degree distribution reveals the distribution of vertices by degree:

$$P(k) = \sum_{k'=k}^{\infty} P(k'), \tag{4}$$

and the percentage of the vertices whose degrees are k is represented as $P(k)$:

$$P(k) = \sum_{k'=k}^{\infty} k'^{-\gamma} \propto k^{-(\gamma-1)}. \tag{5}$$

Under certain circumstances, a network is called scale-free if it fits the power law well and lies between 2 and 3 [29].

3.3. Motif Analysis. Motif, a subgraph constructed by a few edges and vertices, was first used in biological academic area [30]. For a complex network, a motif represents a subnetwork containing a small number of nodes and edges. Biemann et al. [31] first applied motif analysis in linguistic networks and semantic features to explore the difference between natural language text and text generated by an N-gram language model in terms of semantic characteristics.

Besides, motif analysis involves an intermediate level of a network, which specifically means to count the motif constructed by n nodes to approach comparison among networks. As to undirected co-occurrence networks, n is usually at least 3. A 3-node motif is a triple-contained completely in calculating the clustering coefficient. Therefore, we use 4-node motif analysis to compare the semantic differences of co-occurrence networks. All six kinds of undirected 4-node motifs are shown in Figure 1.

4. Experimental Comparisons

This section addresses the experimental comparisons between simplified and traditional Chinese based on methods from complex network science. Section 4.1 describes the dataset used as well as the construction of the word co-occurrence networks. Then, Sections 4.2–4.4 describe the comparisons on small-world and scale-free, kernel lexicons, and motif analysis, respectively.

4.1. Dataset and Network Construction. In this experiment, texts from *Chinese GigaWord Third Edition (LDC2007T38)*(https://catalog.ldc.upenn.edu/LDC2007T38) are used as the experimental materials, of which the simplified Chinese texts are from "*Xinhua News Agency*" (hereinafter referred to as XIN) and the traditional Chinese texts are from "*Central News Agency*" (hereinafter referred to as CNA).

Based on the datasets, word co-occurrence networks are built according to the method proposed by [32]. Concretely, words in the texts are regarded as nodes in the networks, and any two nodes are connected if the distance of the corresponding words is not greater than 2.

After the networks are constructed, their statistical properties are observed and compared. Please note that, only the networks built from the similar text scales are compared which avoids the influence of the text scales. In this experiment, three text scales are used, and the statistics of all the networks are shown in Table 1. For the co-occurrence network of simplified and traditional Chinese words under the same corpus scale, we designed three sets of experiments. The scales of the corpus used in these three sets increased from initial 7 million words to 10 million words and then 15 million words.

4.2. Small-World and Scare-Free. Given the built networks, we use a complex network analysis tool, *Pajek*[2] to calculate the statistical properties of the networks. Table 2 shows the results.

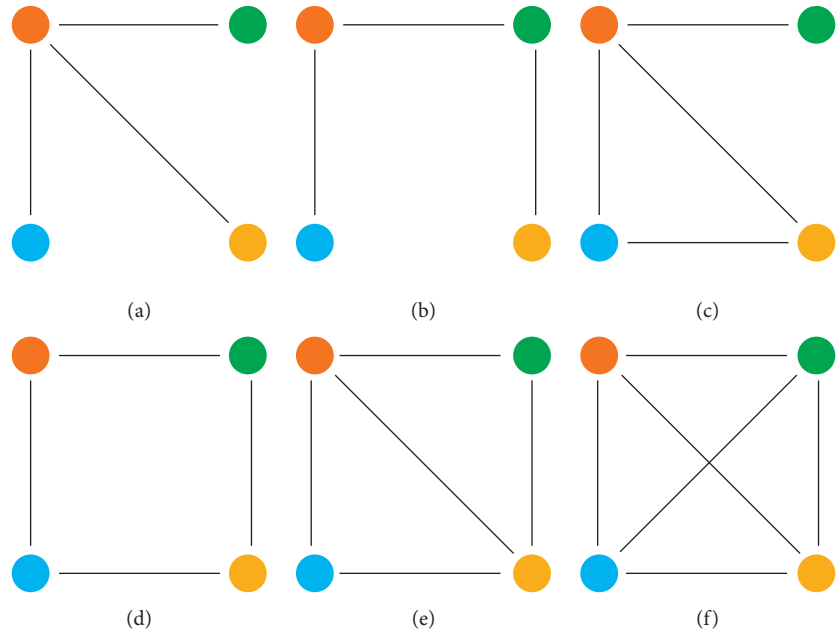


Figure 1: All undirected motifs of size 4. (a) Star; (b) chain; (c) 3-loop-out; (d) box; (e) semiclique; (f) Clique.

Table 1: Statistics of the built word co-occurrence networks. XIN_1, XIN_2, and XIN_3 are from different parts of the XIN dataset; CNA_1, CNA_2, and CNA_3 are from different parts of the CNA dataset.

	Theme (name)	Text scales (# of words) (M)	Sources	# of nodes
Group 1	XIN_1	55.9	XIN (Jan., 2006–May., 2006)	$1.06 * 10^5$
	CNA_1	55.3	CNA (Jan., 2006–Mar., 2006)	$1.14 * 10^5$
Group 2	XIN_2	79.8	XIN (Jan., 2006–Jun., 2006)	$1.26 * 10^5$
	CNA_2	79	CNA (Jan., 2006–Apr., 2006)	$1.38 * 10^5$
Group 3	XIN_3	115	XIN (Jan., 2006–Sep., 2006)	$1.52 * 10^5$
	CNA_3	114	CNA (Jan., 2006–May., 2006)	$1.69 * 10^5$

Table 2: Properties of the built networks. N: number of nodes; E: number of edges; $\overline{k}$: average degree of nodes; C: clustering coefficient; $\overline{d}$: average path length among reachable pairs of nodes; C_{ER}: clustering coefficient of an ER network with same numbers of nodes and edges; d_{ER}: average path length among reachable pairs of nodes in an ER network with same numbers of nodes and edges; and γ: power-law exponent in equation (5).

Metric	Dataset theme					
	XIN_1	CNA_1	XIN_2	CNA_2	XIN_3	CNA_3
N	$1.06 * 10^5$	$1.14 * 10^5$	$1.26 * 10^5$	$1.38 * 10^5$	$1.52 * 10^5$	$1.69 * 10^5$
E	$0.27 * 10^7$	$0.32 * 10^7$	$0.35 * 10^7$	$0.41 * 10^7$	$0.45 * 10^7$	$0.53 * 10^7$
$\overline{k}$	50.01	55.08	54.45	59.39	58.45	62.86
C	0.68	0.68	0.69	0.70	0.72	0.73
$\overline{d}$	2.69	2.72	2.69	2.73	2.70	2.74
C_{ER}	$4.69 * 10^{-4}$	$4.80 * 10^{-4}$	$4.28 * 10^{-4}$	$4.30 * 10^{-4}$	$3.90 * 10^{-4}$	$3.70 * 10^{-4}$
d_{ER}	3.24	3.21	3.26	3.20	3.25	3.20
γ	2.17	2.18	2.16	2.17	2.15	2.15

From Table 2, we can find that all the networks satisfy $\overline{d} \approx d_{ER}$ and $C \gg C_{ER}$, which means that all the networks are small-world networks. However, it could also be observed that the average degrees of traditional networks are about 5 points larger than those of the corresponding simplified networks. The possible reason is the many-to-one mappings between traditional Chinese and simplified Chinese, i.e., different words in |traditional Chinese have the same forms. For example, two traditional Chinese words "編制 (biān zhì)" and "編製 (biān zhì)" have that same form "编制 (biān zhì)" in simplified Chinese. It is the many-to-one mappings between traditional Chinese and simplified Chinese lead to larger numbers of nodes, edges, and average degrees.

Moreover, we plot the cumulative degree distributions of all the networks, as well as their fitting curves in Figure 2. It is clear that both traditional and simplified Chinese networks fit the power law well. In addition, the power-law exponents of all the networks belong to the range of 2 and 3, indicating that all of the networks are scale-free.

4.3. Kernel Lexicons. By observing the cumulative degree distribution curves in Figure 2, we can learn that the scattered points can be fitted by two lines with different slopes. And the whole data set is divided into two parts at the crossover point. The more frequently a word is used in daily life, the more semantic meanings it may contain [33]. And the frequency f of a given word is relevant to its degree k, as follows:

$$k \propto f^{\alpha}, \quad \alpha > 0. \tag{6}$$

Followed [15], we may obtain a kernel dictionary by sorting words according to their degrees and selecting those with more degrees. Concretely, the capacity of kernel lexicons is calculated as follows:

$$N_{KL} = N \times P(k_{\text{cross}}), \tag{7}$$

where N denotes the number of nodes, or specifically the number of words, and k_{cross} denotes the percentage of the words whose degrees are not less than k_{cross}, which is the number at the crossover point.

Table 3 shows the sizes of the constructed kernel lexicons. From Table 3, we can learn that the sizes are all about 10^3 levels and satisfy the claim proposed by [15, 34]. However, we observed the number of traditional Chinese kernel lexicons is much greater than that of simplified Chinese. Concretely, the traditional Chinese kernel lexicons are about 900 words, which are more than simplified Chinese in average.

To find out the possible reasons, we further analysis the part-of-speech tags and the lengths for the words in the kernel lexicons. The results are listed in Tables 4 and 5, respectively.

From Table 4, we found that, both forms of Chinese have a large proportion on entity words (noun and verb) whose orders are roughly the same. The percentage of verb in traditional Chinese is generally greater than that in simplified Chinese, indicating that verb weakening is an important development process in simplified Chinese.

From Table 5, we learned that kernel lexicons extracted from the traditional Chinese corpora contain more 1-character words than the ones extracted from the simplified Chinese corpora. This implies that traditional Chinese maintains some features of classical Chinese, while simplified Chinese does not.

4.4. Motif Verification. Followed [31], we performed the motif analysis upon each networks constructed in Section 4.1. The results are shown in Table 6. There is no difference between simplified Chinese networks and the corresponding traditional Chinese networks, except that the traditional Chinese complex networks tend to have more motifs than the simplified Chinese ones which is due to the larger number of nodes and edges of the traditional Chinese networks. This shows that simplified and traditional Chinese are consistent on the semantics level.

4.5. Example Comparison. We found that parts of speech of these different words are mainly reflected in nouns, verbs, time words, gerunds, adverbs, numerals, and ground nouns, as shown in Table 4. Among them, nouns, verbs, gerunds, and adverbs vary with corpus. However, there are also some words that are unique or frequently used in specific areas due to regional and political reasons, such as "总统", "中华民国'", "卫生署", "社会主义", and "农民工"; time words, numerals, and geographical nouns also have different usage habits or frequency of use due to different regional cultures, such as "二零零五年", "2005年", "二十五", "25", "高雄县", and "长江".

Furthermore, we found that nearly 25% of the different words in traditional Chinese are single-character words, such as "逾/vg", "採/v", "恆/ag", and "常/d". The number of single-character words in different words in simplified Chinese is relatively small. These single-character words frequently appear in the traditional corpus. Some words are function words or substantive words with grammatical effect, and some words are produced by the word segmentation tool incorrectly. But most single-character words appear in sentences mainly in the form of classical Chinese, "黃金/n 博物/n 園區/n 為/v 將/p 此/rz 深/d 具/vg 教育/vn 意義/n 的/uj 活動/vn 推廣/v 至/p 瑞芳/ns 在地/b 的/uj 學校/n 與/c 社區/n 團體/n " and "他/rr 一度/d 懷疑/v 自己/rr 能否/v 常/d 保/v 早先/t 的/uj 成就/n". This shows that many ancient Chinese words still appear in the written language of the traditional Chinese character system with a higher frequency, i.e., the written language of the traditional Chinese character system retains more classical Chinese characteristics.

In summary, the core dictionaries of the simplified and traditional Chinese character systems have a certain degree of versatility. However, in the process of language development, there have been some differences due to regional usage habits, environment, politics, and the generation of new words. In addition, in the development of the traditional Chinese character system, its written language still retains certain characteristics of classical Chinese.

5. Conclusion

In this paper, we proposed complex network to explore differences between simplified Chinese and traditional Chinese. To the best of our knowledge, this is the first work to use complex network-based approaches in comparing differences between simplified and traditional Chinese. Through the comparisons, we achieve 3 interesting results. Firstly, both co-occurrence networks for simplified and for traditional Chinese are small-world and scale-free networks.

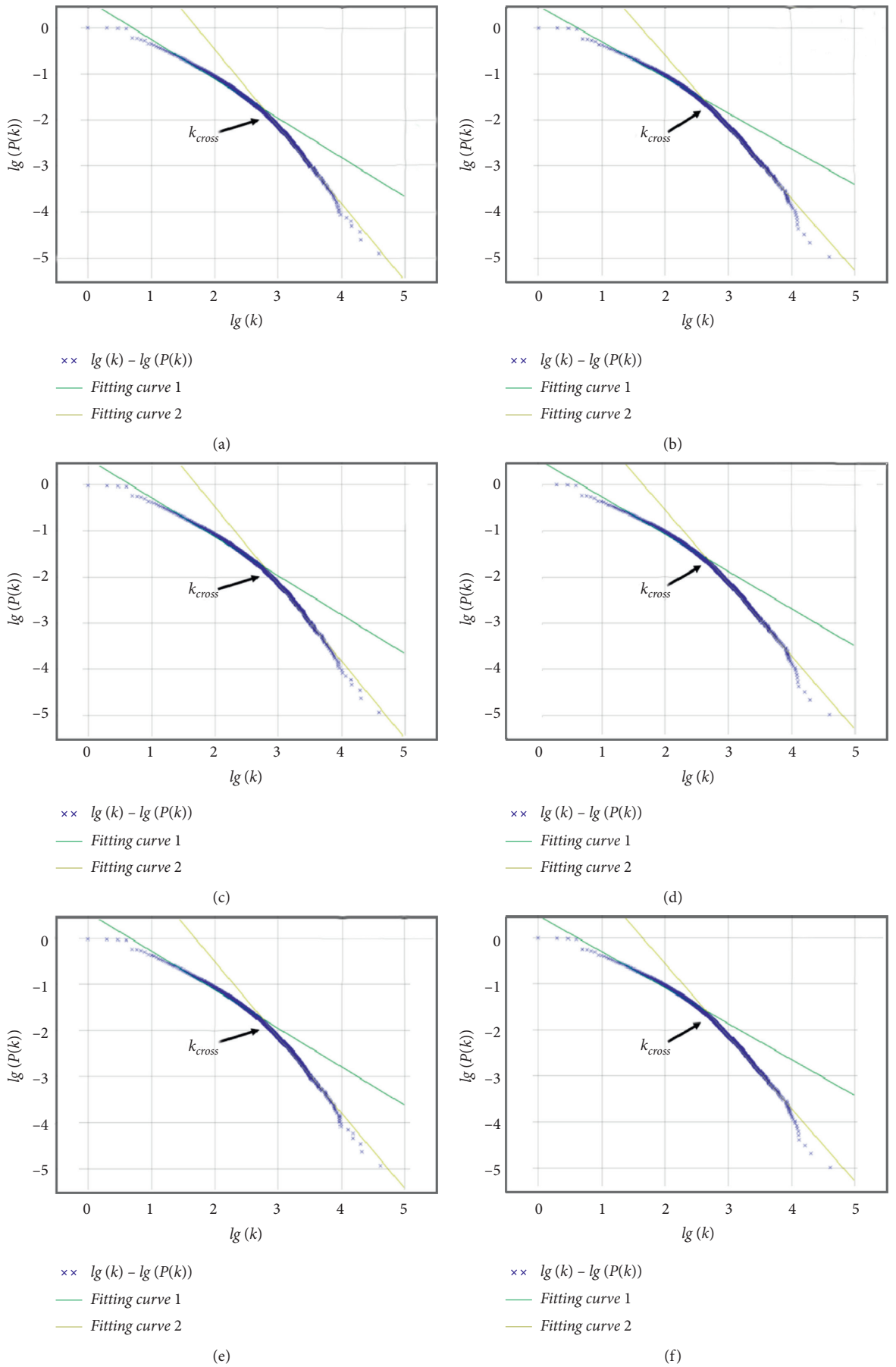


Figure 2: Cumulative degree distributions of all the built networks. (a) XIN_1. (b) CNA_1. (c) XIN_2. (d) CNA_2. (e) XIN_3. (f) CNA_3.

Table 3: Word length statistics in kernel lexicons (%).

	k_{cross}	P (k_{cross})	NKL
XIN_1	606	0.01470	1,193
CNA_1	420	0.02399	2,205
XIN_2	622	0.01442	1,187
CNA_2	494	0.02073	1,944
XIN_3	581	0.01613	1,350
CNA_3	466	0.02207	2,121

Table 4: Comparison on part-of-speech statistics (%).

Metric	Dataset theme					
	XIN_1	CNA_1	XIN_2	CNA_2	XIN_3	CNA_3
Noun	28.83	31.25	29.06	31.07	27.85	31.40
Verb	23.22	26.94	22.91	27.11	22.52	27.11
Adverb	6.87	3.53	6.74	7.00	7.11	6.51
Numeral	4.78	3.36	4.80	3.34	4.67	3.25
Gerund	4.44	3.67	4.38	4.38	3.78	3.30
Time	5.11	3.40	5.22	2.52	5.04	2.69
Noun of Place	3.69	3.31	3.88	2.88	4.96	3.30
Adjective	2.68	2.77	2.78	2.62	2.62	2.83
Quantifier	3.10	2.49	2.95	2.62	2.96	2.50
Preposition	3.35	2.22	3.29	2.52	3.04	2.36
Conjunction	2.01	2.04	2.02	2.16	2.07	2.07
Noun of Locality	2.18	1.72	2.19	1.90	2.30	1.74

Table 5: Word length statistics in kernel lexicons (%).

Word length	Dataset theme					
	XIN_1	CNA_1	XIN_2	CNA_2	XIN_3	CNA_3
1	25.40	27.76	24.26	28.24	24.96	28.52
2	68.73	66.85	69.17	67.28	67.41	66.20
3	5.11	4.85	5.73	4.12	6.15	4.86
4	0.34	0.27	0.34	0.21	0.89	0.19
5	0.42	0.27	0.51	0.15	0.59	0.24

Table 6: Comparison on motif analysis (%).

Word length	Dataset theme					
	XIN_1	CNA_1	XIN_2	CNA_2	XIN_3	CNA_3
Star	93.7959	91.3591	93.7661	91.2177	93.7512	91.1679
Chain	3.4099	4.8152	3.3738	4.7887	3.3632	4.7131
TLO	2.5563	3.4790	2.6098	3.6182	2.6256	3.7186
Box	0.0328	0.0493	0.0332	0.0512	0.0335	0.0523
SCQ	0.1875	0.2725	0.1980	0.2959	0.2059	0.3162
Clique	0.0172	0.0246	0.0188	0.0281	0.0202	0.0316

TLO: three-loop-out. SCQ: semiclique.

However, given the same corpus scale, the co-occurrence networks for traditional Chinese tend to have larger number of nodes, which may be due to the numerous one-to-many character/word mappings from simplified Chinese to traditional Chinese. Secondly, the kernel lexicons of traditional Chinese have more entries than those of simplified Chinese, which may be because that, in traditional Chinese, more ancient Chinese words are kept while less weak verbs are used. Thirdly, the motif analysis shows that there are no differences between the simplified Chinese networks and the corresponding traditional Chinese ones. In other words, simplified Chinese and traditional Chinese are semantically consistent.

Data Availability

The data used can be accessed at https://catalog.ldc.upenn.edu/LDC2007T38.

Conflicts of Interest

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

Authors' Contributions

All authors contributed equally to this paper. Zhongqiang Jiang designed main experiments; wrote the paper; improved the English expression, and corrected the typos and grammatical errors. Dongmei Zhao checked all the symbols, formulas, and algorithms and added some additional explanations during revision process. Jiangbin Zheng wrote the first draft; proposed the idea; participated in experimental discussion; and designed the overall paper structure. Yidong Chen provided guidance and helped to revise the paper structure during revision process.

Acknowledgments

This work was supported in part by the National Social Science Foundation of China under Grant 16AZD049.

References

[1] L. Wang, X. Wang, and J. Wu, "The correspondence simplified characters and traditional characters and the mutual conversion," *Journal of Chinese Information Processing*, vol. 4, 2013.

[2] P. Zhenjun and Y. Tianfang, "Chinese characters conversion system based on lookup table and statistical methods," *Computer Engineering and Applications*, vol. 51, no. 4, p. 24, 2015.

[3] H. Dai, "Linguistic analysis of the intelligent conversion system of simplified and traditional Chinese characters text," *Liaoning Normal University (Social Science Edition)*, vol. 39, no. 2, pp. 115–120, 2016.

[4] L. Wang, "Review of and reflections on the hot topics in the application of contemporary Chinese charactersl Chinese characters text," *Applied Linguistics*, no. 2, 2020.

[5] M.-H. Li, S.-H. Wu, Yi-C. Zeng, P.-C. Yang, and T. Ku, "Chinese characters conversion system based on lookup table and language model," *Computational Linguistics and Chinese Language Processing*, vol. 15, no. 1, pp. 19–36, 2010.

[6] J. Fei, "Comparative analysis of current Chinese characters across the Taiwan straits," *Language Application*, vol. 1993, no. 1, pp. 37–48, 1993.

[7] L. Li, "An analysis of the reasons for the differences in the forms of Chinese characters on both sides of the Taiwan straits," *Journal of Guangxi University*, vol. 20, no. 1, pp. 98–101, 1998.

[8] X. Liu, "*Study on the unification of Chinese characters across the Taiwan straits*," M.S. thesis, Northwest University, Kirkland, WA, USA, 2007.

[9] Y. Jiang, "Differences in Chinese vocabulary between the two sides of the taiwan straits and their reasons," *Jimei University Journal*, vol. 9, no. 3, pp. 31–37, 2006.

[10] X. Li and Z. Qiu, "Definement and treatment of difference words in cross-strait dictionaries-new problems in cross-strait co-edited Chinese dictionaries," *Language Application*, vol. 2012, no. 4, pp. 74–81, 2012.

[11] A. E. Motter, A. P. S. De Moura, Y.-C. Lai, and P. Dasgupta, "Topology of the conceptual network of language," *Physical Review E*, vol. 65, no. 6, Article ID 065102, 2002.

[12] Y. Li, L. Wei, W. Li, Y. Niu, and S. Luo, "Small-world patterns in Chinese phrase networks," *Chinese Science Bulletin*, vol. 50, no. 3, pp. 287–289, 2005.

[13] J. Li, J. Zhou, X. Luo, and Z. Yang, "Chinese lexical networks: the structure, function and formation," *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 21, pp. 5254–5263, 2012.

[14] R. F. I. Cancho and R. V. Solé, "The small world of human language," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 268, no. 1482, pp. 2261–2265, 2001.

[15] Z.-Yuan Liu and M.-Song Sun, "Chinese word cooccurrence network: its small world effect and scale-free property," *Journal of Chinese Information Processing*, vol. 21, no. 6, pp. 52–58, 2007.

[16] S. Zhou, G. Hu, Z. Zhang, and J. Guan, "An empirical study of Chinese language networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 12, pp. 3039–3047, 2008.

[17] W. Liang, Y. Shi, C. K. Tse, J. Liu, Y. Wang, and X. Cui, "Comparison of co-occurrence networks of the Chinese and English languages," *Physica A: Statistical Mechanics and Its Applications*, vol. 388, no. 23, pp. 4901–4909, 2009.

[18] H. Liu and W. Li, "Language clusters based on linguistic complex networks," *Chinese Science Bulletin*, vol. 55, no. 30, pp. 3458–3465, 2010.

[19] R. F. I. Cancho, R. V. Solé, and R. Köhler, "Patterns in syntactic dependency networks," *Physical Review E*, vol. 69, no. 5, Article ID 051915, 2004.

[20] H. Liu, "The complexity of Chinese syntactic dependency networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 12, pp. 3048–3058, 2008.

[21] Z.-Y. Liu, Y.-b. Zheng, and M.-S. Sun, "Complex network properties of Chinese syntactic dependency network," *Complex Systems and Complexity Science*, vol. 2, 2008.

[22] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: statistical analyses and a model of semantic growth," *Cognitive Science*, vol. 29, no. 1, pp. 41–78, 2005.

[23] X. F. Wang, Li Xiang, and G. R. Chen, *Theory of Complex Networks and its Application*, Tsinghua University, Beijing, China, 2006.

[24] H. Liu, "Statistical properties of Chinese semantic networks," *Science Bulletin*, vol. 54, no. 16, pp. 2781–2785, 2009.

[25] R. V. Solé, B. Corominas-Murtra, S. Valverde, and L. Steels, "Language networks: their structure, function, and evolution," *Complexity*, vol. 15, no. 6, pp. 20–26, 2010.

[26] M. Sigman and G. A. Cecchi, "Global organization of the wordnet lexicon," *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, pp. 1742–1747, 2002.

[27] Y. Li, L. Wei, Y. Niu, and J. Yin, "Structural organization and scale-free properties in Chinese phrase networks," *Chinese Science Bulletin*, vol. 50, no. 13, pp. 1305–1309, 2005.

[28] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world"networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[29] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[30] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature Genetics*, vol. 31, no. 1, pp. 64–68, 2002.

[31] C. Biemann, S. Roos, and K. Weihe, "Quantifying semantics using complex network analysis," *Proceedings of Coling 2012*, pp. 263–278, 2012.

[32] R. F. I. Cancho and R. V. Solé, "Two regimes in the frequency of words and the origins of complex lexicons: zipf's law revisited," *Journal of Quantitative Linguistics*, vol. 8, no. 3, pp. 165–173, 2001.

[33] Z. M. Griffin and K. Bock, "Constraint, word frequency, and the relationship between lexical processing levels in spoken word production," *Journal of Memory and Language*, vol. 38, no. 3, pp. 313–338, 1998.

[34] S. N. Dorogovtsev and J. F. F. Mendes, "Language as an evolving word web," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 268, no. 1485, pp. 2603–2606, 2001.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8818794, 10 pages
https://doi.org/10.1155/2020/8818794

Research Article

Stability Analysis for Nonlinear Impulsive Control System with Uncertainty Factors

Zemin Ren,[1] **Shiping Wen**,[2] **Qingyu Li**,[1] **Yuming Feng**,[3] **and Ning Tang**[3]

[1]*School of Mathematics, Physics and Data Science, Chongqing University of Science and Technology, Chongqing 401331, China*
[2]*Australian AI Institute, University of Technology Sydney, Ultimo, NSW 2007, Australia*
[3]*Key Laboratory of Intelligent Information Processing and Control, Chongqing Three Gorges University, Wanzhou, Chongqing 404100, China*

Correspondence should be addressed to Yuming Feng; yumingfeng25928@163.com

Received 13 August 2020; Revised 9 September 2020; Accepted 6 November 2020; Published 21 November 2020

Academic Editor: Michele Migliore

Considering the limitation of machine and technology, we study the stability for nonlinear impulsive control system with some uncertainty factors, such as the bounded gain error and the parameter uncertainty. A new sufficient condition for this system is established based on the generalized Cauchy–Schwarz inequality in this paper. Compared with some existing results, the proposed method is more practically applicable. The effectiveness of the proposed method is shown by a numerical example.

1. Introduction

Impulse control is based on impulsive differential equation and has many applications [1–6], such as digital communication system, artificial intelligence, and financial sector. In comparison with other methods, impulse control is more efficient in dealing with the stability of complex systems. The stability is an important property of the impulsive control system. Mathematically, its goal is to stabilize an unstable system by proper impulse. Up to now, a wide variety of achievements of impulse control theory have been developed in the literature [7–13].

Generally, there are at least one "impulsively" changeable state variable appearing in a plant P, which could be described as following control system:

$$\begin{cases} \dot{x}(t) = Ax + \phi(x), t \neq \tau_k, \\ \Delta x = U(k,x), t = \tau_k, \quad k = 1, 2, \ldots, \\ x(t_0) = x_0. \end{cases} \tag{1}$$

Here, $x \in \mathbb{R}^n$ denotes the state variable and $U(k,x)$ the impulse control law. We assume that the control instance satisfies

$$\begin{aligned} &t_0 < \tau_1 < \cdots\cdots < \tau_k < \tau_{k+1} \cdots, \\ &\lim_{k \longrightarrow \infty} \tau_k = \infty. \end{aligned} \tag{2}$$

A continuous nonlinear function $\phi(x)$: $\mathbb{R}^n \longrightarrow \mathbb{R}^n$ stratifies $\phi(t,0) = 0$ and $\|\phi(x)\| \leq L\|x\|$, where L is a Lipschitz constant. Many researchers have paid more attention on control system (1) and achieved many sufficient conditions for the stability of these systems [14–20]. Feng et al. consider single state-jumps impulsive systems with periodically time windows and give stability criteria for the new model [21]. To make the nonlinear impulse control system more reasonable, parameter uncertainty and bounded gain error are introduced into the corresponding impulsive differential equations [22–25]. Considering the limitation of machine and technology, Ma et al. investigate stabilization of impulse control systems with gain error and obtain a sufficient criterion for global exponential stability [26]. Zou et al. study impulsive systems with bounded gain error and form a sufficient criterion for the stability [27].

Cauchy–Schwarz inequality is an important tool to study nonlinear systems [28–31]. Recently, Peng et al. generalize the Cauchy–Schwarz inequality, which is used to deduce asymptotic stability for a class of nonlinear control systems [30]. Under the assumption $U(k,x) = BCx$, they study the after nonlinear system:

$$\begin{cases} \dot{x}(t) = Ax + \phi(x), t \neq \tau_k, \\ \Delta x = BCx, t = \tau_k, \quad k = 1, 2, \ldots, \\ x(t_0) = x_0, \end{cases} \tag{3}$$

where B and C are constant matrixes. Based on the generalized Cauchy–Schwarz inequality, we consider a class of nonlinear impulsive control systems with the parameter uncertainty, which can be written as follows:

$$\begin{cases} \dot{x}(t) = (A + \Delta A)x + \phi(x), t \neq \tau_k, \\ \Delta x = BCx, t = \tau_k, \quad k = 1, 2, \ldots, \\ x(t_0) = x_0. \end{cases} \tag{4}$$

Generally, one can express the parameter uncertainty as $\Delta A = GF(t)H$ with $F^T(t)F(t) \leq I$. Here, matrixes G and H are given with appropriate dimensions. In this paper, we will find some conditions for the stability of system (4). We organize the paper as follows. In Section 2, we briefly introduce some related lemmas. Then, we show sufficient conditions in Section 3. The simulation experiment is shown in Section 4, and conclusion is listed in Section 4.

2. Related Lemmas

First of all, we introduce some lemmas to be used later. Throughout this paper, $\lambda_{\max}$ and $\lambda_{\min}$ are denoted as the largest eigenvalue and the smallest eigenvalue, respectively. $\|\cdot\|$ is denoted as the Euclidian norm of matric or vector.

Lemma 1 (see [30]). *Suppose that P is positive definite. If $x, y \in R^n$ satisfy $|x^T y| \leq \sigma(x^T x)(y^T y)$ for a certain $\sigma \in [0, 1]$, then*

$$\left(x^T P y\right)^2 \leq \left(\frac{\lambda_{\max}(P) - g(\sqrt{\sigma})\lambda_{\min}(P)}{\lambda_{\max}(P) + g(\sqrt{\sigma})\lambda_{\min}(P)}\right)^2 \left(x^T P x\right)\left(y^T P y\right), \tag{5}$$

where $g(\sigma) = (1 - \sigma/1 + \sigma)$.

Lemma 2 (see [27]). *Suppose that Q is symmetric and positive definite; then, for any $A, B \in R^{n\times n}$ and $\mu > 0$,*

$$A^T QB + B^T QA \leq \mu A^T QA + \frac{1}{\mu} B^T QB. \tag{6}$$

Lemma 3 (see [32]). *Suppose that H is a real symmetric matrix; then,*

$$\lambda_{\min}(H)x^T x \leq x^T Hx \leq \lambda_{\max}(H)x^T x. \tag{7}$$

3. The Proposed Results

We give the main results in this section. Specifically, we will analyze the stabilization of impulsive control system (4) with bounded gain error and parameter uncertainty and then list some sufficient conditions which assure the origin of the related systems is asymptotically stable.

Theorem 1. *Suppose $P \in \mathbb{R}^{n\times n}$ be a symmetric and positive definite matrix, $\lambda_1 = \lambda_{\min}(P)$, $\lambda_2 = \lambda_{\max}(P)$, I be the identity matrix, λ_3 be the largest eigenvalue of $P^{-1}(PA + A^T P)$, and λ_4 be the largest eigenvalue of the matrix $P^{-1}(I + BC)^T P(I + BC)$. If*

$$\left|x^T(t)\phi(x(t))\right| \leq \sigma\left(x^T(t)x(t)\right)\left(\phi(x(t))^T \phi(x(t))\right), \tag{8}$$

for a certain $\sigma \in [0, 1]$ and

$$\left(\lambda_3 + 2\sqrt{\left(\frac{\lambda_2 \lambda_{\max}\left(G^T G\right)\lambda_{\max}\left(H^T H\right)}{\lambda_1}\right)} + 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\frac{\lambda_2}{\lambda_1}}\right)(\tau_{k+1} - \tau_k) \leq -\ln(\gamma\lambda_4), \tag{9}$$

where

$$g(\sigma) = \frac{1 - \sigma}{1 + \sigma}, \gamma > 1. \tag{10}$$

then, we obtain that the origin of impulsive control system (4) is asymptotically stable.

Proof. We choose the Lyapunov function as follows:

$$V(x(t)) = x^T(t)Px(t). \tag{11}$$

When $t \neq \tau_k$, we obtain Dini's derivative of $V(x(t))$ for impulsive control system (4) as follows:

$$\begin{aligned} D^+V(x(t)) &= 2x^T(t)P((A + \Delta A)x(t) + \phi(x(t))), \\ &= 2x^T(t)PAx(t) + 2x^T(t)P\Delta Ax(t) + 2x^T(t)P\phi(x(t)). \end{aligned} \tag{12}$$

Next, we will calculate the three parts of the above formula (12), respectively. The matrices $P^{-1}(PA + A^TP)$ and $P^{-0.5}(PA + A^TP)P^{-0.5}$ have the same eigenvalues. By Lemma 3, we have

$$\begin{aligned} 2x^T(t)PAx(t) &= x^T(t)\left(PA + A^TP\right)x(t), \\ &= \left(x^T(t)P^{0.5}\right)\left(P^{-0.5}\left(PA + A^TP\right)P^{-0.5}\right)\left(P^{0.5}x(t)\right), \\ &\le \lambda_3\left(x^T(t)P^{0.5}\right)\left(P^{0.5}x(t)\right), \\ &= \lambda_3 V(x(t)). \end{aligned} \tag{13}$$

According to the Cauchy–Schwarz inequality, we obtain

$$x^T(t)P\Delta Ax(t) \le \sqrt{\left(x^T(t)P^2x(t)\right)\left(x^T(t)\Delta A^T\Delta Ax(t)\right)}. \tag{14}$$

Since parameter uncertainty $\Delta A = GF(t)H$ and $F^T(t)F(t) \le I$, inequality (14) can be rewritten as

$$\begin{aligned} 2x^T(t)P\Delta Ax(t) &\le 2\sqrt{\left(x^T(t)P^2x(t)\right)\left(x^T(t)H^TF^T(t)G^TGF(t)Hx(t)\right)}, \\ &\le 2\sqrt{\left(\left(x^T(t)P^{1/2}\right)P\left(P^{1/2}x(t)\right)\right)\left(x^T(t)H^TF^T(t)G^TGF(t)Hx(t)\right)}, \\ &\le 2\sqrt{\left(\lambda_2V((x(t)))\right)\left(\lambda_{\max}\left(G^TG\right)x^T(t)H^TIHx(t)\right)}, \\ &\le 2\sqrt{\left(\lambda_2V(x(t))\right)\left(\lambda_{\max}\left(G^TG\right)\lambda_{\max}\left(H^TH\right)x^T(t)x(t)\right)}, \\ &= 2\sqrt{\left(\lambda_2V(x(t))\right)\left(\lambda_{\max}\left(G^TG\right)\lambda_{\max}\left(H^TH\right)\left(x^T(t)P^{1/2}\right)P^{-1}\left(P^{(1/2)}x(t)\right)\right)}, \\ &\le 2\sqrt{\left(\frac{\lambda_2\lambda_{\max}\left(G^TG\right)\lambda_{\max}\left(H^TH\right)}{\lambda_1}\right)V(x(t))}. \end{aligned} \tag{15}$$

According to Lemma 1, we obtain

$$\begin{aligned} 2x^T(t)P\phi(x(t)) &\le 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\left(x^T(t)Px(t)\right)\left(\phi(x(t))^TP\phi(x(t))\right)}, \\ &\le 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\lambda_2\left(x^T(t)Px(t)\right)\left(\phi(x(t))^T\phi(x(t))\right)}. \end{aligned} \tag{16}$$

Since $\|\phi(x)\| \le L\|x\|$, inequality (16) can be obtained as follows:

$$\begin{aligned} 2x^T(t)P\phi(x(t)) &\le 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\lambda_2\left(x^T(t)Px(t)\right)\left(x(t)^Tx(t)\right)}, \\ &\le 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\lambda_2\left(x^T(t)Px(t)\right)\left(x(t)^Tx(t)\right)}, \\ &\le 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\left(\frac{\lambda_2}{\lambda_1}\right)\left(x^T(t)Px(t)\right)\left(x(t)^TPx(t)\right)}, \\ &= 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\frac{\lambda_2}{\lambda_1}}V(x(t)). \end{aligned} \tag{17}$$

Combining inequalities (13), (15), and (17), we obtain

$$D^{+}V(x(t)) \leq \left(\lambda_3 + 2\sqrt{\left(\frac{\lambda_2 \lambda_{\max}\left(G^T G\right) \lambda_{\max}\left(H^T H\right)}{\lambda_1} \right)} + 2L \frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1} \sqrt{\frac{\lambda_2}{\lambda_1}} \right) V(x(t)). \tag{18}$$

When $t = \tau_k$, we compute the value of V as follows:

$$\begin{aligned} V(x(t) + BCx(t))|_{t=\tau_k} &= (x(t) + BCx(t))^T P(x(t) + BCx(t))|_{t=\tau_k}, \\ &= x(t)^T (I + BC)^T P(I + BC)x(t)|_{t=\tau_k}, \\ &= \left(x^T(t)P^{0.5}\right)\left(P^{-0.5}(I + BC)^T P(I + BC)P^{-0.5}\right)\left(P^{0.5}x(t)\right)|_{t=\tau_k}. \end{aligned} \tag{19}$$

It is known that the matrix $P^{-0.5}(I + BC)^T P(I + BC)P^{-0.5}$ has the same eigenvalues with the matrix $P^{-1}(I + BC)^T P(I + BC)$. Thus, it follows from (19) that

$$\begin{aligned} V(x(t) + BCx(t))|_{t=\tau_k} &\leq \lambda_4 \left(x^T(t)P^{0.5}\right)\left(P^{0.5}x(t)\right)|_{t=\tau_k}, \\ &= \lambda_4 V(x(t))|_{t=\tau_k}. \end{aligned} \tag{20}$$

Now, we analyze the following comparison system:

$$\begin{aligned} \dot{\omega} &= \left(\lambda_3 + 2\sqrt{\left(\frac{\lambda_2 \lambda_{\max}\left(G^T G\right) \lambda_{\max}\left(H^T H\right)}{\lambda_1} \right)} + 2L \frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1} \sqrt{\frac{\lambda_2}{\lambda_1}} \right) \omega(t),\ t \neq \tau_k, \\ \omega(\tau_k^+) &= \lambda_4 \omega(\tau_k), \\ \omega(\tau_0^+) &= \omega_0 \geq 0. \end{aligned} \tag{21}$$

According to the related conclusion (see Theorem 3 in [29]), we obtain that if

$$\int_{\tau_k}^{\tau_{k+1}} \left(\lambda_3 + 2\sqrt{\left(\frac{\lambda_2 \lambda_{\max}\left(G^T G\right) \lambda_{\max}\left(H^T H\right)}{\lambda_1} \right)} + 2L \frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1} \sqrt{\frac{\lambda_2}{\lambda_1}} \right) dt + \ln(\gamma\lambda_4) \leq 0,\ \gamma > 1. \tag{22}$$

The origin of impulsive control system (4) is asymptotically stable. □

Remark 1. If the parameter uncertainty $\Delta A = 0$, the condition of (9) became the result of Theorem 3.1 in reference [30]. Thus, the proposed method is a generalization of Peng's method.

In many practical applications, it is inevitable to put impulses with errors due to the limitation of machine and technology. So, we integrate the bounded gain error into the impulsive system (4). For simplicity, let $D = BC$. We rewrite

the corresponding system as

$$\begin{cases} x(t) = (A + \Delta A)x(t) + \phi(x(t)), t \neq \tau_k, \\ \Delta x(t) = (D + \Delta D)x(t), t = \tau_k, \quad k = 1, 2, \ldots, \\ x(t_0) = x_0, \end{cases} \tag{23}$$

where ΔD denotes the bounded gain error and has the following form: $\Delta D = mF(t)D$ with $m > 0$ and $F^T(t)F(t) \leq I$. It is easy to obtain a similar analysis from Theorem 1.

Theorem 2. *Let $P \in \mathbb{R}^{n\times n}$ be a symmetric and positive definite matrix, $\lambda_1 = \lambda_{\min}(P)$, $\lambda_2 = \lambda_{\max}(P)$, I be the identity matrix, and λ_3 be the largest eigenvalue of $P^{-1}(PA + A^TP)$. If*

$$\left|x^T(t)\phi(x(t))\right| \leq \sigma\left(x^T(t)x(t)\right)\left(\phi(x(t))^T\phi(x(t))\right). \tag{24}$$

for a certain $\sigma \in [0, 1]$ and

$$\left(\lambda_3 + 2\sqrt{\left(\frac{\lambda_2\lambda_{\max}\left(G^TG\right)\lambda_{\max}\left(H^TH\right)}{\lambda_1}\right)} + 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\frac{\lambda_2}{\lambda_1}}\right)(\tau_{k+1} - \tau_k) \leq -\ln(\gamma\lambda_4), \tag{25}$$

where

$$\lambda_4 = \frac{\lambda_2}{\lambda_1}\left((1 + \mu)\lambda_{\max}\left((I + D)^T(I + D)\right) + \left(1 + \frac{1}{\mu}\right)m^2\lambda_{\max}\left(D^TD\right)\right), \tag{26}$$

$$g(\sigma) = \frac{1 - \sigma}{1 + \sigma}, \gamma > 1. \tag{27}$$

Then, the origin of impulsive control system (23) is asymptotically stable.

Proof. We choose the following Lyapunov function as follows:

$$V(x(t)) = x^T(t)Px(t). \tag{28}$$

According to inequality (18), Dini's derivative of $V(x(t))$ for impulsive control system (23) is acquired as follows:

$$D^+V(x(t)) \leq \left(\lambda_3 + 2\sqrt{\left(\frac{\lambda_2\lambda_{\max}\left(G^TG\right)\lambda_{\max}\left(H^TH\right)}{\lambda_1}\right)} + 2L\frac{\lambda_2 - g(\sqrt{\sigma})\lambda_1}{\lambda_2 + g(\sqrt{\sigma})\lambda_1}\sqrt{\frac{\lambda_2}{\lambda_1}}\right)V(x(t)). \tag{29}$$

Then, we just need to compute $V(x(t) + (D + \Delta D)x(t))|_{t=\tau_k}$.

We perform some calculations on $V(x(t) + (D + \Delta D)x(t))|_{t=\tau_k}$ and obtain

$$\begin{aligned} V(x(t) + (D + \Delta D)x(t))|_{t=\tau_k} &= (x(t) + (D + \Delta D)x(t))^T P(x(t) + (D + \Delta D)x(t))|_{t=\tau_k}, \\ &= x(t)^T((I + D) + \Delta D)^T P((I + D) + \Delta D)x(t)|_{t=\tau_k}, \\ &\leq \lambda_2 x(t)^T((I + D) + \Delta D)^T((I + D) + \Delta D)x(t)|_{t=\tau_k}, \\ &\leq \lambda_2 x(t)^T\left((I + D)^T(I + D) + (I + D)^T\Delta D + \Delta D^T(I + D) + \Delta D^T\Delta D\right)x(t)|_{t=\tau_k}. \end{aligned} \tag{30}$$

By using Lemma 2 and $\Delta D = mF(t)D$, we rewrite inequality (30) as

$$\begin{aligned} V(x(t)+(D+\Delta D)x(t))|_{t=\tau_k} &\le \lambda_2 x^T(t)\Big((I+D)^T(I+D)+(I+D)^T\Delta D+\Delta D^T\Delta D+\Delta D^T(I+D)\Big)x(t)|_{t=\tau_k}, \\ &\le \lambda_2 x^T(t)\left((1+\mu)(I+D)^T(I+D)+\left(1+\frac{1}{\mu}\right)\Delta D^T\Delta D\right)x(t)|_{t=\tau_k}, \\ &= \lambda_2 x^T(t)\left((1+\mu)(I+D)^T(I+D)+\left(1+\frac{1}{\mu}\right)m^2D^TF^T(t)F(t)D\right)x(t)|_{t=\tau_k}. \end{aligned} \tag{31}$$

It follows from (15) that

$$x^T(t)x(t) = \left(x^T(t)P^{1/2}\right)P^{-1}\left(P^{1/2}x(t)\right) \le \frac{V(x(t))}{\lambda_1}. \tag{32}$$

Combine inequalities (31) and (32) and $F^T(t)F(t) \le I$, we obtain

$$\begin{aligned} V(x(t)+(D+\Delta D)x(t))|_{t=\tau_k} &= \lambda_2 x^T(t)\left((1+\mu)(I+D)^T(I+D)+\left(1+\frac{1}{\mu}\right)m^2D^TF^T(t)F(t)D\right)x(t)|_{t=\tau_k}, \\ &\le \lambda_2 x^T(t)\left((1+\mu)(I+D)^T(I+D)+\left(1+\frac{1}{\mu}\right)m^2D^TD\right)x(t)|_{t=\tau_k}, \\ &\le \frac{\lambda_2}{\lambda_1}\left((1+\mu)\lambda_{\max}\Big((I+D)^T(I+D)\Big)+\left(1+\frac{1}{\mu}\right)m^2\lambda_{\max}\Big(D^TD\Big)\right)V(x(t))|_{t=\tau_k}, \\ &= \lambda_4 V(x(t))|_{t=\tau_k}. \end{aligned} \tag{33}$$

Here, we emit the rest analysis process, which is similar to Theorem 1. Thus, from equalities (29) and (33), we obtain that if

$$\begin{aligned} &\left(\lambda_3+2\sqrt{\left(\frac{\lambda_2\lambda_{\max}\Big(G^TG\Big)\lambda_{\max}\Big(H^TH\Big)}{\lambda_1}\right)}+2L\frac{\lambda_2-g(\sqrt{\sigma})\lambda_1}{\lambda_2+g(\sqrt{\sigma})\lambda_1}\sqrt{\frac{\lambda_2}{\lambda_1}}\right)(\tau_{k+1}-\tau_k) \le -\ln(\gamma\lambda_4), \\ &\lambda_4 = \frac{\lambda_2}{\lambda_1}\left((1+\mu)\lambda_{\max}\Big((I+D)^T(I+D)\Big)+\left(1+\frac{1}{\mu}\right)m^2\lambda_{\max}\Big(D^TD\Big)\right), \end{aligned} \tag{34}$$

the origin of impulsive control system (23) is asymptotically stable. This completes the proof. □

4. A Numerical Example

In this section, we perform the proposed model on a numerical example to display its effectiveness. The example is produced by Qi and Chen [33]. Let $x = [x_1, x_2, x_3]^T$, $\phi(x) = [x_2x_3, -x_1x_3, x_1x_2]^T$, and

$$A = \begin{bmatrix} -a & a & 0 \\ c & -1 & 0 \\ 0 & 0 & -b \end{bmatrix}. \tag{35}$$

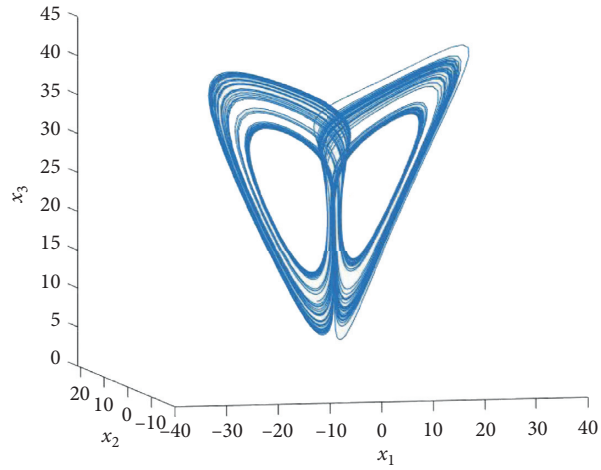


Figure 1: The chaotic phenomenon of system (36) with the initial condition: $x(0) = [3, 5, 10]^T$.

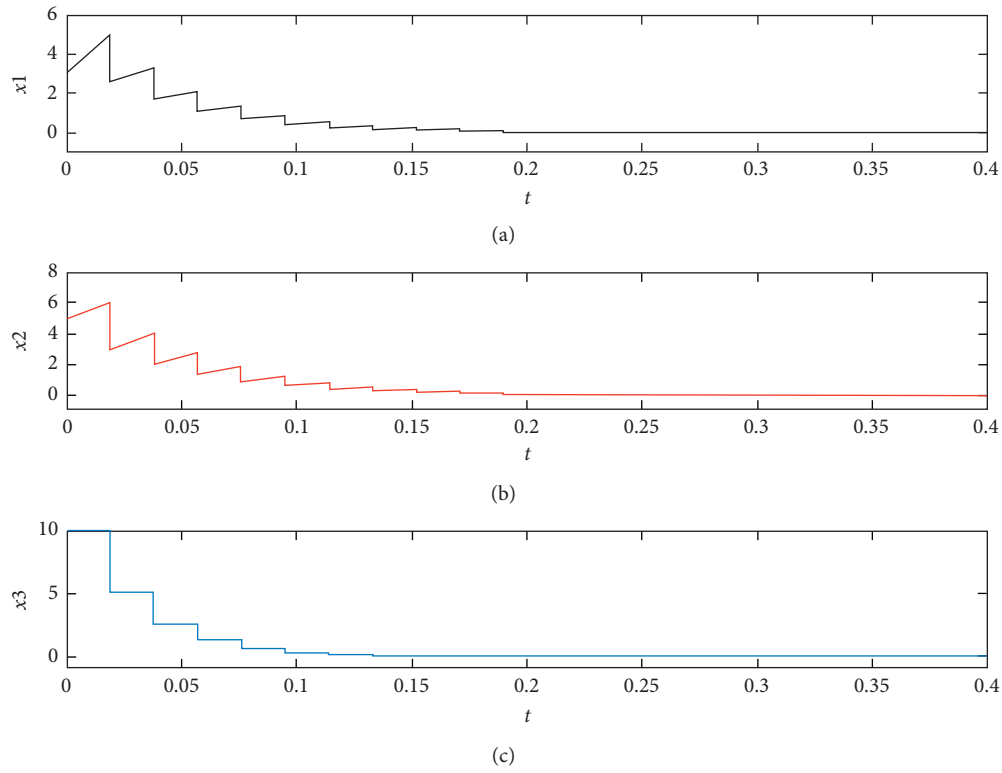


Figure 2: Time response curves for the controlled system (36) with the parameter uncertainty.

The corresponding state equation can be described as

$$\dot{x} = Ax + \phi(x). \tag{36}$$

According to the strategy of [33], some parameters of this system are set as $a = 35$, $b = (8/3)$, and $c = 25$. From Figure 1, we can see that system (36) is chaotic for the initial condition: $x(0) = [3, 5, 10]^T$.

After simple calculation, we obtain that

$$\begin{aligned} \|\phi(x)\| &= \sqrt{(x_2x_3)^2 + (x_1x_3)^2 + (x_1x_2)^2}, \\ &\le \max\{|x_1|, |x_2|, |x_3|\}\sqrt{x_1^2 + x_2^2 + x_3^2}, \\ &= \max\{|x_1|, |x_2|, |x_3|\}\|x\|. \end{aligned} \tag{37}$$

From Figure 1, we can intuitively find $\max\{|x_1|, |x_2|, |x_3|\} \le 45$. Combining with inequality (37), the parameter L can be set as 45. Since

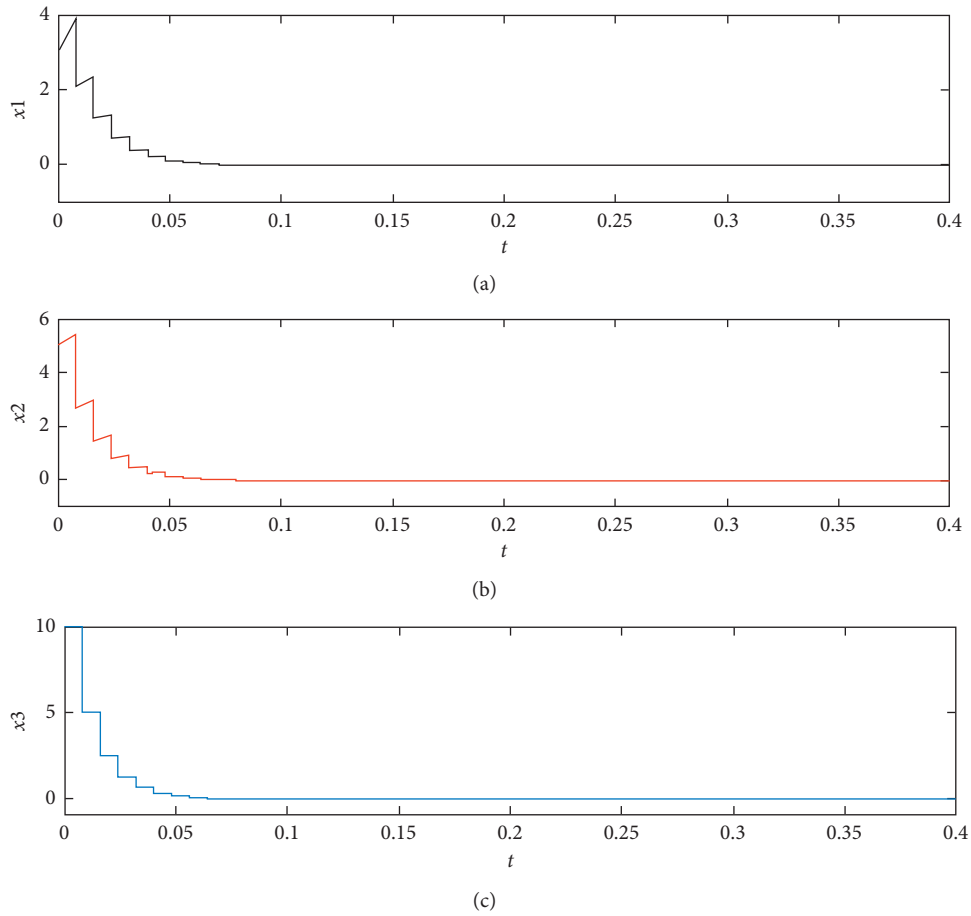


Figure 3: Time response curves for the controlled system (36) with parameter uncertainty and gain error.

$$\left|x^T\phi(x)\right|^2 \le \frac{1}{9}\left(x^Tx\right)\left(\phi(x)^T\phi(x)\right), \tag{38}$$

the parameter σ is chosen as $\sigma = (1/9)$. In this section, some matrices are chosen as follows:

$$G = H = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix},$$

$$P = C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{39}$$

$$B = \begin{bmatrix} -0.5 & -0.01 & 0.02 \\ -0.01 & -0.5 & 0 \\ 0.02 & 0 & -0.5 \end{bmatrix}.$$

Thus, the parameter uncertainty can be formed as

$$\Delta A = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} 0.1\sin(t) & 0 & 0 \\ 0 & 0.1\sin(t) & 0 \\ 0 & 0 & 0.1\sin(t) \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}. \tag{40}$$

According to Theorem 1, we calculate $\lambda_3 = 32.9638$ and $\lambda_4 = 0.2729$. It follows from (8) that

$$\tau_{k+1} - \tau_k \le -\frac{\ln(\gamma\lambda_4)}{63.4638}. \tag{41}$$

If $\gamma = 1.1$, it yields $\tau_{k+1} - \tau_k \le 0.0190$. We choose $\tau_{k+1} - \tau_k = 0.0190$ and show the simulation result in Figure 2. The impulsive control system (36) is asymptotically stable.

Next, we consider the controlled system (36) with the parameter uncertainty and the bounded gain error. The gain error is detailed as $\Delta D = m \sin(t)D$ in this section. We perform some similar calculation on (25) and obtain $\lambda_3 = 32.9638$. We choose $\mu = 1$ and then obtain $\lambda_4 = 0.5458(1 + m^2)$ from (26). Let $\gamma = 1.1$ and $m = 0.05$; then,

$$\tau_{k+1} - \tau_k \leq 0.0080. \tag{42}$$

Thus, we choose $\tau_{k+1} - \tau_k = 0.0080$ and show the experimental result in Figure 3. From this figure, we can obtain that the impulsive control system (36) is asymptotically stable.

5. Conclusion

We study the asymptotic stability of impulsive control systems with some uncertainty factors, such as the bounded gain error and the parameter. The proposed sufficient condition is established based on the generalized Cauchy–Schwarz inequality. We think the proposed issue is more practically applicable than some existing ones.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All the authors contributed equally to write this paper. Z. Ren, S. Wen, and Y. Feng have proposed the main idea of the paper. Z. Ren, Q. Li, and N. Tang have proved the main theory. All authors read and approved the final manuscript.

Acknowledgments

This work was partially supported by the Chongqing Natural Science Foundation of China (cstc2019jcyj-msxmX0060), Key Project of Science and Technology Research Program of Chongqing Education Commission of China (KJZD-K202001503), Key Project of Chongqing Municipal Key Laboratory of Institutions of Higher Education ([2017]3), National Natural Science Foundation of China (61601068 and 62061016), and Foundation of Chongqing Development and Reform Commission (2017[1007]).

References

[1] P. Naghshtabrizi, J. P. Hespanha, and A. R. Teel, "Exponential stability of impulsive systems with application to uncertain sampled-data systems," *Systems & Control Letters*, vol. 57, no. 5, pp. 378–385, 2008.

[2] J. Sun, F. Qiao, and Q. Wu, "Impulsive control of a financial model," *Physics Letters A*, vol. 335, no. 4, pp. 282–288, 2005.

[3] X. L. Chai, Z. H. Gan, and C. X. Shi, "Impulsive synchronization and adaptive-impulsive synchronization of a novel financial hyperchaotic system," *Mathematical Problems in Engineering*, vol. 2013, Article ID 751616, , 2013.

[4] S. Gao, L. Chen, and Z. Teng, "Impulsive vaccination of an SEIRS model with time delay and varying total population size," *Bulletin of Mathematical Biology*, vol. 69, no. 2, pp. 731–745, 2007.

[5] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3310–3315, 2015.

[6] Z. G. Zeng and J. Wang, "Design and analysis of high-capacity associative memories based on a class of discrete-time recurrent neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 6, pp. 1525–1536, 2008.

[7] Y. Feng, X. Yang, Q. Song, and J. Cao, "Synchronization of memristive neural networks with mixed delays via quantized intermittent control," *Applied Mathematics and Computation*, vol. 339, pp. 874–887, 2018.

[8] X. Hu and L. Nie, "Exponential stability of nonlinear systems with impulsive effects and disturbance input," *Advances in Difference Equations*, vol. 2018, Article ID 354, 2018.

[9] Q. Song, H. Yan, Z. Zhao, and Y. Liu, "Global exponential stability of complex-valued neural networks with both time-varying delays and impulsive effects," *Neural Networks*, vol. 79, pp. 108–116, 2016.

[10] Y. Feng, X. Xiong, R. Tang, and X. Yang, "Exponential synchronization of inertial neural networks with mixed delays via quantized pinning control," *Neurocomputing*, vol. 310, no. 8, pp. 165–171, 2018.

[11] Q. Song, H. Yan, Z. Zhao, and Y. Liu, "Global exponential stability of impulsive complex-valued neural networks with both asynchronous time-varying and continuously distributed delays," *Neural Networks*, vol. 81, pp. 1–10, 2016.

[12] X. Yang, J. Cao, and Z. Yang, "Synchronization of coupled reaction-diffusion neural networks with time-varying delays via pinning-impulsive controller," *SIAM Journal on Control and Optimization*, vol. 51, no. 5, pp. 3486–3510, 2013.

[13] X. Yang, J. Lam, D. W. C. Ho, and Z. Feng, "Fixed-Time synchronization of complex networks with impulsive effects via nonchattering control," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5511–5521, 2017.

[14] J. Lu and D. W. C. Ho, "Globally exponential synchronization and synchronizability for general dynamical networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 2, pp. 350–361, 2010.

[15] J. Lu, C. Ding, J. Lou, and J. Cao, "Outer synchronization of partially coupled dynamical networks via pinning impulsive controllers," *Journal of the Franklin Institute*, vol. 352, no. 11, pp. 5024–5041, 2015.

[16] X. Wang, C. Li, T. Huang, and X. Pan, "Impulsive control and synchronization of nonlinear system with impulse time window," *Nonlinear Dynamics*, vol. 78, no. 4, pp. 2837–2845, 2014.

[17] D. Yang, G. Qiu, and C. Li, "Global exponential stability of memristive neural networks with impulse time window and time-varying delays," *Neurocomputing*, vol. 171, pp. 1021–1026, 2016.

[18] J.-L. Wang, H.-N. Wu, and L. Guo, "Stability analysis of reaction-diffusion Cohen-Grossberg neural networks under impulsive control," *Neurocomputing*, vol. 106, pp. 21–30, 2013.

[19] J. Lu, D. W. C. Ho, and J. Cao, "A unified synchronization criterion for impulsive dynamical networks," *Automatica*, vol. 46, no. 7, pp. 1215–1221, 2010.

[20] H. Wang, S. Duan, C. Li, L. Wang, and T. Huang, "Globally exponential stability of delayed impulsive functional differential systems with impulse time windows," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1655–1665, 2016.
[21] Y. Feng, C. Li, and T. Huang, "Periodically multiple state-jumps impulsive control systems with impulse time windows," *Neurocomputing*, vol. 193, pp. 7–13, 2016.
[22] F. Chen, H. Wang, and C. Li, "Impulsive control of memristive chaotic systems with impulsive time window," *Mathematical Problems in Engineering*, vol. 2015, Article ID 927327, , 2015.
[23] H. Wang, S. Duan, C. Li, L. Wang, and T. Huang, "Stability criterion of linear delayed impulsive differential systems with impulse time windows," *International Journal of Control, Automation and Systems*, vol. 14, no. 1, pp. 174–180, 2016.
[24] X. Wang, J. Yu, C. Li, H. Wang, T. Huang, and J. Huang, "Robust stability of stochastic fuzzy delayed neural networks with impulsive time window," *Neural Networks*, vol. 67, pp. 84–91, 2015.
[25] Y. Zhou, C. Li, T. Huang, and X. Wang, "Impulsive stabilization and synchronization of Hopfield-type neural networks with impulse time window," *Neural Computing and Applications*, vol. 28, no. 4, pp. 775–782, 2017.
[26] T. Ma and F. Zhao, "Impulsive stabilization of a class of nonlinear system with bounded gain error," *Chinese Physics B*, vol. 23, no. 12, Article ID 150504, 2014.
[27] L. Zou, Y. Peng, Y. Peng, Y. Feng, and Z. Tu, "Impulsive control of nonlinear systems with impulse time window and bounded gain error," *Nonlinear Analysis: Modelling and Control*, vol. 23, no. 1, pp. 40–49, 2018.
[28] T. Yang, "Impulsive control," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1081–1083, 1999.
[29] T. Yang, :*Impulsive Control Theory"*, Springer, Berlin, Germany, 2001.
[30] Y. Peng, J. Wu, L. Zou, Y. Feng, and Z. Tu, "A generalization of the cauchy-schwarz inequality and its application to stability analysis of nonlinear impulsive control systems," *Complexity*, vol. 2019, Article ID 6048909, 7 pages, 2019.
[31] X. L. Hu and J. Wang, "Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1487–1499, 2006.
[32] R. A. Horn and C. R. Johnson, *"Matrix Analysis"* Cambridge University Press, Cambrige, England, 1985.
[33] G. Qi, G. Chen, S. Du, Z. Chen, and Z. Yuan, "Analysis of a new chaotic system," *Physica A: Statistical Mechanics and its Applications*, vol. 352, no. 2–4, pp. 295–308, 2005.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8846438, 8 pages
https://doi.org/10.1155/2020/8846438

Research Article

Adaptive State Observer Design for Dynamic Links in Complex Dynamical Networks

Zilin Gao,[1,2] **Jiang Xiong**,[1,2] **Jing Zhong**,[1,2] **Fuming Liu**,[1,2] **and Qingshan Liu**[3]

[1]*Key Laboratory of Intelligent Information Processing and Control of Chongqing Municipal Institutions of Higher Education, Chongqing Three Gorges University, Chongqing 404100, China*
[2]*School of Computer Science and Engineering, Chongqing Three Gorges University, Chongqing 404100, China*
[3]*School of Mathematics, Southeast University, Nanjing 210009, China*

Correspondence should be addressed to Zilin Gao; gaozilin321@163.com

Received 19 May 2020; Revised 15 June 2020; Accepted 19 August 2020; Published 27 October 2020

Academic Editor: Michele Migliore

The state observer for dynamic links in complex dynamical networks (CDNs) is investigated by using the adaptive method whether the networks are undirected or directed. In this paper, a complete network model is proposed, which is composed of two coupled subsystems called nodes subsystem and links subsystem, respectively. Especially, for the links subsystem, associated with some assumptions, the state observer with parameter adaptive law is designed. Compared to the existing results about the state observer design of CDNs, the advantage of this method is that a estimation problem of dynamic links is solved in directed networks for the first time. Finally, the results obtained in this paper are demonstrated by performing a numerical example.

1. Introduction

In recent past decades, the research on CDNs has become a hot topic in many fields [1–4]. From the perspective of large system, a complete CDN contains many nodes and links (weights of connections between nodes), which implies that a complete CDN is composed of the nodes subsystem and links subsystem, and the two subsystems are usually coupled with each other [5–7]. It is worth noting that the existing researches mainly focus on the nodes subsystem because some behaviors are reflected by nodes such as synchronization [8, 9], stabilization [10, 11], and consensus [12, 13].

From the above results about the synchronization, stabilization, consensus, or other problems of CDNs, it is easy to see all states in CDNs, including the states of nodes and links, are required to be measured accurately. However, this assumption is too hard to be satisfied in practice because of the influence of external environment, measurement costs, and technical constraints [14]. Thus, constructing state observers for the CDNs to estimate the unknown states is very necessary and important. Fortunately, some scholars have discussed the state estimation problems of CDNs and obtained some research results, including cases with the coupling time delays [15, 16], packet loss [17, 18], stochastic noisy disturbance [19], and uncertain coupling strength [20].

However, the above results only consider the estimation problems of the states in nodes subsystem, and assume that the links between nodes are known. It implies that the measurement and state estimation problems of links in the CDNs are ignored. In fact, due to the limitation of measurement methods, the state values of links in CDNs are more difficult to be measured accurately in practical situation, compared to the states of nodes. Hence, only a few papers have studied and discussed the effective measurement problem of the links between individuals (nodes), and the measurement method mainly depends on the physical interaction between individuals [21] or the adaptive weights of links [22]. Similar to the state values of nodes, not all state values of links' weights can be measured and obtained. Therefore, it is necessary to design observers to estimate the unmeasured state values of links. As we know, there is only one paper to have discussed the state estimation problem of dynamic links in CDNs [23]. Unfortunately, the method proposed in [23] is only effective for undirected networks

and cannot solve the estimation problem of dynamic links in directed networks.

Inspired by the above discussions, this paper mainly focuses on the state observer design for dynamic links in directed networks. Specifically, a mathematical model for a class of directed CDNs is proposed, which is described by both the nodes subsystem and links subsystem with coupling between the two subsystems, and we have designed a state observer for the links subsystem by using the adaptive method. This means that a state estimation problem of dynamic links in directed networks is solved for the first time, which is also regarded as the biggest contribution of this paper.

The rest of this paper is organized as follows: in Section 2, a complete CDN model is proposed, which is composed of the nodes subsystem and links subsystem with outputs; Section 3 introduces the design process of state observer for the links subsystem; in Section 4, the simulation example is presented and used to demonstrate the effectiveness of this method; finally, the conclusions are given in Section 5.

1.1. Notations. The n-dimensional Euclidean space is denoted as R^n, the set of $n \times n$ real matrices is denoted as $R^{n\times n}$, the Euclidean norm of a vector or a matrix is denoted as $\|\cdot\|$, and the transpose of matrix A and n-dimensional identity matrix is denoted as A^{T} and I_n, respectively.

2. Preliminaries and Model Description

If the states of nodes and links in CDNs evolve over time, then the mathematical model of CDNs, including directed and undirected networks, can be described by both the nodes subsystem and links subsystem, where the two subsystems are coupled with each other. In this paper, we only consider the case that each node is n-dimensional continuous system in CDNs with N nodes, then the nodes subsystem and links subsystem can be described by vector differential equations and matrix differential equation as follows, respectively:

$$\dot{x}_i = A_i x_i + B_i f_i(x_i) + c_i \sum_{j=1}^{N} p_{ij}(t) H_j(x_j), \quad i = 1, 2, \ldots, N, \tag{1}$$

$$\begin{cases} \dot{P} = \Theta_1 P + P\Theta_2^{\mathrm{T}} + G(x), \\ Y_1 = \Upsilon P, Y_2 = \Upsilon P^{\mathrm{T}}, \end{cases} \tag{2}$$

where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^{\mathrm{T}} \in R^n$ is the state vector of node i; the constant matrices $A_i \in R^{n\times n}$ and $B_i \in R^{n\times m}$; the vector functions $f_i(x_i) = [f_{i1}(x_i), f_{i2}(x_i), \ldots, f_{im}(x_i)]^{\mathrm{T}}$ and $H_j(x_j) = [H_{j1}(x_j), H_{j2}(x_j), \ldots, H_{jn}(x_j)]^{\mathrm{T}}$; $c_i > 0$ is a known constant, which denotes the common connection strength of node i in the network; the constant matrices $\Theta_1 \in R^{N\times N}$ and $\Theta_2 \in R^{N\times N}$; the coupling matrix $G(x) \in R^{N\times N}$, and $x = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}, \ldots, x_N^{\mathrm{T}}]^{\mathrm{T}} \in \Lambda \subseteq R^{Nn}$, where Λ is a bounded and closed set in R^{Nn}; the output matrix $\Upsilon \in R^{N_1\times N}$ is given; and the links matrix $P = P(t) = (p_{ij}(t))_{N\times N}$, where the state variable $p_{ij}(t)$ denotes the weight of link from node j to node i. Especially, $p_{ji} = p_{ij}$ for undirected networks, and at least, one pair i, j such that $p_{ji} \neq p_{ij}$ for directed networks. In addition, if $i = j$, then p_{ij} denotes the link strength of node i itself.

For the CDNs composed of subsystems (1) and (2), the following instructions are given:

(1) The background of dynamic links is given as follows. For example, the biological neural networks consist of neurons (nodes) and synapse (links), and Gamma oscillations in neurons may cause the synaptic facilitation, which is regarded as a dynamic behavior of the links [5, 6, 24]. Similarly, the web winding systems can be regarded to be composed of motors (nodes) and the web (links), and the regulation values of web tensions vary with the speed of the motors, which is also regarded as a dynamic behavior of links [25]. In the above examples, the state values of links need to be measured by some sensors.

(2) The existing research results show that the nodes in networks can emerge synchronization or stabilization phenomenon with the help of the links, which mean that the nodes are the main body of synchronization and stabilization [8–11]. In contrast, the links as another part of networks can also emerge some characteristic phenomena in many real networks, such as the structural balance in social networks [5, 6, 26]. It is worth noting that the paper [26] has researched on structural balance by using the Riccati matrix differential equation, and the reason is that this type of equation is more easily to emerge the phenomenon of structural balance. In view of this, we choose linear Riccati matrix differential equation to describe the links subsystem. Clearly, the model of CDNs, composed of both nodes subsystem (1) and links subsystem (2), can help us to understand and explain the dynamic behaviors of networks in a better way.

(3) The subsystem (2) is used to describe dynamic change of links' weights in the CDNs, and in general, the CDNs are directed. However, if $\Theta_1 = \Theta_2$ and $G(x) = (G(x))^{\mathrm{T}}$, then we can obtain $P = P^{\mathrm{T}}$, while the initial value of the state in subsystem (2) satisfies $P(0) = (P(0))^{\mathrm{T}}$. Hence, the subsystems (1) and (2) can be used to describe both undirected and directed networks (the undirected networks can be regarded as a special case of directed networks). To the best of my knowledge, there is only one paper to have solved the state estimation problem of links subsystem [23]. However, this method is only effective for undirected networks, but not for directed networks. This drives us to study estimation problems of dynamic links in directed networks.

(4) It is difficult to accurately measure all states of the links between individuals (nodes) in practical applications, which imply that only partial states in (2) can be measured accurately and made available ($N_1 < N$). It is worth noting that the precise

measurement of the partial states is bidirectional; that is, if $p_{ij}(t)$ is measurable, then $p_{ji}(t)$ must also be measurable. That is why the two outputs Y_1 and Y_2 appear in (2).

Now, some useful definitions and operators involved in this paper will be introduced as follows.

Definition 1 (see [27]). The application vec: $R^{k\times l} \longrightarrow R^{kl}$ is defined by

$$\mathrm{vec}(H) = [h_{11}, \ldots, h_{1l}, h_{21}, \ldots, h_{2l}, \ldots, h_{k1}, \ldots, h_{kl}]^{\mathrm{T}}, \tag{3}$$

where the matrix $H = (h_{ij})_{k\times l}$ is called the vectorization operator.

Definition 2 (see [27]). If there are two matrices $H \in R^{k\times l}$ and $Z \in R^{c\times d}$, then the Kronecker product of H and Z is denoted as $H \otimes Z \in R^{kc\times ld}$ and defined as follows:

$$H \otimes Z = \begin{pmatrix} h_{11}Z & h_{12}Z & \cdots & h_{1l}Z \\ h_{21}Z & h_{22}Z & \cdots & h_{2l}Z \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ h_{k1}Z & h_{k2}Z & \cdots & h_{kl}Z \end{pmatrix}. \tag{4}$$

By using Definitions 1 and 2, the following basic properties about Kronecker product and operator $\mathrm{vec}(\cdot)$ can be obtained and shown as follows [27]:

(1) $(H \otimes S)(X \otimes W) = (HX) \otimes (SW)$
(2) $(H \otimes W)^{\mathrm{T}} = H^{\mathrm{T}} \otimes W^{\mathrm{T}}$
(3) $(S \otimes X)^{-1} = S^{-1} \otimes X^{-1}$
(4) $\mathrm{vec}(HSW) = (H \otimes W^{\mathrm{T}})\mathrm{vec}(S)$
(5) $\mathrm{vec}(HS + SW) = (H \otimes I + I \otimes W^{\mathrm{T}})\mathrm{vec}(S)$

S and X are the matrices with compatible dimensions, and I represents the identity matrix with compatible dimensions. Especially, it is assumed that both S and X are invertible in property (3).

According to Definitions 1 and 2 and their corresponding properties, the Riccati differential equation (2) can be rewritten as

$$\begin{cases} \mathrm{vec}(\dot{P}) = A\mathrm{vec}(P) + \mathrm{vec}(G(x)), \\ \mathrm{vec}(Y_1) = C_1\mathrm{vec}(P), \mathrm{vec}(Y_2) = C_1\mathrm{vec}\left(P^{\mathrm{T}}\right), \end{cases} \tag{5}$$

where $A = \Theta_1 \otimes I_N + I_N \otimes \Theta_2$ and $C_1 = \Upsilon \otimes I_N$.

Assumption 1. For the links subsystem (2), the double matrices (Θ_1, Υ) and (Θ_2, Υ) are completely stable.

If Assumption 1 is true, then we can obtain matrices $K_1 \in R^{N\times N_1}$ and $K_2 \in R^{N\times N_1}$, which can make $\Theta_1 + K_1\Upsilon$ and $\Theta_2 + K_2\Upsilon$ to be Hurwitz stable, respectively. Thus, as long as any matrices $Q_1 > 0$ and $Q_2 > 0$ are given, there must be positive definite matrices $M_1 \in R^{N\times N}$ and $M_2 \in R^{N\times N}$ that satisfy the following two Lyapunov equations, respectively:

$$(\Theta_1 + K_1\Upsilon)^{\mathrm{T}}M_1 + M_1(\Theta_1 + K_1\Upsilon) = -Q_1, \tag{6}$$

$$(\Theta_2 + K_2\Upsilon)^{\mathrm{T}}M_2 + M_2(\Theta_2 + K_2\Upsilon) = -Q_2. \tag{7}$$

Lemma 1. *If Assumption 1 is true, then the following Lyapunov equations*

$$\begin{aligned} (\Theta_1 \otimes I_N + \widetilde{K}_1C_1)^{\mathrm{T}}\widetilde{M} + \widetilde{M}(\Theta_1 \otimes I_N + \widetilde{K}_1C_1) &= -\widetilde{Q}_1, \\ (I_N \otimes \Theta_2 + \widetilde{K}_2C_2)^{\mathrm{T}}\widetilde{M} + \widetilde{M}(I_N \otimes \Theta_2 + \widetilde{K}_2C_2) &= -\widetilde{Q}_2, \end{aligned} \tag{8}$$

hold, where $\widetilde{M} = M_1 \otimes M_2$, $\widetilde{Q}_1 = Q_1 \otimes M_2$, $\widetilde{Q}_2 = M_1 \otimes Q_2$, $\widetilde{K}_1 = K_1 \otimes I_N$, $\widetilde{K}_2 = I_N \otimes K_2$, *and* $C_2 = I_N \otimes \Upsilon$. *Clearly,* $\widetilde{M} > 0$, $\widetilde{Q}_1 > 0$, *and* $\widetilde{Q}_2 > 0$.

Proof. If Assumption 1 holds, then the following equations can be obtain from (6) and (7):

$$\left[(\Theta_1 + K_1\Upsilon)^{\mathrm{T}}M_1\right] \otimes I_N + [M_1(\Theta_1 + K_1\Upsilon)] \otimes I_N = -Q_1 \otimes I_N, \tag{9}$$

$$I_N \otimes \left[(\Theta_2 + K_2\Upsilon)^{\mathrm{T}}M_2\right] + I_N \otimes [M_2(\Theta_2 + K_2\Upsilon)] = -I_N \otimes Q_2. \tag{10}$$

Using the properties of Kronecker product, (9) and (10) can be rewritten as

$$\begin{aligned} &[\Theta_1 \otimes I_N + (K_1 \otimes I_N)(\Upsilon \otimes I_N)]^{\mathrm{T}}(M_1 \otimes I_N) \\ &\quad + (M_1 \otimes I_N)[\Theta_1 \otimes I_N + (K_1 \otimes I_N)(\Upsilon \otimes I_N)] = -Q_1 \otimes I_N, \\ &[I_N \otimes \Theta_2 + (I_N \otimes K_2)(I_N \otimes \Upsilon)]^{\mathrm{T}}(I_N \otimes M_2) \\ &\quad + (I_N \otimes M_2)[I_N \otimes \Theta_2 + (I_N \otimes K_2)(I_N \otimes \Upsilon)] = -I_N \otimes Q_2. \end{aligned} \tag{11}$$

Thus, we can get

$$\begin{aligned} &[\Theta_1 \otimes I_N + \widetilde{K}_1C_1]^{\mathrm{T}}(M_1 \otimes I_N) + (M_1 \otimes I_N)[\Theta_1 \otimes I_N + \widetilde{K}_1C_1] \\ &\quad = -Q_1 \otimes I_N, \end{aligned} \tag{12}$$

$$\begin{aligned} &[I_N \otimes \Theta_2 + \widetilde{K}_2C_2]^{\mathrm{T}}(I_N \otimes M_2) + (I_N \otimes M_2)[I_N \otimes \Theta_2 + \widetilde{K}_2C_2] \\ &\quad = -I_N \otimes Q_2. \end{aligned} \tag{13}$$

If we multiply both sides of the equalities (12) and (13) by $(I_N \otimes M_2)$ and $(M_1 \otimes I_N)$ from right, respectively, then we get that

$$\begin{aligned} &[\Theta_1 \otimes I_N + \widetilde{K}_1C_1]^{\mathrm{T}}(M_1 \otimes I_N)(I_N \otimes M_2) \\ &\quad + (M_1 \otimes I_N)[\Theta_1 \otimes I_N + \widetilde{K}_1C_1](I_N \otimes M_2) \\ &\quad = -(Q_1 \otimes I_N)(I_N \otimes M_2), \end{aligned} \tag{14}$$

$$\begin{aligned} &[I_N \otimes \Theta_2 + \widetilde{K}_2C_2]^{\mathrm{T}}(I_N \otimes M_2)(M_1 \otimes I_N) \\ &\quad + (I_N \otimes M_2)[I_N \otimes \Theta_2 + \widetilde{K}_2C_2](M_1 \otimes I_N) \\ &\quad = -(I_N \otimes Q_2)(M_1 \otimes I_N). \end{aligned} \tag{15}$$

It is noticed that $(M_1 \otimes I_N)(I_N \otimes M_2) = M_1 \otimes M_2 = (I_N M_1) \otimes (M_2 I_N) = (I_N \otimes M_2)(M_1 \otimes I_N)$. Therefore, the equalities (14) and (15) can be rewritten as follows:

$$\begin{aligned}&\left[\Theta_1 \otimes I_N + \widetilde{K}_1 C_1\right]^{\mathrm{T}}(M_1 \otimes M_2) + (M_1 \otimes M_2)\left[\Theta_1 \otimes I_N + \widetilde{K}_1 C_1\right] \\ &\quad = -Q_1 \otimes M_2, \\ &\left[I_N \otimes \Theta_2 + \widetilde{K}_2 C_2\right]^{\mathrm{T}}(M_1 \otimes M_2) + (M_1 \otimes M_2)\left[I_N \otimes \Theta_2 + \widetilde{K}_2 C_2\right] \\ &\quad = -M_1 \otimes Q_2.\end{aligned} \tag{16}$$

Thus, Lemma 1 is completely proved.

Assumption 2. For subsystem (2), in which the coupling matrix $G(x)$ satisfies that $G(x) = M_1^{-1}\Upsilon^{\mathrm{T}}\Psi(x)M_2^{-1}$, where $\Psi(x) = (\psi_{ij})_{N_1 \times N}$ and $\psi_{ij} = x_i^{\mathrm{T}} x_j$.

If Assumption 2 holds, then we can get that $\|\Psi(x)\| = \sqrt{\sum_{i=1}^{N_1}\sum_{j=1}^{N}(x_i^{\mathrm{T}}x_j)^2} \le \sqrt{\sum_{i=1}^{N_1}\sum_{j=1}^{N}(\|x_i\| \cdot \|x_j\|)^2} \le \sqrt{\sum_{i=1}^{N}\|x_i\|^2 \sum_{j=1}^{N}\|x_j\|^2} = \|x\|^2$. Meanwhile, we note that Λ is a bounded and closed set in R^{Nn}, and $x \in \Lambda$. Thus, there exists a positive constant L to satisfy the inequality $\|x\|^2 \le L$.

General speaking, L is unknown. However, we can use the adaptive method to estimate it. In this paper, we use $\widehat{L} = \widehat{L}(t)$ to denote the estimated value of L. Hence, the estimation error is denoted as $\widetilde{L} = \widehat{L} - L$.

3. Main Results

Definition 3. Designing a matrix differential system $\dot{\widehat{P}} = F(\widehat{P}, Y_1, Y_2, \widehat{L})$, if the state $\widehat{P}$ satisfies $\lim_{t \longrightarrow +\infty}(P - \widehat{P}) = 0$, then the matrix differential system $\dot{\widehat{P}} = F(\widehat{P}, Y_1, Y_2, \widehat{L})$ can be regarded as a state observer of the links subsystem (2).

If Assumptions 1 and 2 hold, the state observer of the links subsystem (2) can be designed and presented as follows:

$$\begin{aligned}\dot{\widehat{P}} = &\left(\Theta_1 + K_1\Upsilon\right)\widehat{P} + \widehat{P}\left(\Theta_2 + K_2\Upsilon\right)^{\mathrm{T}} \\ &+ \Gamma\left(\widehat{P}, Y_1, Y_2, \widehat{L}\right) - K_1 Y_1 - Y_2^{\mathrm{T}} K_2^{\mathrm{T}},\end{aligned} \tag{17}$$

with the following adaptive law

$$\dot{\widehat{L}} = \frac{1}{\rho}\left\|\mathrm{vec}(Y_1) - C_1\mathrm{vec}(\widehat{P})\right\|, \tag{18}$$

where $\widehat{P}$ denotes the estimated value of the state P in (2); the robust term $\Gamma(\widehat{P}, Y_1, Y_2, \widehat{L}) = \begin{cases}\Omega, \Upsilon\widehat{P} \ne Y_1 \\ 0, \Upsilon\widehat{P} = Y_1\end{cases}$, where $\Omega = \widehat{L}((M_1^{-1}\Upsilon^{\mathrm{T}}(Y_1 - \Upsilon\widehat{P})M_2^{-1})/(\|Y_1 - \Upsilon\widehat{P}\|))$, ρ is a given positive constant, and the matrices K_1, K_2, M_1, and M_2 can be obtained by solving the Lyapunov equations (6) and (7), respectively.

According to (3) and (4), we can deduce from (17) that

$$\begin{aligned}\mathrm{vec}(\dot{\widehat{P}}) = &\left(A + \widetilde{K}_1 C_1 + \widetilde{K}_2 C_2\right)\mathrm{vec}(\widehat{P}) \\ &+ \mathrm{vec}\left(\Gamma\left(\widehat{P}, Y_1, Y_2, \widehat{L}\right)\right) - \widetilde{K}_1\mathrm{vec}(Y_1) \\ &- \widetilde{K}_2\mathrm{vec}\left(Y_2^{\mathrm{T}}\right).\end{aligned} \tag{19}$$

Clearly, $\|Y_1 - \Upsilon\widehat{P}\| = \|Y_1^{\mathrm{T}} - \widehat{P}^{\mathrm{T}}\Upsilon^{\mathrm{T}}\| = \|\mathrm{vec}(Y_1 - \Upsilon\widehat{P})\| = \|\mathrm{vec}(Y_1^{\mathrm{T}} - \widehat{P}^{\mathrm{T}}\Upsilon^{\mathrm{T}})\|$; thus, we get $\mathrm{vec}(\Omega) = \widehat{L}((\widetilde{M}^{-1}C_1^{\mathrm{T}}[\mathrm{vec}(Y_1) - C_1\mathrm{vec}(\widehat{P})])/(\|\mathrm{vec}(Y_1) - C_1\mathrm{vec}(\widehat{P})\|))$.

In this paper, the estimation error is denoted by $E = P - \widehat{P}$. By using (3), (4), and properties about Kronecker product and $\mathrm{vec}(\cdot)$ operator, we can get the following error system:

$$\begin{aligned}\mathrm{vec}(\dot{E}) = &\left(A + \widetilde{K}_1 C_1 + \widetilde{K}_2 C_2\right)\mathrm{vec}(E) + \widetilde{M}^{-1}C_1^{\mathrm{T}}\mathrm{vec}(\Psi(x)) \\ &- \mathrm{vec}\left(\Gamma\left(\widehat{P}, Y_1, Y_2, \widehat{L}\right)\right).\end{aligned} \tag{20}$$

Theorem 1. *If Assumptions 1 and 2 are true, then the matrix differential system (17) with the parameter adaptive law (18) is the state observer of the links subsystem (2).*

Proof. Consider the following Lyapunov function:

$$V = \frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\widetilde{M}\mathrm{vec}(E) + \frac{1}{2}\rho\widetilde{L}^2. \tag{21}$$

Calculating the orbit derivative of V along (20) gives that

$$\begin{aligned}\dot{V} &= \mathrm{vec}(E)^{\mathrm{T}}\widetilde{M}\mathrm{vec}(\dot{E}) + \rho\widetilde{L}\dot{\widehat{L}} \\ &= \mathrm{vec}(E)^{\mathrm{T}}\widetilde{M}\left\{\left(A + \widetilde{K}_1 C_1 + \widetilde{K}_2 C_2\right)\mathrm{vec}(E) + \widetilde{M}^{-1}C_1^{\mathrm{T}}\mathrm{vec}(\Psi(x)) - \mathrm{vec}\left(\Gamma\left(\widehat{P}, Y_1, Y_2, \widehat{L}\right)\right)\right\} + \rho\widetilde{L}\dot{\widehat{L}} \\ &= \mathrm{vec}(E)^{\mathrm{T}}\widetilde{M}\left(\Theta_1 \otimes I_N + \widetilde{K}_1 C_1\right)\mathrm{vec}(E) + \mathrm{vec}(E)^{\mathrm{T}}\widetilde{M}\left(I_N \otimes \Theta_2 + \widetilde{K}_2 C_2\right)\mathrm{vec}(E) \\ &\quad + \mathrm{vec}(E)^{\mathrm{T}}C_1^{\mathrm{T}}\mathrm{vec}(\Psi(x)) + \rho\widetilde{L}\dot{\widehat{L}} - \begin{cases}\widehat{L}\dfrac{\mathrm{vec}(E)^{\mathrm{T}}C_1^{\mathrm{T}}[C_1\mathrm{vec}(E)]}{\left\|C_1\mathrm{vec}(E)\right\|}, C_1\mathrm{vec}(\widehat{P}) \ne \mathrm{vec}(Y_1) \\ 0, C_1\mathrm{vec}(\widehat{P}) = \mathrm{vec}(Y_1)\end{cases}\end{aligned}$$

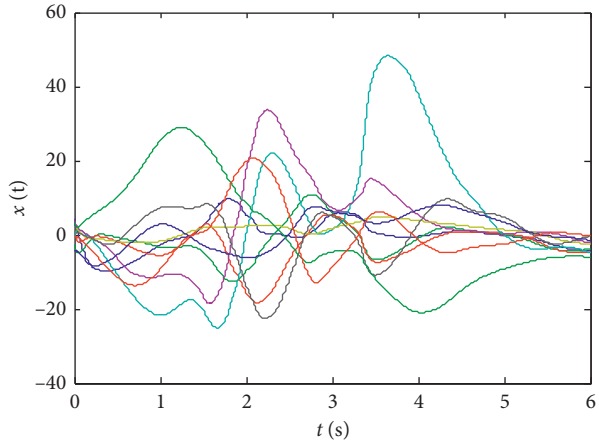


Figure 1: State trajectories of subsystem (1).

$$
\begin{aligned}
&\le \frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\left[(\Theta_1 \otimes I_N + \tilde{K}_1 C_1)^{\mathrm{T}}\tilde{M} + \tilde{M}(\Theta_1 \otimes I_N + \tilde{K}_1 C_1)\right]\mathrm{vec}(E) \\
&\quad + \frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\left[(I_N \otimes \Theta_2 + \tilde{K}_2 C_2)^{\mathrm{T}}\tilde{M} + \tilde{M}(I_N \otimes \Theta_2 + \tilde{K}_2 C_2)\right]\mathrm{vec}(E) \\
&\quad + \left\|\mathrm{vec}(E)^{\mathrm{T}} C_1^{\mathrm{T}}\right\|\left\|\mathrm{vec}(\Psi(x))\right\| + \rho\tilde{L}\dot{\hat{L}} - \begin{cases} \hat{L}\left\|\mathrm{vec}(E)^{\mathrm{T}} C_1^{\mathrm{T}}\right\|, C_1\mathrm{vec}(\hat{P}) \neq \mathrm{vec}(Y_1) \\ 0, C_1\mathrm{vec}(\hat{P}) = \mathrm{vec}(Y_1) \end{cases} \\
&\le -\frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\left(\tilde{Q}_1 + \tilde{Q}_2\right)\mathrm{vec}(E) + L\left\|\mathrm{vec}(E)^{\mathrm{T}} C_1^{\mathrm{T}}\right\| + \rho\tilde{L}\dot{\hat{L}} \\
&\quad - \begin{cases} \hat{L}\left\|\mathrm{vec}(E)^{\mathrm{T}} C_1^{\mathrm{T}}\right\|, C_1\mathrm{vec}(\hat{P}) \neq \mathrm{vec}(Y_1) \\ 0, C_1\mathrm{vec}(\hat{P}) = \mathrm{vec}(Y_1) \end{cases} \\
&= -\frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\left(\tilde{Q}_1 + \tilde{Q}_2\right)\mathrm{vec}(E) + \rho\tilde{L}\dot{\hat{L}} + \hat{L}\left\|\mathrm{vec}(E)^{\mathrm{T}} C_1^{\mathrm{T}}\right\| - \tilde{L}\left\|\mathrm{vec}(Y_1) - C_1\mathrm{vec}(\hat{P})\right\| \\
&\quad - \begin{cases} \hat{L}\left\|\mathrm{vec}(E)^{\mathrm{T}} C_1^{\mathrm{T}}\right\|, C_1\mathrm{vec}(\hat{P}) \neq \mathrm{vec}(Y_1) \\ 0, C_1\mathrm{vec}(\hat{P}) = \mathrm{vec}(Y_1) \end{cases} \\
&= -\frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\left(\tilde{Q}_1 + \tilde{Q}_2\right)\mathrm{vec}(E) + \tilde{L}\left(\rho\dot{\hat{L}} - \left\|\mathrm{vec}(Y_1) - C_1\mathrm{vec}(\hat{P})\right\|\right) \\
&= -\frac{1}{2}\mathrm{vec}(E)^{\mathrm{T}}\left(\tilde{Q}_1 + \tilde{Q}_2\right)\mathrm{vec}(E). \qquad (22)
\end{aligned}
$$

From inequality (22), we can obtain that the estimation error matrix E is bounded and $E \xrightarrow{t \to +\infty} 0$. Thus, Theorem 1 is completely proved.

4. Simulation Example

In this paper, we consider a continuous analog Hopfield network with 10 neurons ($N = 10$) [23, 28], which is composed of nodes subsystem and links subsystem, where the nodes subsystem is described as follows:

$$
\dot{x}_i = A_i x_i + B_i f_i(x_i) + c_i \sum_{j=1}^{10} p_{ij} H_j(x_j), \quad i = 1, 2, \ldots, 10, \qquad (23)
$$

where $A_i = B_i = -i$, $f_i(x_i) = -5\cos t$, $c_i = i$, and $H_j(x_j) = (1 - e^{-x_j})/(1 + e^{-x_j})$.

Meanwhile, we assume that the changes in the links' weights $p_{ij}(t)$ satisfy the Riccati differential equation (2). If we choose $N_1 = 5$ and $\rho = 100$ and randomly select matrices $\Theta_1 \in R^{10\times 10}$, $\Theta_2 \in R^{10\times 10}$, and $\Upsilon \in R^{5\times 10}$ satisfying

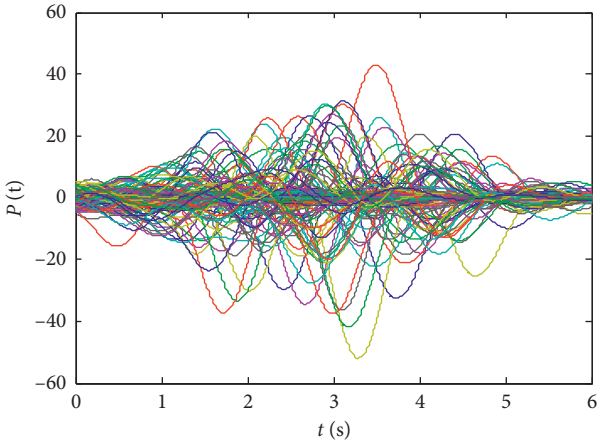


Figure 2: State trajectories of subsystem (2).

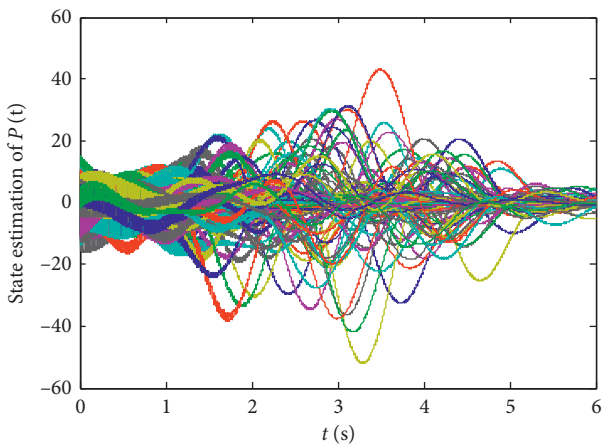


Figure 3: State trajectories of state observer (17).

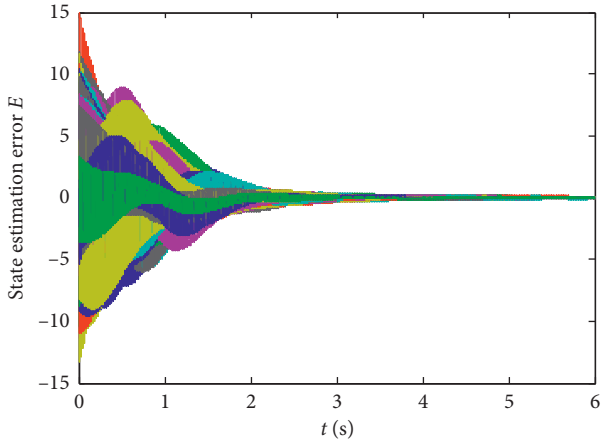


Figure 4: State trajectories of estimation error system (20).

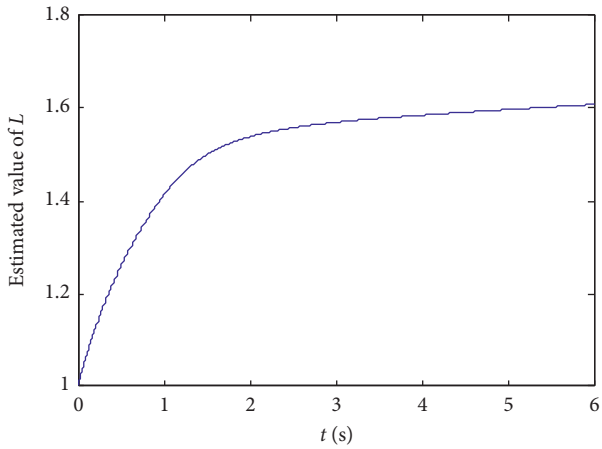


Figure 5: State trajectories of parameter adaptive system (18).

Assumption 1, then the matrices K_1, M_1 and K_2, M_2 can be obtained by solving the Lyapunov equations (6) and (7), respectively. Thus, we can get the coupling matrix $G(x) = M_1^{-1}\Upsilon^{\mathrm{T}}\Psi(x)M_2^{-1}$ in (2) satisfying Assumption 2.

Finally, randomly select the initial values of states $x_i(0), \widehat{L}(0)$, and $p_{ij}(0)$, $i, j = 1, 2, \ldots, 10$ in the range $(-5, 5)$, and the numerical results are shown in Figures 1–5:

(i) From Figures 2–4, we can see that the estimation error converges asymptotically to zero. According to Definition 3, we know that the Riccati dynamical equation (17) with the adaptive law (18) is a state observer of the subsystem (2), and the state observer is effective.

(ii) Compared to the results in [23], our advantage is that the result about the state observer of the subsystem (2) is true whatever the network is directed or undirected. Meanwhile, it is worth noting that, due to the effect of the parameter adaptive law (18), the state observer (17) does not contain the states of the nodes. This shows that the state observer is less affected by the dynamic changes in the nodes and thus improves the robustness of the state observer.

5. Conclusions

In this paper, a complete model of CDNs is proposed, which is composed of two coupled subsystems, called nodes subsystem and links subsystem, respectively. Contrary to the existing results on the state estimation problem of nodes subsystem, we mainly focus on the state estimation of the links subsystem with outputs and have designed a state observer with the parameter adaptive law to estimate the state of the links subsystem in this paper. In particular, this method solves the estimation problem of dynamic links in directed networks for the first time. Meanwhile, it implies that we can use the state estimation information of the links to directly design a controller for the links subsystem; thus, some control problems may be solved effectively. Therefore, the design method of state observer for dynamic links proposed in this paper can enrich the achievements about the state estimation of CDNs.

Data Availability

In this paper, we submitted data mainly related to theoretical proof and numerical simulation, in which the part of numerical simulation is realized by Matlab software; if

necessary, we can provide simulation source program and relevant data at any time.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Key Laboratory of Chongqing Municipal Institutions of Higher Education ([2017]3), the Program of Chongqing Development and Reform Commission (2017[1007]), the Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJ1710244, KJQN201801215, KJQN201801209, KJQN201901236, and KJ1710241), and the Scientific Research Project of Chongqing Three Gorges University (19QN07).

References

[1] C. L. P. Chen, J. Wang, C.-H. Wang, and L. Chen, "A new learning algorithm for a fully connected neuro-fuzzy inference system," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1741–1757, 2014.

[2] H. Kim, H. Lee, M. Ahn, H.-B. Kong, and I. Lee, "Joint subcarrier and power allocation methods in full duplex wireless powered communication networks for OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4745–4753, 2016.

[3] R. Zhang, D. Zeng, S. Zhong, and Y. Yu, "Event-triggered sampling control for stability and stabilization of memristive neural networks with communication delays," *Applied Mathematics and Computation*, vol. 310, pp. 57–74, 2017.

[4] L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility maximization resource allocation in wireless networks: methods and algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1018–1034, 2015.

[5] Z. Gao and Y. Wang, "The structural balance analysis of complex dynamical networks based on nodes' dynamical couplings," *PLoS One*, vol. 13, no. 1, Article ID e0191941, 2018.

[6] Z. Gao, Y. Wang, and L. Zhang, "Adaptive control of structural balance for complex dynamical networks based on dynamic coupling of nodes," *International Journal of Modern Physics B*, vol. 32, no. 4, Article ID 1850042, 2018.

[7] L.-Z. Liu, Y.-H. Wang, and Z.-L. Gao, "Tracking control for the connection relationships of discrete-time complex dynamical network associated with the controlled nodes," *International Journal of Control, Automation and Systems*, vol. 17, no. 9, pp. 2252–2260, 2019.

[8] Y. Wang, Y. Fan, Q. Wang, and Y. Zhang, "Stabilization and synchronization of complex dynamical networks with different dynamics of nodes via decentralized controllers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 8, pp. 1786–1795, 2012.

[9] X. Yang, J. Lam, D. W. C. Ho, and Z. Feng, "Fixed-time synchronization of complex networks with impulsive effects via nonchattering control," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5511–5521, 2017.

[10] W.-J. Yuan, X.-S. Luo, P.-Q. Jiang, B.-H. Wang, and J.-Q. Fang, "Stability of a complex dynamical network model," *Physica A: Statistical Mechanics and Its Applications*, vol. 374, no. 1, pp. 478–482, 2007.

[11] X. Liu, J. H. Park, N. Jiang, and J. Cao, "Nonsmooth finite-time stabilization of neural networks with discontinuous activations," *Neural Networks*, vol. 52, pp. 25–32, 2014.

[12] J. Ma, H. Ji, D. Sun, and G. Feng, "An approach to quantized consensus of continuous-time linear multi-agent systems," *Automatica*, vol. 91, pp. 98–104, 2018.

[13] H.-X. Hu, G. Wen, W. Yu, Q. Xuan, and G. Chen, "Swarming behavior of multiple Euler-Lagrange systems with cooperation-competition interactions: an auxiliary system approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5726–5737, 2018.

[14] X. Wu, G.-P. Jiang, and X. Wang, "State estimation for general complex dynamical networks with packet loss," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 11, pp. 1753–1757, 2018.

[15] L. Zou, Z. Wang, H. Gao, and X. Liu, "State estimation for discrete-time dynamical networks with time-varying delays and stochastic disturbances under the round-robin protocol," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1139–1151, 2017.

[16] H. Li, Z. Ning, Y. Yin, and Y. Tang, "Synchronization and state estimation for singular complex dynamical networks with time-varying delays," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 194–208, 2013.

[17] J. Hu, Z. Wang, S. Liu, and H. Gao, "A variance-constrained approach to recursive state estimation for time-varying complex networks with missing measurements," *Automatica*, vol. 64, pp. 155–162, 2016.

[18] C.-X. Fan, F. Yang, and Y. Zhou, "State estimation for coupled output discrete-time complex network with stochastic measurements and different inner coupling matrices," *International Journal of Control, Automation and Systems*, vol. 10, no. 3, pp. 498–505, 2012.

[19] R. Sakthivel, M. Sathishkumar, B. Kaviarasan, and S. Marshal Anthoni, "Synchronization and state estimation for stochastic complex networks with uncertain inner coupling," *Neurocomputing*, vol. 238, pp. 44–55, 2017.

[20] W. Li, J. Sun, Y. Jia, J. Du, and X. Fu, "Variance-constrained state estimation for nonlinear complex networks with uncertain coupling strength," *Digital Signal Processing*, vol. 67, pp. 107–115, 2017.

[21] C. Ma, J. Cao, L. Yang, J. Ma, and Y. He, "Effective social relationship measurement based on user trajectory analysis," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 1, pp. 39–50, 2014.

[22] F. Zeng, N. Zhao, and W. Li, "Effective social relationship measurement and cluster based routing in mobile opportunistic networks," *Sensors*, vol. 17, no. 5, p. 1109, 2017.

[23] Z.-L. Gao, Y.-H. Wang, J. Xiong, L.-L. Zhang, and W.-L. Wang, "Robust state observer design for dynamic connection relationships in complex dynamical networks," *International Journal of Control, Automation and Systems*, vol. 17, no. 2, pp. 336–344, 2019.

[24] M. Bartos, I. Vida, and P. Jonas, "Synaptic mechanisms of synchronized gamma oscillations in inhibitory interneuron networks," *Nature Reviews Neuroscience*, vol. 8, no. 1, pp. 45–56, 2007.

[25] P. R. Pagilla, N. B. Siraskar, and R. V. Dwivedula, "Decentralized control of web processing lines," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 1, pp. 106–117, 2007.

[26] S. A. Marvel, J. Kleinberg, R. D. Kleinberg, and S. H. Strogatz, "Continuous-time model of structural balance," *Proceedings of the National Academy of Sciences*, vol. 108, no. 5, pp. 1771–1776, 2011.

[27] D. Bahuguna, A. Ujlayan, and D. N. Pandey, "Advanced type coupled matrix Riccati differential equation systems with Kronecker product," *Applied Mathematics and Computation*, vol. 194, no. 1, pp. 46–53, 2007.

[28] J. Hopfield and D. Tank, "Computing with neural circuits: a model," *Science*, vol. 233, no. 4764, pp. 625–633, 1986.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8816125, 11 pages
https://doi.org/10.1155/2020/8816125

Research Article

An Improved Sign Language Translation Model with Explainable Adaptations for Processing Long Sign Sentences

Jiangbin Zheng,[1] **Zheng Zhao**,[2] **Min Chen**,[2] **Jing Chen**,[2] **Chong Wu**,[3] **Yidong Chen**,[1] **Xiaodong Shi**,[1] **and Yiqi Tong**[1]

[1]*Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, China*
[2]*China Mobile (Suzhou) Software Technology Co., LTD, Suzhou 215000, China*
[3]*Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong*

Correspondence should be addressed to Yidong Chen; ydchen@xmu.edu.cn

Received 22 May 2020; Revised 20 June 2020; Accepted 5 September 2020; Published 24 October 2020

Academic Editor: Nian Zhang

Sign language translation (SLT) is an important application to bridge the communication gap between deaf and hearing people. In recent years, the research on the SLT based on neural translation frameworks has attracted wide attention. Despite the progress, current SLT research is still in the initial stage. In fact, current systems perform poorly in processing long sign sentences, which often involve long-distance dependencies and require large resource consumption. To tackle this problem, we propose two explainable adaptations to the traditional neural SLT models using optimized tokenization-related modules. First, we introduce a *frame stream density compression* (FSDC) algorithm for detecting and reducing the redundant similar frames, which effectively shortens the long sign sentences without losing information. Then, we replace the traditional encoder in a *neural machine translation* (NMT) module with an improved architecture, which incorporates a *temporal convolution* (T-Conv) unit and a *dynamic hierarchical bidirectional GRU* (DH-BiGRU) unit sequentially. The improved component takes the temporal tokenization information into consideration to extract deeper information with reasonable resource consumption. Our experiments on the *RWTH-PHOENIX-Weather 2014T* dataset show that the proposed model outperforms the state-of-the-art baseline up to about 1.5+ BLEU-4 score gains.

1. Introduction

Sign languages are visual-based natural languages used by the deaf people for their communication. Since most hearing people cannot understand sign language, *sign language translation* (SLT) has become an important application to bridge the communication gap between deaf and hearing people. In recent years, researchers have successively proposed deep learning models for neural SLT (e.g., [1–6]).

The existing SLT models basically follow a multimodal architecture, where *convolutional neural network* (CNN) and *neural machine translation* (NMT) are sequentially connected. The CNN module is used to extract image-level features, reduce the fine-grained input, and generate a tokenization layer as the input to the NMT module; the NMT module is the main translation module for encoding and decoding to generate target sentences. The above basic SLT architecture was first proposed by Camgoz et al. [1]. The tokenization layer serves as a hub layer in this architecture. Hence, optimizing it can improve the performance of both CNN and NMT.

However, most of the current SLT works only improve the CNN or NMT module separately, resulting in poor connection between the two modules which causes two serious problems:

(1) Poor interpretability: most of the improvements focus on some common tricks, rather than considering the uniqueness of SLT. The characteristics of SLT determine that it is a special NMT task, although

the input form is different from conventional spoken language. Therefore, analyzing from the input form may help us to find some interesting SLT phenomena and get a better interpretability. For a spoken sentence, the input is usually a series of words. Although there are semantic connections between words, they are expressed in a discrete form. As for a sign sentence, the input is usually a video signal. In actual application, the video needs to be framed into continuous frame images. Intuitively, we can compare each video frame to the basic word element of sign language. Unlike spoken language, the video frames of any sign sentence are continuous, and the order is closely related. In other words, it is illegal to reverse the order between any frames. Specifically, we found that there are many similar frames in the neighborhood, and these frames repeatedly express some meanings, which will cause redundant information and long sentence. However, no works use this visual phenomenon to custom optimization algorithms for sign language.

(2) Poor performance for long sentences: longer sentences result in long-distance dependencies, large resource consumption, and low evaluation scores. This shows that both CNN and NMT modules need to be improved. However, the visual CNN module is obtained more attention, and the work of the innovative NMT module is obtained less attention. Besides, the improvement from the perspective of model interpretation is also a very important aspect.

Longer sentences mean more frames. The longer the sentence is, the more complicated the relationship between video frames will have, which leads to insufficient connection between frames. In theory, the amount of calculation may increase exponentially. Hence, the SLT model generally specifies a maximum number of input frames for the CNN module. For longer sentences, how to express more effective information within a certain window size is a meaningful research point. However, there is no work considering reducing useless frames from understandable visual features. Especially for longer sentences, CNN is more pressured and less efficient. If we can reduce the number of sign language frames according to the visual surface image features, then we may still get the same sentence meaning with a fewer frames (like turning long sentences into short sentences), which can not only reduce the convolution pressure, but also generate a higher quality tokenization layer. Moreover, the tokenization layer is then input into the NMT module, so optimizing it in the tokenization level will be a key role for improving the subsequent NMT.

To solve the above mentioned issues, we propose a novel SLT model with a better interpretability for longer sentences, as shown in Figure 1. There are two improvements with tokenization-related units.

First, we propose a frame-level *frame stream density compression* (FSDC) algorithm, which can compare pixels at the image level in an unsupervised manner, reducing redundant frames in temporal neighborhood. Intuitively, it can be understood as retaining high-density information by comparing the similarity of input image frames in the neighborhood. The reduced convolution information can generate tokenization with a smaller size, which allows more information to be transmitted within the limited window length. Besides, for the NMT module, reducing the number of input frames means a shorter length of input. Overall, this is a visually interpretable optimization of sign language that converts long sentences into short sentences.

Second, we replace the traditional encoder in the NMT module with an improved architecture to further strengthen the association between long sentence video frames. Inspired by the study of FairSeq [7], a hybrid model is proposed. The model incorporates a *temporal convolution* (T-Conv) unit and a *dynamic hierarchical bidirectional GRU* (DH-BiGRU) unit sequentially. It first convolves the input in the time domain and then encodes the semantic information in the subsequent deep hierarchical RNNs. We can still treat the tokenization layer as a vector representation layer of the dimensionality-reduced frames. As an improvement, 3DCNN/C3D was used in the CNN module [8, 9] to strengthen the association between frames in the time domain. However, it requires larger resource consumption and does not always work well in the case of low sign language resources. We observed that, if the NMT module convolves the sign sentences at the tokenized level in the time domain using 2DCNN, it can not only approach the function of 3DCNN/C3D, but also approach the speed of 2DCNN. All in all, this also shortens long sentences in the time domain and deepens the RNN structure in a hierarchical way. In this case, the NMT structure can handle longer sentences as easily as short sentences.

The main contributions of this paper are as follows:

(1) We have proposed a novel SLT model with tokenization-related units, which can better handle longer sentences in lower resource consumption, and has a better interpretability.

(2) We have introduced for the first time an unsupervised FSDC algorithm to compress the density of the input frames without removing key information. This method is suitable for many similar video tasks.

(3) We have proposed a novel NMT module for SLT with optimized encoder-related units, *temporal convolution and dynamic hierarchical bidirectional GRU hybrid network* (TC-DHBG-Net), which compresses the effective information of the tokenization layer from the time domain so that long sentences are further shortened on the time domain to facilitate hierarchical GRUs to find semantic information.

(4) Moreover, our improved neural SLT model has been made publicly available (https://github.com/binbinjiang/nslt_xmu).

2. Our Proposed Approaches

As a special language, sign language has its own specific linguistic rules as well [10], so the SLT model follows the

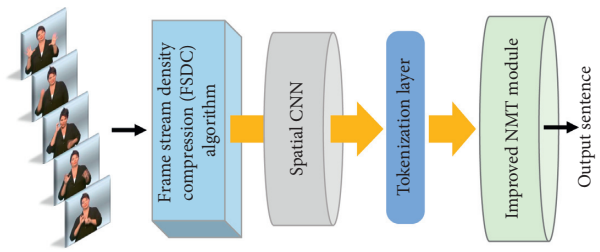


Figure 1: Overview of our proposed end-to-end SLT model with improved tokenization-related units, which includes an *FSDC* optimization algorithm and an improved NMT module.

NMT framework, as shown in Figure 1. Now suppose that $\mathbf{y} = (y_1, y_2, \ldots, y_{T_y})$ is an output sentence that corresponds to the sign video frame sequence $\mathbf{x} = (x_1, x_2, \ldots, x_{T_x})$ in the training set. At the very beginning, we use the unsupervised *FSDC* algorithm module to optimize the frame-level input sentences. Then, a spatial CNN is used to convolute frames to gain tokenization layer which is then input into the NMT module for encoding and decoding. In this section, we will introduce the proposed approaches in detail.

2.1. Unsupervised FSDC Module. As shown in Figure 2(a), the spatial CNN is mainly used to reduce the fine-grained input of video frames. In SLT, the video frame is the most basic input unit. The compression of video frames directly affects the processing efficiency of CNN and the quality of the tokenization layer. Therefore, optimizing the number of frames also means optimizing the tokenization layer.

For any video dataset, we must follow a fixed *frames per second* (FPS) to frame all the videos, which leads to massive similar redundant frames in the temporal neighborhood. As an illustration, a signer signs the same sign language at fast and slow speeds, respectively. Although the two express the same meaning, they produce videos of different lengths. Obviously, a video signed at a slower speed will get more redundant similar frames in temporal neighborhood.

To reduce this effect, the *FSDC* algorithm is proposed. We delete the less-important frames by comparing the similarity index and to keep the sequence of the frames fixed at the same time. In theory, it helps us to reduce the amount of training data as well as errors caused on account of sign speed and *FPS*.

We use the SSIM algorithm [11] to calculate the similarity between two images, which is close to the intuitive feeling of the human eye. When calculating the structural similarity of frame f_i and frame f_j, the corresponding calculation flow chart is shown in Figure 3. The formula of the SSIM algorithm is as follows:

$$\mathrm{SSIM}(f_i, f_j) = \left[L(f_i, f_j)\right]^x \cdot \left[C(f_i, f_j)\right]^y \cdot \left[S(f_i, f_j)\right]^z, \quad (1)$$

where $L(^*)$ denotes the luminance comparison, $C(*)$ denotes the contrast comparison, and $S(*)$ denotes the structure comparison. Note that $x > 0, y > 0,$ and $z > 0$, we initialize $x = y = z = 1$. SSIM$(*)$ is a decimal between 0 and 1. Extremely, SSIM = 1 means two images are completely identical, while SSIM = 0 means completely different.

The *FSDC* calculates the SSIM indexes for both each frame and all frames in the neighborhood. If the SSIM index is greater than a certain threshold $\delta\,(0 < \delta < 1)$, only one of them will be retained, while the rest will be discarded as redundant frames. A running example of Algorithm 1 is shown in Figure 2(b).

Formally, we explore frame-level input tokenization as shown in Figure 2(a) and map the feature vectors to the tokenization layer as

$$\Gamma = \mathrm{SpatialCNN}\,(FSDC\,(x)). \quad (2)$$

2.2. TC-DHBG-Net for Encoding Stage. Figure 4 shows the improved NMT module we proposed. Specifically, we improve the encoder in two folds. The first is *T-Conv* unit for the tokenization layer; and the second is *DH-BiGRUs* for mining semantic information.

The T-Conv unit is inspirited by the work of Bérard et al. [12] on the end-end speech task. It takes as input a sequence of features for tokenization layer. These features are given as input to two nonlinear (tanh) layers, which output new features of size n. In order to enhance the optical flow feature capture, we concatenate the positional encoding [13] to obtain the feature vectors with position information. Like [14], this new set of features is then passed to a stack of two convolutional layers. Each layer applies 16 convolution filters of shape (3, 3, depth) with a stride of (2, 2) w.r.t. time and feature dimensions; depth is 1 for the first layer and 16 for the second layer. We get features of shape (T_x/2, n/2, 16) after the 1st layer and (T_x/4, n/4, 16) after the 2nd layer. This latter tensor is flattened with shape ($T_x = T_x$/4, 4n) before being passed to a stack of three-level *DH-BiGRUs*. This set of features has 1/4th the time length of the initial features, which speeds up the raining because the complexity of the model is quadratic with respect to the source length.

The DH-BiGRU unit computes a sequence of annotations $h = h_i, \ldots, h_{T_x}$, where each annotation h_i is a concatenation of the corresponding forward and backward states. The hidden state of the last GRU layer in each hierarchy is inserted into the next hierarchy. Formally, first we insert the tokenized vectors into a recurrent neural structure to obtain the semantic information of the context sequence. For recurrent unit type, we choose GRU [15] instead of LSTM [16] because the former has fewer gate structures. The hierarchical structure [2, 12, 17] and bidirectional structure can extract deeper relevant information. Suppose that the hierarchy of HGRU is n, then

$$\xi_{\mathrm{encoder}} = \varphi_{en\_rnn_n, en\_rnn_{n-1}, \ldots, en\_rnn_1}(\Gamma) = (h_1, h_2, \ldots, h_{n'}), \quad (3)$$

where $(h_1, h_2, \ldots, h_{n'})$ are the hidden states of the last GRU layer, and n' is a variable, and $\varphi_{en\_rnn}(^*)$ indicates the processing of RNN in the encoder.

2.3. Decoder and Attention Mechanism

2.3.1. Decoder. For the word embedding, we use a fully connected layer that learns a linear projection from one-hot

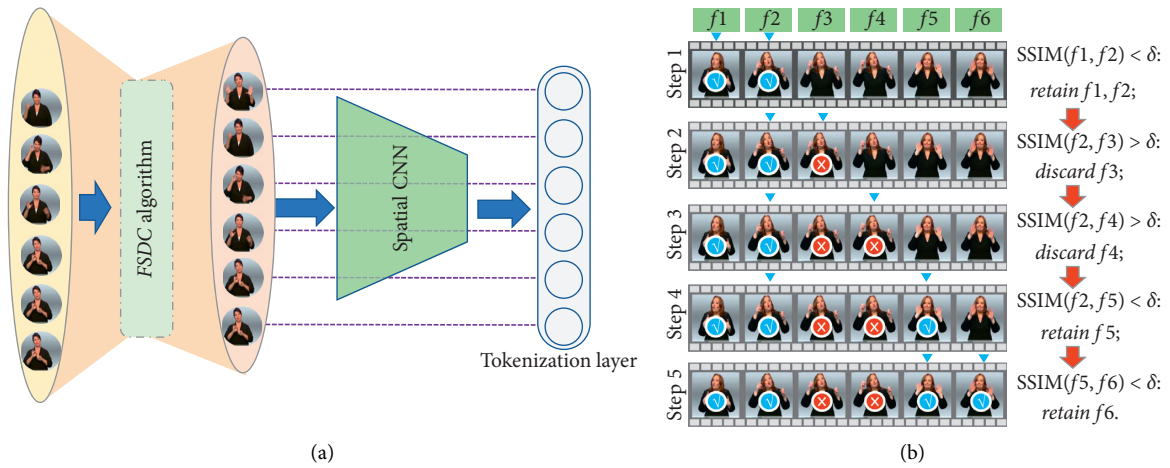


Figure 2: (a) The spatial CNN part with the proposed *FSDC* algorithm module. (b) Scaled *FSDC* module with a running example.

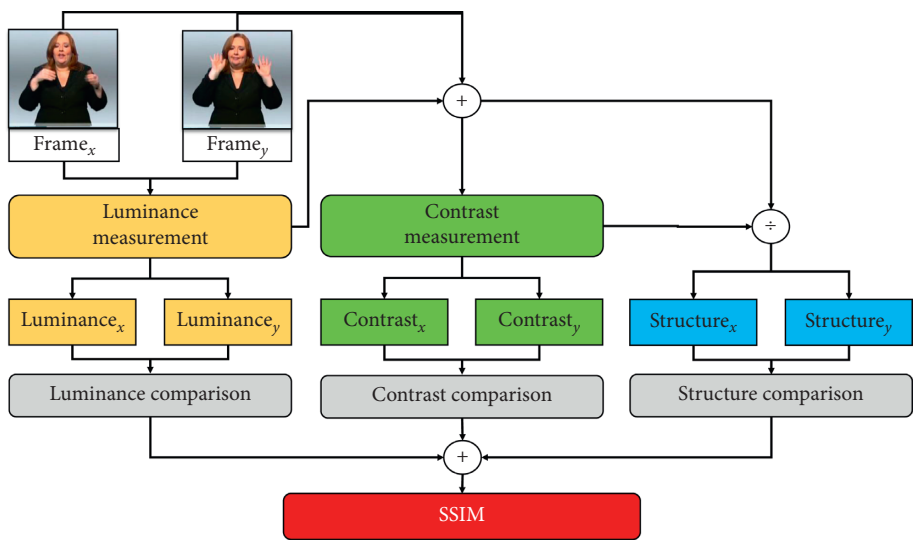


Figure 3: The process of comparing SSIM values between two images.

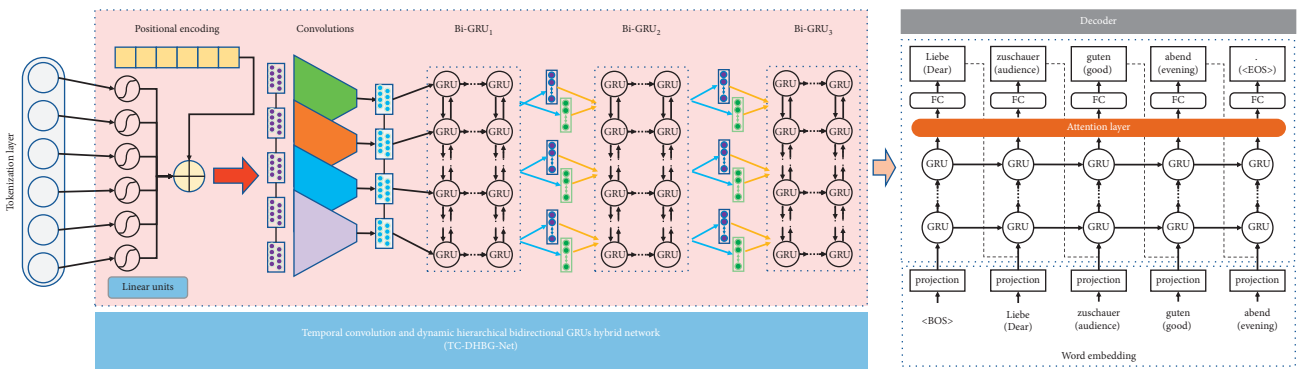


Figure 4: The improved encoder in the NMT module with a TC-DHBG-Net.

vectors of spoken language words to a denser space as follows:

$$\omega_i = \text{WordEmbedding}(y_i), \tag{4}$$

where ω_i is the embedded version of the spoken word y_i.

In the decoding stage, we aim at maximizing the probability $p(\mathbf{y} \mid \mathbf{x})$. The decoder computes a probability of the translation $\mathbf{y}$ by decomposing the joint probability into the ordered conditional probabilities as follows:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{T_y} p\left(y_i \mid y_1, y_2, \ldots, y_{i-1}, (h_1, h_2, \ldots, h_{n'})\right). \tag{5}$$

2.3.2. Attention Mechanism. Like other SLT models, we may also suffer from long-term dependencies, vanishing gradients, and performance deterioration with many input frames. To solve the issues, we utilize attention mechanisms which have been proved useful in various tasks including but not limited to machine translation. The most common attention mechanisms are the mechanisms of Bahdanau et al. [18] and Luong et al. [19]. Based on hyperparameter experiments, we take Bahdanau as our attention mechanism. Given the input $\mathbf{x}$, we define each conditional probability at time i depending on a dynamically computed context vector c_i as follows:

$$p\left(y_i \mid y_1, y_2, \ldots, y_{i-1}, x\right) = \text{softmax}\left(g\left(s_i\right)\right), \tag{6}$$

where s_i is the hidden state of the decoder at time i and g is a linear transformation that outputs a vocabulary-sized vector. Note that the hidden state s_i is computed as

$$s_i = \varphi_{de_{rnn}}(\omega_{i-1}, s_{i-1}, \omega_i), \tag{7}$$

where $\varphi_{de_{rnn}}(^*)$ indicates the processing of RNN in the decoder and ω_{i-1} is the word embedding of the previously predicted word y_{i-1}, s_{i-1} is the last hidden state of the decoder, and c_i is computed as a weighted sum of the hidden states from encoder as

$$c_i = \sum_{j=1}^{T_y} \alpha_{ij} h_j, \tag{8}$$

where α_{ij} is the weight of each annotation h_j.

3. Experiments

In this section, we conducted a series of experiments on the *RWTH-PHOENIX-Weather 2014T* dataset by employing our improved SLT model with tokenization-related units compared to the baseline.

3.1. Baseline. As described above, the baseline is an attention-based structure combined by 2DCNN and Seq2Seq sequentially. The spatial 2DCNN is an AlexNet [20], and its parameters are pretrained on Imagenet [21]. The encoder and decoder of Seq2Seq are nonhierarchical GRUs. In order to compare with the baseline fairly, all experiments run in the same dataset and GPU environment. Except for the differences mentioned in the paper, other configurations for all models are consistent by default.

3.2. Dataset. The *RWTH-PHOENIX-Weather 2014T* is the most popular continuous SLT dataset. It is collected by extending the German sign language recognition (SLR) dataset, *RWTH-PHOENIX-Weather 2014 Corpus* [22]. Compared with other SLT datasets, this dataset has larger data and higher quality. It contains 4,839 vocabulary, 8,257 video clips, 947,756 frames, and 113,717 words in total, as shown in Table 1. Each video corresponds to a translation sentence. Although the dataset includes sign language gloss corpus, our model is trained without gloss-level alignment, where the glosses give the meaning and the order of signs [1, 23, 24]. Nevertheless, the use of glosses is limited to a prerequisite that word label in sentences is consistent with the order of corresponding visual content. In the other words, if the word is out of order, it is unsuitable to tackle sequential frame-level classification under word labels in disorder. In fact, most datasets do not include gloss annotations. Although we do not consider it for this work, we conducted NMT experiments using gloss to gain optimal settings as [1].

3.3. Settings. Based on baseline conclusions and our experience, we preset some important hyperparameters. We use GRU as the recursive module for both encoder and decoder, where each recurrent layer contains 1,000 hidden units. During the training, the optimizer used is Adam [25], and the learning rate is 0.00001 with a decay factor of 0.98 and a batch size of 1. During the decoding, we use *beam search* with a width size of 3 to generate sentences.

3.4. Evaluation. We use BLEU [26] and ROUGE [27] as the evaluation metrics, which are most used in machine translation tasks. Note that the BLEU score is represented by BLEU-1, 2, 3, 4 and the ROUGE score refers to ROUGE-L F1-SCORE. In training, the BLEU-4 score on the development set is used to select the best model.

3.5. Comparison to Existing Approaches. Table 2 shows the performance comparison between our proposed systems and the existing baseline systems.

The existing baseline systems use different attention mechanisms, of which the Bahdanau mechanism performs best. It is worth mentioning that although the transformer has good performance in many NMT tasks, it does not achieve good results in the SLT dataset due to its small data size.

Our proposed systems contain innovations in multiple places, so we added different improved modules on the baseline for comparison. We can see that after using the unsupervised *FSDC* algorithm (#2h), the model achieves better performance. As for the improvement of the encoder in NMT module, either *T-Conv* or *DH-BiGRUs* units have a

Input: input F; threshold δ $(0 \le \delta \le 1)$; number of video frames N.
Output: F'
Initialize $x = 0$, $i = 1$
for $x + i \le N$, **do**
  if $\mathrm{SSIM}(f_x, f_{x+i}) > \delta$, **then**
    Retain $x + i$, discard f_{x+i}, update $i = i + 1$
  else if $\mathrm{SSIM}(f_x, f_{x+i}) \le \delta$, **then**
    Retain f_x, f_{x+i}, update $x = x + i, i = 1$
  end if
end for

Algorithm 1: *FSDC* algorithm for temporal neighborhood.

Table 1: Key statistics of the German datasets.

	Train	Dev	Test
Vocab.	2,887	951	1,001
Clips	7,096	519	642
Frames	827,354	55,775	64,627
Tot. words	99,081	6,820	7,816

promoting effect as shown in Table 2 (#2e and #2f). The complete improved encoder module which uses both *T-Conv* and *DH-BiGRUs* units (i.e., TC-DHBG-Net) improves more significantly as shown in Table 2 (#2g). From the performance, we can see that the improved encoder in the NMT module is the most important and the *FSDC* algorithm can slightly improve the basis as shown in Table 2 (#2i). Overall, the proposed tokenization-related units without extra information improve significantly for the SLT.

3.6. Validation on TC-DHBG-Net. In order to validate the role of the *T-Conv}* unit of the *TC-DHBG-Net*, we only add *T-Conv* units to the encoder of the baseline, while the recursive neural unit remains unchanged. In Table 2, #2e exceeds the baseline moderately, which proves the positive role of the *T-Conv* unit.

The *DH-BiGRUs* unit is another important component of the *TC-DHBG-Net*. We replace the original GRUs of the baseline with our *DH-BiGRUs* unit in 3 levels by default. As shown in Table 2 (#2f), the multilevel structure is introduced and the performance is moderately improved, proving the effect of the hierarchical structure.

Although *T-Conv* unit and *DH-BiGRUs* unit have been proved by the above experiments, it does not mean that the combination of the two will be better. Therefore, it is necessary to introduce Table 2 (#2g). Compared with baseline, #2g improves significantly, which is better than any single module (#2e or #2f).

3.7. Ablation on the Levels of DH-BiGRUs. The *DH-BiGRU* has an important hyperparameter, the number of RNN levels. To test the scores for different levels of *DH-BiGRU* in the recurrent neural unit, we set the number N_{level} to 1, 2, 3, and 4, respectively. We conducted experiments based on the previous experiment as shown in Table 2 (#2g). Table 3 illustrates that the hierarchical structure has a significant impact on the scores. When N_{level} is set to less than 3, the scores increase as the number of levels increases, and when $N_{\text{level}} = 3$, the score increases to peak; but when $N_{\text{level}} > 3$, the score starts to drop. As a conclusion, a larger number of layers do not mean a higher score. Therefore, we set $N_{\text{level}} = 3$ to the optimal hyperparameter.

3.8. Validation on FSDC Algorithm. At the very beginning, we analyze the structural similarity of all frames in the dataset. Figure 5(a) shows that the number or proportion of the separable redundant frames varies with different thresholds. Even if the threshold is set to 75%, we can see that the number of frames for temporal neighborhood exceeds 85%. Once the threshold is lower, the proportion of frames will be greater. This indicates that the relationship between the frames is tight. A reasonable initial threshold is crucial to the model, but the threshold is an empirical and experimental hyperparameter. If the threshold is set too low, much more useful frame information may loss; on the contrary, the optimization will not work at all. Analyzing Figure 5(a), we think that the similarity threshold is set to at least 94%.

To validate the *FSDC* algorithm, we set the thresholds from 94% to 99%, to control the percentage of redundant frames. We conducted the experiment on the baseline (Table 4 (#4a)) and the structure we proposed (Table 4 (#4b)), respectively. Figure 5(b) shows that within a reasonable range, the *FSDC* algorithm can be positive relative to the improvement of the baseline, especially when the threshold is set to 95%. But the relative value of negative numbers in Table 4 (#4b) also shows that not all thresholds can improve performance.

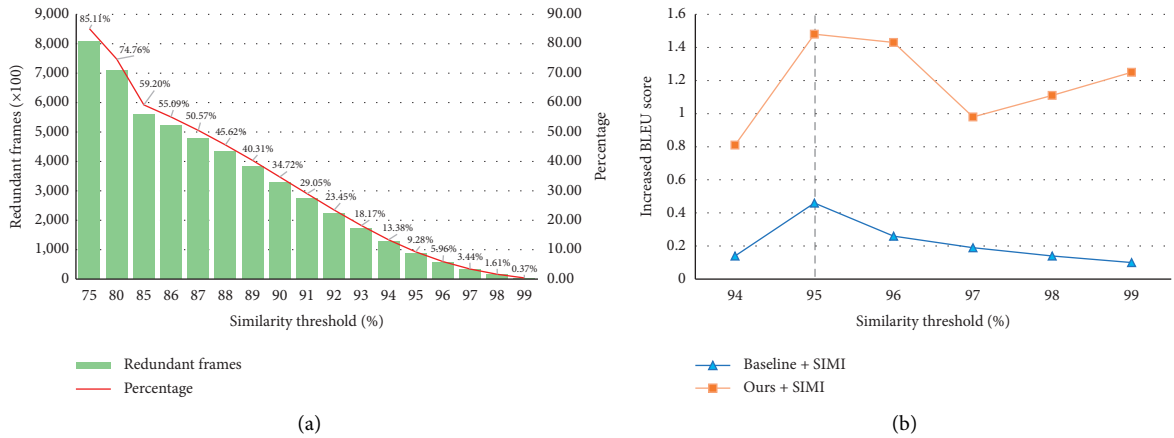
Moreover, it is worth mentioning that the size of the training data is reduced by 9.28% when the threshold is set to 95%. The optimized dataset not only saves storage space, but also saves processing time (about 10% reduction).

3.9. About Length. Figure 6(a) shows the distribution of the number of sentences with respect to the different lengths of source sentences (frames) on the test set. Since the frame number of most sentences is less than 100, we think that more than 100 frames are considered as long sentences.

Table 2: Experiments on the existing baseline systems vs. variants of our novel model.

#	Model	Development set					Test set				
		ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Existing baseline systems											
2a	None	29.54	28.33	15.71	10.32	8.57	28.60	26.65	15.02	10.27	8.24
2b	Transformer	30.28	29.82	16.98	11.89	8.93	29.89	29.45	16.72	11.78	8.82
2c	Luong	31.67	**32.18**	18.56	12.38	9.46	30.71	30.01	17.43	12.11	9.02
2d	Bahdanau	**31.93**	31.66	**18.70**	**12.79**	**9.53**	**31.56**	**31.32**	**18.36**	**12.36**	**9.25**
Our proposed systems											
2e	+T-Conv	32.08	30.08	18.15	12.88	9.97	31.34	30.94	18.26	12.71	9.76
2f	+DH-BiGRUs	31.55	30.21	18.29	13.05	9.84	31.20	31.46	17.64	12.40	9.65
2g	+TC-DHBG-Net (+T-Conv + DH-BiGRUs)	31.69	31.23	18.62	13.15	10.16	32.25	**32.19**	19.38	13.71	10.66
2h	+*FSDC*	32.13	**31.72**	18.84	12.98	9.79	31.52	31.72	19.04	13.01	9.71
2i	+*FSDC* + TC-DHBG-Net	**32.76**	31.43	**19.12**	**13.40**	**10.35**	**32.99**	31.86	**19.51**	**13.81**	**10.73**

Bold indicates the best performance.



(a) (b)

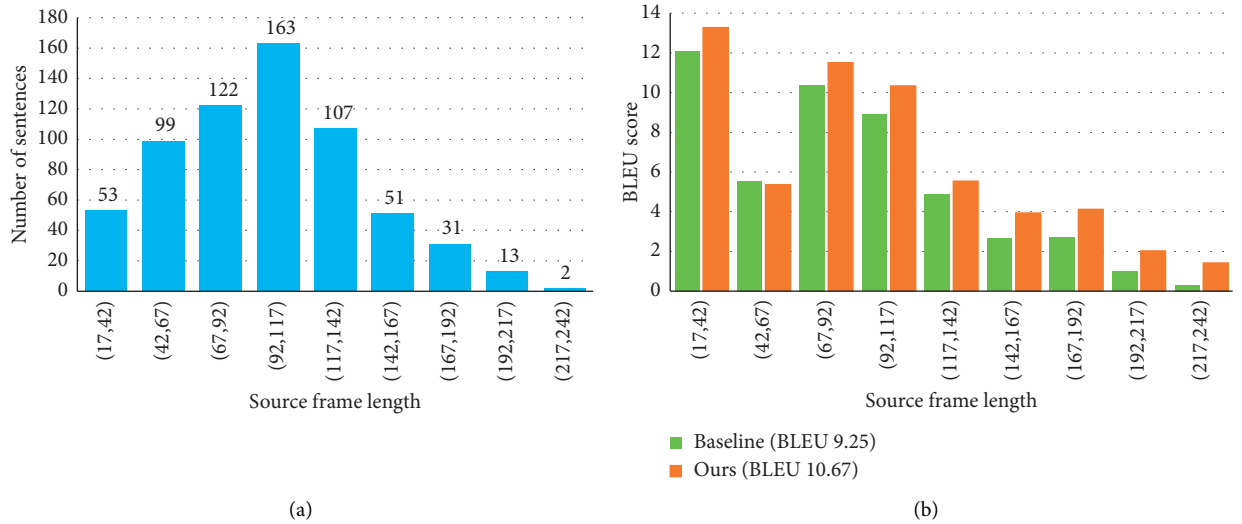
Figure 5: (a) Numbers and percentage of redundant frames with respect to different similarity thresholds. (b) The increased absolute values of BLEU compared to the baseline after using the *FSDC* algorithm. When the threshold is around 95%, both models reach the peak.

Table 3: BLEU scores on *DH-BiGRU* unit in different levels.

#	Levels	Development set					Test set				
		ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
3a	1	31.34	30.94	18.26	12.71	9.76	32.18	31.60	18.52	12.43	9.52
3b	2	31.69	31.23	18.62	13.15	10.16	32.08	30.08	18.15	12.88	9.97
3c	3	**33.02**	**32.37**	**19.49**	**13.44**	**10.21**	**32.25**	**32.19**	**19.38**	**13.71**	**10.66**
3d	4	31.52	31.40	18.71	13.00	9.87	31.58	31.85	18.95	13.17	10.03

Figure 6(b) shows the BLEU scores of generated translations on the test set with respect to the lengths of the source sentences. In particular, we split the translations into different bins according to the length of source sentences (frames), and then test the BLEU scores for translations in each bin separately with the results reported in Figure 6(b). Our approach can achieve big improvements over the baseline system in almost all bins, especially in the long sentences which have more than 117 frames. The performance comparison intuitively shows that our model can better adapt to the translation of long sentences, which benefits the *FSDC* algorithm and the improved encoder.

3.10. Qualitative Comparison. As shown in Table 5, to help readers understand our translations better, we qualitatively analyze the results of the sentence-level experiments. The sentences shown in the examples are both long sentences. The frame numbers of examples (a) and (b) are 192 and 196 frames, respectively. After using our *FSDC* optimization algorithm, the frame numbers are reduced to 182 and 169 frames, respectively. Since long sentences have serious long-distance dependency problems, both examples show that the current SLT models have poor translation ability to deal with long sentences. Comparing the baseline and our model, our model is relatively more accurate, and the meanings of the sentences are



(a) (b)

Figure 6: (a) Numbers and percentage of redundant frames with respect to different similarity thresholds. (b) The increased absolute values of BLEU compared to the baseline after using the *FSDC* algorithm. When the threshold is around 95%, both models reach the peak.

Table 4: BLEU scores vary in different thresholds.

#	Thresholds	94	95	96	97	98	99	100
	Baseline	—	—	—	—	—	—	9.25
4a	+*FSDC*	9.39	**9.71**	9.51	9.44	9.39	9.35	—
	△	+0.14	**+0.46**	+0.26	+0.19	+0.14	+0.10	—
	+Ours	—	—	—	—	—	—	10.66
4b	+Ours + *FSDC*	10.06	**10.73**	10.68	10.23	10.36	10.50	—
	△	+0.81(−0.60)	**+1.48 (+0.07)**	+1.43 (+0.02)	+0.98 (−0.43)	+1.11 (−0.30)	+1.25 (−0.16)	—

△ represents the increased absolute values of BLEU from the baseline, and the scores in parentheses represent the relative change value from +Ours. The *FSDC* algorithm does not work when the threshold is 100%.

closer to the ground true. Note that the translation results closer to the target in Table 5 are marked in bold.

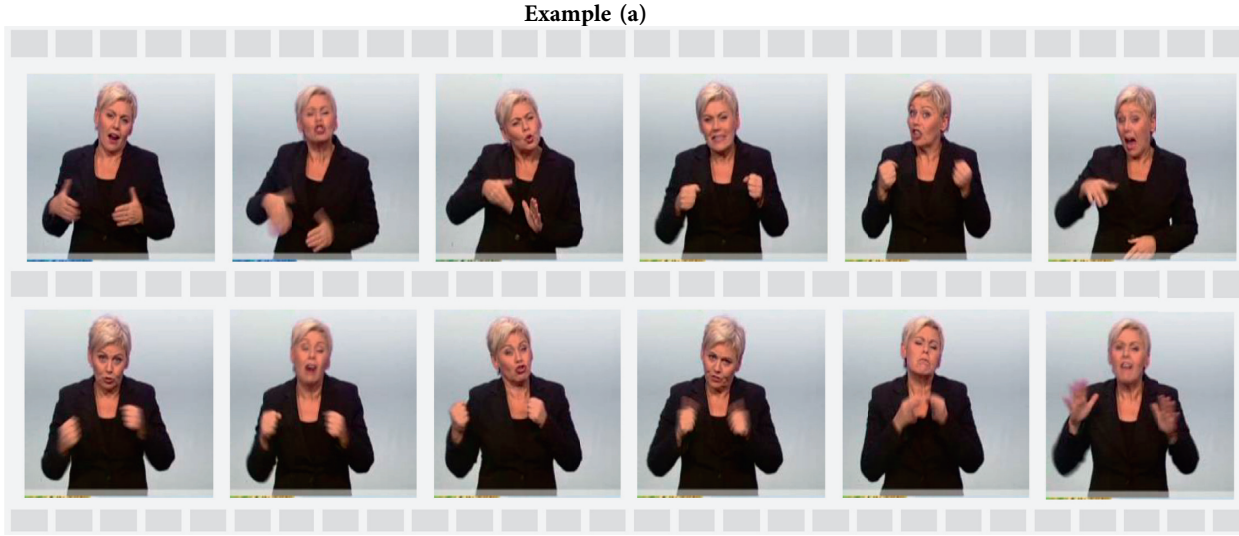
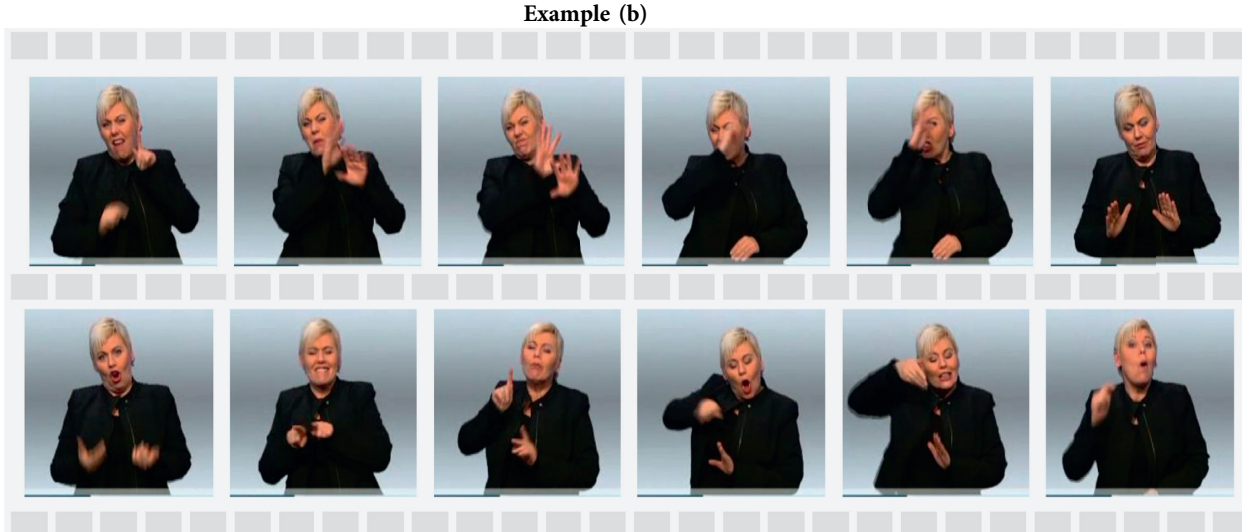
4. Related Work

According to a recent review [28], sign language is an ongoing research that began decades ago. The SLR system can be classified into three based on the type: (1) fingerspelling recognition; (2) isolated word recognition; (3) continuous sign sentence recognition. As for SLT, it is a more advanced task to further understand the semantic information of sign language.

In earlier work, the SLR system employed traditional recognition methods. For instance, Gao et al. [29] used HMM to recognize SLR words; The authors of [30, 31] used SVM to classify continuous sign language alphabets and isolated words; Baccouche et al. [32] performed a trajectory matching to classify the isolated words. Compared to the above, deep learning-based models have been employed recently. CNNs [33, 34], LSTMs [2, 35–37], or hybrid models [3, 38] have been used for continuous sentence recognition.

When it comes to SLT, few research results are published up to now. However, the development of SLR has laid a foundation for SLT. Camgoz et al. [1] released the first available continuous SLT dataset and proposed a neural SLT model. They combined CNN with the classic machine translation model-Seq2Seq. Their work maintains state of the art on the RWTH-PHOENIX-Weather 2014T dataset. Later, Ko et al. [4] proposed a neural SLT model based on human pose estimation, converting a video frame to keypoints, which simplifies the complexity of recognition, but ignored much important semantic information, e.g., expressions. We believe that it is under consideration. Guo et al. [2] proposed a hierarchical LSTM model that performed both SLR and SLT experiments on a Chinese dataset. They used 3DCNN for features extraction and compared it with the video captioning model S2VT [39]. The critical problem about their dataset is that it only includes 100 sentences, which is inappropriate for translation tasks. Overall, SLT achievement is still underperforming, limited by a lack of large-scale datasets and better translation models.

Table 5: Comparison of translations between our model and baseline.

Example (a)	
Source	
Target	der wind weht mäßig bis frisch mit starken bis stürmischen böen im bergland teilweise schwere sturmböen im südosten mitunter nur schwacher wind. (The wind blows moderately to fresh with strong to stormy gusts in the mountains, sometimes severe gusts in the southeast, sometimes only weak winds.)
BASE	der wind weht mäßig im norden frisch mit frisch mit stürmischen böen an der nordsee schwere sturmböen. (**The wind blows moderately** in the north fresh with fresh with stormy gusts at the north sea heavy gusts of wind.)
OURS	der wind weht mäßig bis frisch bei schauern und gewittern kann es stürmische böen auf den bergen sturmböen. (**The wind blows moderately to fresh** during showers and thunderstorms, it can be **stormy gusts on the mountains**.)
Frames	From 192 to 182
Example (b)	
Source	
Target	und morgen wird es dann in der südosthälfte nochmal ähnlich werden wie heute allerdings im nordwesten bereits dichtere wolken. (and tomorrow it will be similar again in the southeast half of the day as in the northwest, however, with thicker clouds.)
BASE	morgen im süden und süden bleibt es allerdings schon wolkenlücken und gewitter das wird es schon schon werden werden aus den westen. (**Tomorrow** in the south and south there will be cloud gaps and thunderstorms it will be from the west.)
OURS	und morgen wird es dann in der südosthälfte nochmal ähnlich am alpenrand wieder mal südwestwind und gewitter. (**and tomorrow it will be similar in the south-east half again** on the edge of the alps again south-west wind and thunderstorm.)
Frames	From 196 to 169

BASE: baseline model; **Ours**: the optimal model mentioned above; and the texts in parentheses represent the English translation corresponding to German.

5. Conclusion

In this work, we propose a novel weakly supervised SLT model with improved tokenization-related modules to adapt to longer sentences. We first propose an *FSDC* algorithm for temporal neighborhood to optimize the limited training data by removing the redundant frames and compress the sentence length to get a better interpretability. Then we introduce a *T-Conv* and *DH-BiGRU*-mixed NMT, which can consider the temporal information with reasonable resource

consumption as well as succeed in extracting deeper information. To evaluate our approaches, we conducted experiments on the public dataset-*RWTH-PHOENIX-Weather 2014T*. Compared with the existing state-of-the-art baseline, our model can reduce the size of training data by **9.3%** and outperform the baseline up to about **1.5+** BLEU-4 score on the sign-to-text translation task. Moreover, we conducted a series of comparison and ablation experiments and analyzed the translation performance qualitatively.

Despite the improved performance, SLT still has a lot of room to be studied. In future work, we will explore better interpretative methods to translate longer sentences.

Data Availability

The data we use can be accessed at https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX-2014-T/.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61573294, the National Social Science Foundation of China under Grant 16AZD049, the Outstanding Achievement Late Fund of the State Language Commission of China under Grant WT135-38, and the Fundamental Research Funds for the Central Universities under Grant 20720181002.

References

[1] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7784–7793, IEEE, Salt Lake City, UT, USA, March 2018.

[2] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical lstm for sign language translation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI, New Orleans, LA, USA, February 2018.

[3] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, AAAI, New Orleans, LA, USA, February 2018.

[4] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," *Applied Sciences*, vol. 9, no. 13, p. 2683, 2019.

[5] O. Koller, C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306–2320, 2019.

[6] S. Wang, D. Guo, W.-g. Zhou, Z.-J. Zha, and M. Wang, "Connectionist temporal fusion for sign language translation," in *Proceedings of the 26th ACM international conference on Multimedia*, ACM, New York, NY, USA, pp. 1483–1491, October 2018.

[7] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1243–1252, Sydney, Australia, August 2017.

[8] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *TPAMI*, vol. 35, no. 1, pp. 221–231, 2012.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, Santiago, Chile, December 2015.

[10] W. C. Stokoe, "Sign language structure," *Annual Review of Anthropology*, vol. 9, no. 1, pp. 365–390, 1980.

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

[12] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228, IEEE, Calgary, Canada, April 2018.

[13] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

[14] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," 2017, https://arxiv.org/abs/1703.08581.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] L. Gao, X. Li, J. Song, and H. T. Shen, "Hierarchical lstms with adaptive attention for visual captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1112–1131, 2020.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[19] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, https://arxiv.org/abs/1508.04025.

[20] 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.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, Miami, FL, USA, June 2009.

[22] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus rwth-phoenix-weather," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, European Language Resources Association, Reykjavik, Iceland, pp. 1911–1916, May 2014.

[23] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proceedings of the 2017 IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), pp. 7361–7369, IEEE, Honolulu, HI, USA, July 2017.

[24] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4297–4305, IEEE, Honolulu, HI, USA, July 2017.

[25] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics-ACL'02*, pp. 311–318, Stroudsburg, PA, USA, 2002.

[27] C.-Y. Lin, "Rouge: a package for automatic evaluation of summaries," in *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pp. 74–81, Barcelona, Spain, July 2004.

[28] S. M. Kamal, Y. Chen, S. Li, X. Shi, and J. Zheng, "Technical approaches to Chinese sign language processing: a review," *IEEE Access*, vol. 7, pp. 96926–96935, 2019.

[29] W. Gao, G. Fang, D. Zhao, and Y. Chen, "A Chinese sign language recognition system based on sofm/srn/hmm," *Pattern Recognition*, vol. 37, no. 12, pp. 2389–2402, 2004.

[30] T.-Y. Pan, L.-Y. Lo, C.-W. Yeh, J.-W. Li, H.-T. Liu, and M.-C. Hu, "Sign language recognition in complex background scene based on adaptive skin colour modelling and support vector machine," *IJBDI*, vol. 5, no. 1-2, pp. 21–30, 2018.

[31] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen, "Isolated sign language recognition with grassmann covariance matrices," *TACCESS*, vol. 8, no. 4, 2016.

[32] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behaviour Understanding*, pp. 29–39, Springer, Berlin, Germany, 2011.

[33] Z.-j. Liang, S.-b. Liao, and B.-z. Hu, "3d convolutional neural networks for dynamic sign language recognition," *The Computer Journal*, vol. 61, no. 11, pp. 1724–1736, 2018.

[34] S. Yang and Q. Zhu, "Video-based Chinese sign language recognition using convolutional neural network," in *Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, IEEE, Guangzhou, China, pp. 929–934, May 2017.

[35] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Phoenix, AZ, USA, pp. 2871–2875, September 2016.

[36] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Seattle, WA, USA, pp. 4165–4174, August 2019.

[37] S. Yang and Q. Zhu, "Continuous Chinese sign language recognition with cnn-lstm," in *Proceedings of the Ninth International Conference on Digital Image Processing (ICDIP 2017)*, Hong Kong, China, August 2017.

[38] C. Mao, S. Huang, X. Li, and Z. Ye, "Chinese sign language recognition with sequence to sequence learning," in *CCF Chinese Conference on Computer Vision*, pp. 180–191, Springer, Berlin, Germany, 2017.

[39] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4534–4542, Santiago, Chile, December 2015.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8882279, 12 pages
https://doi.org/10.1155/2020/8882279

Research Article

Application of Offshore Visibility Forecast Based on Temporal Convolutional Network and Transfer Learning

Zhenyu Lu,[1,2] **Cheng Zheng**,[2,3] **and Tingya Yang**[4]

[1]*School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China*
[2]*Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment, Nanjing, China*
[3]*School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China*
[4]*Jiangsu Meteorological Observatory, Nanjing, China*

Correspondence should be addressed to Zhenyu Lu; luzhenyu76@163.com

Received 22 June 2020; Revised 3 September 2020; Accepted 5 October 2020; Published 20 October 2020

Academic Editor: Nian Zhang

Visibility forecasting in offshore areas faces the problems of low observational data and complex weather. This paper proposes an intelligent prediction method of offshore visibility based on temporal convolutional network (TCN) and transfer learning to solve the problem. First, preprocess the visibility data sets of the source and target domains to improve the quality of the data. Then, build a model based on temporal convolutional network and transfer learning (TCN_TL) to learn the visibility data of the source domain. Finally, after transferring the knowledge learned from a large amount of data in the source domain, the model learns the small data set in the target domain. After completing the training, the model data of the European Mid-Range Weather Forecast Center (ECMWF) meteorological field were selected to test the model performance. The method proposed in this paper has achieved relatively good results in the visibility forecast of Qiongzhou Strait. Taking Haikou Station in the spring and winter of 2018 as an example, the forecast error is significantly lower than that before the transfer learning, and the forecast score is increased by 0.11 within the 0-1 km level and the 24 h forecast period. Compared with the CUACE forecast results, the forecast error of TCN_TL is smaller than that of the former, and the TS score is improved by 0.16. The results show that under the condition of small data sets, transfer learning improves the prediction performance of the model, and TCN_TL performs better than other deep learning methods and CUACE.

1. Introduction

Atmospheric visibility is an indicator used to judge the transparency of the atmosphere. It refers to the maximum horizontal distance that the person with normal vision can distinguish the outline of the target from the background when observing the black target with the sky as the background under the weather conditions at that time [1]. There are many climatic factors that affect the visibility of offshore waters, such as fog, haze, smoke, dust, precipitation, etc. The most important factor is fog [2, 3]. Low visibility often affects the travel safety of offshore vessels and can easily cause accidents at sea. Therefore, how to predict the occurrence of low-visibility weather in offshore areas as much as possible is a problem that researchers are concerned about. Here, the offshore visibility prediction method based on transfer learning proposed in this paper aims to improve the current situation of offshore visibility prediction.

The visibility forecast used to rely on traditional numerical forecasting methods. The weather conditions in the future were calculated through numerical forecasting methods based on the theoretical basis of fluid mechanics, atmospheric dynamics, and thermodynamics. In recent years, with the continuous development and improvement of machine learning and deep learning, the application of machine learning and deep learning to visibility prediction has also become a hot spot for researchers [4–6].

As a commonly used meteorological forecasting method, traditional numerical forecasting is widely used. The National Center for Atmospheric Research (NCAR) has jointly developed a multiscale, multiprocess model system that is online and fully coupled. It is a widely used

meteorological–chemical coupled model named weather research and forecast model coupled with chemistry (WRF-Chem) [7, 8]. The US Environmental Protection Agency (EPA) has also developed a universal multiscale air quality model named congestion mitigation and air quality (CMAQ) [9, 10]. These two forecasting models have been localized in various regions in China, and the WRF-Chem model has been introduced in East China to establish a numerical forecast model named Beijing regional environmental meteorology prediction system (BREMPS) in East China. It is applied to visibility prediction in East China [11, 12]. The introduction of the CMAQ model in the southern region established the Southern China regional environmental meteorological numerical forecasting model named Guangdong Regional Assimilation Chemistry Environmental System (GRACES), which was applied to the operational forecasting of the Pearl River Delta [9]. At the same time, the domestic environmental meteorological numerical model is also continuously researched. The Chinese Academy of Meteorological Sciences has independently developed a national-level numerical haze forecasting system named CMA Unified Atmosphere Chemistry Environment (CUACE). CUACE online coupled with the chemical weather model, which has been used in environmental meteorological operations nationwide [13, 14].

In recent years, researchers have achieved a lot of results in machine learning and deep learning [15–19] and have applied them to various fields [20–24]. Among them, machine learning and deep learning also show advantages in weather forecasting such as visibility forecasting and air quality forecasting [25–28]. Machine learning algorithms have excellent performance in visibility forecasting. The Support Vector Machine (SVM) method was used to select multiple kernel functions for the forecasting modeling experiment of low-visibility weather in Shuangliu Airport and study the impact of various meteorological elements on visibility [29]. Long short-term memory-fully connected (LSTM-FC) neural network was used to predict the pollutants in Beijing area and achieved good results [30]. Li et al. used a hybrid CNN-LSTM model developed by combining the convolutional neural network (CNN) with the long short-term memory (LSTM) neural network for forecasting the next 24 h PM2.5 concentration in Beijing [31]. BP neural network was used to construct the visibility forecast model of the Bohai Rim city to reduce the visibility forecast error [32]. In terms of visibility prediction for offshore waters, people used classification and tree regression to establish a forecast model of sea fog along the coast of Qingdao [33]. However, these methods are not ideal for forecasting offshore visibility. They can only predict the general trend of visibility. The accuracy of low-visibility forecasting and visibility-level forecasting needs to be improved.

Both machine learning and deep learning network models are inseparable from large amounts of data. Using deep learning to predict visibility requires a large amount of meteorological data to train the network. However, offshore observation sites are scarce, so using deep learning to face offshore visibility forecasting is faced with the problem of insufficient data. Therefore, this article aims to use transfer learning to solve this problem.

Transfer learning is a machine learning method and a problem-solving idea [34]. In recent years, it has been widely used in various research fields to solve the difficulties caused by insufficient data and achieved excellent performance in building energy prediction and text prediction [35–37]. It is a way to achieve the knowledge transfer between the source and target domains by modeling the distribution of data in the source and target domains, thereby improving the performance of the algorithm [38]. The transfer learning method can better solve the problem of poor forecast performance caused by less data.

In order to improve the forecast accuracy of offshore visibility under the condition of a small data set, this paper uses time convolutional network (TCN) and transfer learning to establish a forecast model to achieve an objective forecast of visibility in offshore areas such as Qiongzhou Strait. At the same time, it improves the ability of forecasting and early warning services for the haze and sea fog in the strait, provides technical support for the low-visibility weather forecast in this area, and ensures the safety of ships traveling.

2. Materials and Methods

2.1. Data Source and Data Preprocessing. The climatic conditions in the coastal area of South China are subtropical monsoon climate, which belongs to the East Asian monsoon region. The meteorological background is greatly affected by relative humidity, and sea fog occurs frequently in winter and spring, so low-visibility weather occurs more frequently. The training data used in this article is divided into two parts. The first part is the meteorological data and environmental data of the stations in Leizhou Peninsula and northern Hainan, which is the source domain data. The second part is a small amount of meteorological data and environmental protection data of the weather stations on both sides of the Qiongzhou Strait, which is the target domain data. The specific data of the first part include routine ground observation data and high-altitude data of Leizhou Peninsula area and northern Hainan area from 2016 to 2018, such as data on wind speed, relative humidity, temperature, pollutants, etc., and visibility observation data. The specific data of the second part include visibility observation data and meteorological data and environmental protection data of the offshore sites of the Qiongzhou Straits on both sides of the north and south sides of the Qiongzhou Strait from January to April 2016–2018. The verification data used in this paper is the European Meteorological Forecast Center (ECMWF) meteorological field model data in 2018. In this paper, four inland sites are selected as source domain site data, namely, Haikang (59750), Danxian (59845), Chengmai (59843), and Anding (59851). The data of four stations on the seashore were selected to conduct a forecast experiment on the visibility of Qiongzhou Strait, namely, Xu Wen (59754), Haikou (59758), Lingao (59842), and Qiongshan (59757). Figure 1 shows the distribution of the eight sites.

Because the weather forecasting factors are composed of different parts, it is necessary to compose the data for time series analysis and time-space matching and then compose the input data that meets the requirements. At the same time, due to various factors in the real-time observation of

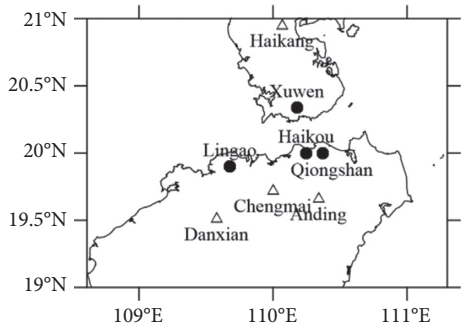


Figure 1: Distribution of representative sites of Qiongzhou Strait.

the meteorological conditions by the meteorological station, the meteorological observation data will often have some observations missing and abnormal, so the data should be cleaned. This paper uses Lagrange interpolation to fill in missing values. Lagrange interpolation formula is

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}, \tag{1}$$

$$\begin{aligned} L(x) = {} & y_1 \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)} \\ & + y_2 \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)} \\ & + \cdots y_n \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_n-x_1)(x_n-x_3)\cdots(x_n-x_{n-1})}. \end{aligned} \tag{2}$$

Equation (2) solves the Lagrange interpolation polynomial $L(x)$ and then substitutes the point corresponding to the missing value x to obtain the approximate value of the missing value.

The visibility data sample is composed of two parts, meteorological data and environmental protection data, which have a large number of features. In order to improve efficiency, the characteristics of the data samples need to be screened. In this paper, the Pearson correlation coefficient method is used to measure the degree of correlation between two variables. The Pearson correlation coefficient method is used to calculate the correlation between the features that affect visibility and the observation of visibility, and the features are screened by comparing the correlations of the various impact features. The formula of the correlation coefficient is shown in formula (3). In this paper, X represented the meteorological feature in the data, and Y represented the value of visibility:

$$\rho_{XY} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu x)(Y-\mu y)]}{\sigma_X \sigma_Y}. \tag{3}$$

The forecast meteorological elements of visibility in the data include 40 different forecast meteorological elements such as temperature, relative humidity, wind shear, and wind speed. The meteorological elements of visibility forecast after screening by the Pearson coefficient method are shown in Table 1. The visibility forecast meteorological elements screened by the Pearson coefficient method are shown in Table 1. In this paper, 12 meteorological elements are selected and input into the network as forecast factors. Table 1 shows the correlation coefficients between each forecasting factor and the observation of visibility. It can be clearly observed that the correlation coefficients of temperature and humidity are relatively high. Visibility in Haikou area is highly correlated with temperature, relative humidity, etc., which is a good proof of the reliability of using the correlation coefficient method to filter visibility prediction factors here [39]. The forecast feature data includes historical weather data and historical environmental data, which are matched in time and space to form the original visibility data. The raw data are cleaned to obtain the final visibility forecast data. The processing flow is shown in Figure 2.

2.2. Method

2.2.1. Temporal Convolutional Network. Because the Temporal Convolutional Network (TCN) is a network structure that can better handle time series data, this paper uses TCN to build a prediction model of the visibility of the source domain. TCN is the second architecture that can analyze temporal data in addition to the Recurrent Neural Network (RNN) architecture. The structure of TCN is shown in Figure 2. TCN has two main characteristics: First, there is a causal relationship between the layers of the convolutional network, which means that there will be no "bobble" historical information or future data. Even if a long short-term memory network (LSTM) with the same time series processing function has a memory gate, it cannot completely remember all the historical information, let alone if some information is useless in the LSTM will gradually be forgotten [40]. Second, architecture of TCN can be flexibly adjusted to any length, and it can be mapped to correspond to several interfaces according to the output terminal. This is the same as the RNN framework, which is very convenient. The TCN network adopts the form of convolution, which is mainly composed of causal convolution, hole convolution, residual module, and full convolution network. Among them, causal convolution is used to make the network suitable for sequence models. The output of the convolution layer at time t is only convolved with the elements of the current layer and the previous layer. The causal convolution calculation formula at x_t is

$$(F * X)(x_t) = \sum_{k=1}^{K} f_k x_{t-K+k}. \tag{4}$$

Among them, filter is $F = (f_1, f_2, \ldots, f_k)$, and input sequence is $X = (x_1, x_2, \ldots, x_T)$. The use of the hole convolution and residual modules allows the TCN network to remember history. The hole convolution kernel is

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot X_{s-d \cdot i}, \tag{5}$$

where d is the hole coefficient, k is the size of the filter, and $s - d \cdot i$ is the past direction [40]. The structure of the hollow convolution is shown in Figure 3(a). When the filter is

Table 1: Forecast factors and correlation coefficients.

Predictor	Pearson's correlation coefficient absolute value
Temperature	0.563
925 hPa temperature	0.511
900 hPa temperature	0.498
950 hPa horizontal wind speed	0.489
Atmospheric pressure	0.473
925 hPa horizontal wind speed	0.397
Vertical wind speed	0.390
925 hPa relative humidity	0.379
900 hPa horizontal wind speed	0.372
900 hPa relative humidity	0.338
Depression of the dew point	0.333
Relative humidity	0.328
950 hPa vertical wind speed	0.323

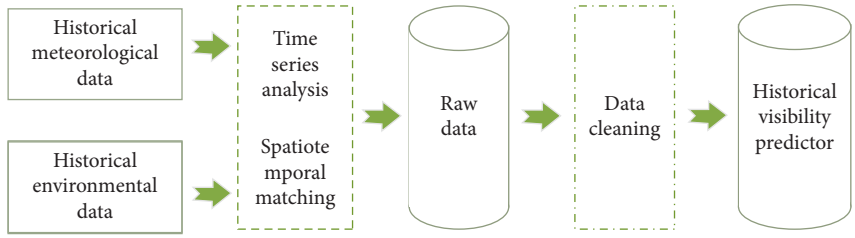


Figure 2: Data preprocessing flowchart.

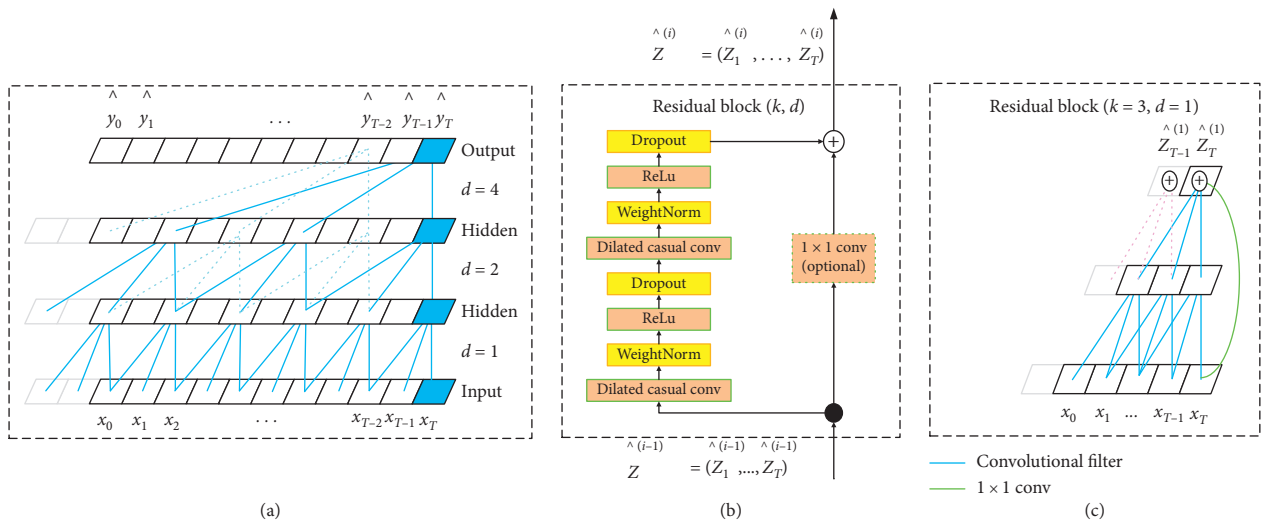


Figure 3: TCN architecture diagram. (a) Skip connection in TCN residual module. (b) Residual module of TCN. (c) Skip connection in TCN residual module.

$F = (f_1, f_2, \ldots, f_k)$, input sequence is $X = (x_1, x_2, \ldots, x_T)$, and the convolution of the holes at x_t is

$$(F *_d X)(x_t) = \sum_{k=1}^{K} f_k x_{t-(K-k)d}. \quad (6)$$

In formula (6), $d$can increase the receptive field, and the size is $(K-1)d+1$. The formula for the residual module is

$$o = \text{activation}(x + \mathcal{F}(x)). \quad (7)$$

In formula (7), $\mathcal{F}$is a part of a series of transformations, and x is an input. The structure diagrams are shown in Figures 3(b) and 3(c).

Full convolutional networks make the output and input dimensions consistent, simplifying the network. It is these parts that make up TCN, which makes the network have the advantages of good parallelism, flexible receptive fields, small training memory, and adjustable input sequence length compared with LSTM.

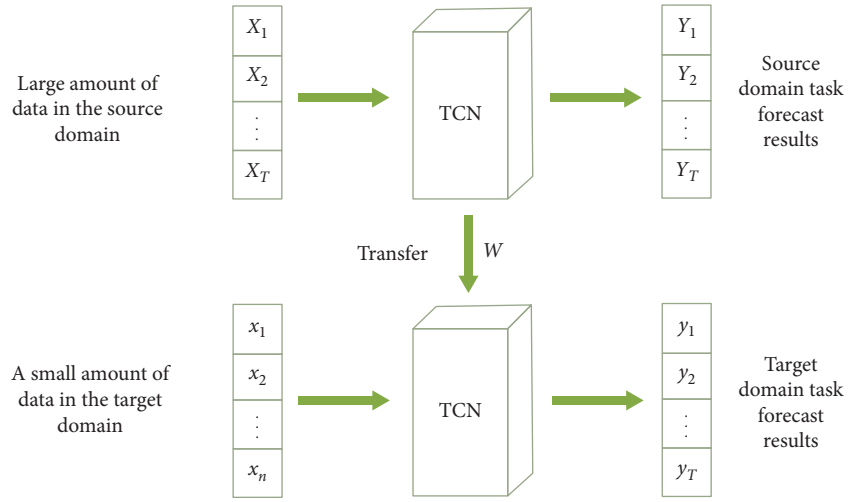


Figure 4: The transfer process of the source domain model to the target domain model.

2.2.2. Transfer Learning. Time convolutional networks belong to deep neural networks. During the training process of deep neural networks, the network will generate a large number of network parameters, so a large amount of data is needed to train the parameters. However, there is a problem of low data volume in offshore visibility, and the learning effect is not good. Therefore, the method of transfer learning is used to solve this problem.

However, there is a problem of low data volume in offshore visibility, and the learning effect is not good. Therefore, the method of transfer learning is used to solve this problem. Researcher defined the transfer learning in detail: given source domain $D_S = \{(X_1^{(S)}, Y_1^{(S)}), \ldots, (X_{n_s}^{(S)}, Y_{n_s}^{(S)})\}$ and source domain learning tasks T_s, as well as target domain $D_t = \{X_1^{(t)}, \ldots, X_{n_t}^{(t)}\}$ and target domain learning tasks T_t. The ultimate goal of transfer learning is to gradually improve the performance of the prediction function in the target domain D_S and the target domain task T_s by learning the knowledge in the source domain D_t and the source domain task T_s [41]. In this paper, a prediction model is established in the source domain task, and a large amount of source domain data is used as the training data of the model, and the pretrained model is saved after training. Transfer learning is used to make the target domain task network inherit the weights of the pretrained model. When performing a new task, a small amount of new visibility data is used as input, and the weight of the pretrained model loaded into the source domain task is used as the initial weight of the target domain network for training. The training process is shown in Figure 4.

2.3. Evaluation Method. From January to April 2018, there was frequent sea fog in the Qiongzhou Strait area with low visibility. The classification of visibility observation data during this period is relatively clear, and the selection of data during this period can make the experiment comprehensive effect better.

In this paper, there are two ways to evaluate the forecast performance, which are numerical test and classification test. The numerical test uses two indicators: root mean square error (RMSE) and mean absolute error (MAE). RMSE is used to measure the deviation between the observed value and the predicted value. It usually reflects the precision of the measurement well. The smaller the value of RMSE, the higher the precision. MAE represents the average value of the absolute error between the predicted value and the observed value. The smaller the value of MAE, the higher the accuracy of prediction. MAE represents the average value of the absolute error between the predicted value and the observed value. The smaller the value of MAE, the higher the accuracy of prediction. The calculation formula of the two is as follows:

$$\mathrm{RMSE}(X,P,O) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(p(x_i) - o_i\right)^2}, \tag{8}$$

$$\mathrm{MAE}(X,P,O) = \frac{1}{m}\sum_{i=1}^{m}\left|p(x_i) - o_i\right|. \tag{9}$$

In formulas (8) and (9), m represents the number of samples, X represents the forecast factor of historical visibility, P represents the predicted value of the model for visibility, and O represents the observed value of historical visibility.

In this paper, the visibility level is mainly divided into four levels, namely, 0~1 km, 1~5 km, 5~10 km, and above 10 km. Among them, improving the forecast accuracy of the visibility level of 0 to 1 km has important practical guiding significance. The graded forecast test uses TS scoring rules to test forecast performance of the model for each visibility level. The formula for the test method is as follows:

$$\mathrm{TS} = \frac{\mathrm{NA}}{\mathrm{NA} + \mathrm{NB} + \mathrm{NC}}. \tag{10}$$

In the formula, NA represents the number of correct forecasts, NB represents the number of empty forecasts, and NC represents the number of missed forecasts. Correct forecast means that the forecast level is the same as the live

level; empty forecast means that the forecast level is less than the live level; and missed forecast means that the forecast level is higher than the live level.

2.4. Forecasting Process. The forecast technology flow used in this paper is shown in Figure 5. This forecast model uses the TCN model and transfer learning method to forecast the Qiongzhou Strait. In the first step, the source domain prediction factor obtained after data preprocessing is used as the input feature of the TCN network to train the network. Based on the loss function result, adjust the parameters to optimize the model performance iteratively and save the optimal training model. In the second step, the transfer learning method based on parameter transfer is used to transfer the weight of the pretrained model in the source domain as the initial weight of the network and start learning the data in the target domain. Finally, input the EWMCF meteorological field data of the forecast period to check the model performance and output the forecast results.

3. Results and Discussion

3.1. Lab Environment. The experiment in this paper is implemented on TensorFlow 1.14.0 framework under Ubuntu 16.04 system and uses GPU to accelerate. The hardware configuration of the experimental platform is CPU: Intel Core i5-8600k, GPU: NVIDIA GTX 1080Ti, and memory is 16G.

3.2. Experimental Parameter Setting. Hyperparameters are the parameters set by the network model before learning, not the parameters obtained through training. Hyperparameter settings are crucial to the efficiency of model learning. Under normal circumstances, the model training should be selected before the start of learning. After observing the value of the loss function and the training status during the model learning process and adjusting the hyperparameters, the model learning efficiency is the highest. Among many hyperparameters, hyperparameters such as learning rate and batch size have the greatest impact on the efficiency and accuracy of learning. After repeated experiments, the model performs best when the learning rate is 0.002, the batch size is 60, and the hidden unit is 150.

3.3. Result Analysis

3.3.1. Comparison of Results before and after Transfer Learning. After establishing the prediction model named TCN_TL based on TCN and transfer learning, the network was tested and evaluated by using the offshore visibility data from January to April 2018. Figure 6 shows the MAE and RMSE of the forecasted visibility values of Haikou Station before and after the transfer learning in February 2018, including 24 h, 48 h, 72 h, and 96 h four-time error comparison. Figure 6 takes the Haikou Observation Station in February 2018 as an example and gives the visibility observation values and forecast values that change daily. It is not difficult to find that in each forecast period, the visibility before and after transfer learning can better reflect the change trend of visibility, but there is a deviation between the two in the forecast value and magnitude. Table 2 shows the graded forecast score of Haikou Station in February 2018, named TS score. By comparing the RMSE and MAE of different forecast aging and the grading forecast scores of different aging, we can find that, regardless of migration, the error of short-term forecast is always smaller than the error of long-term forecast. And the TS score of short-term forecasts is always higher than the longer time-sensitive TS score. Therefore, it can be concluded that the performance of short-term forecast is better than that of long-term forecast.

Taking the 24-hour time-effect forecast as an example, the visibility forecast of the Haikou Station before and after transfer learning in February 2018 is analyzed. Figure 7 shows the forecast situation before and after transfer learning. The forecast value of TCN_TL for visibility is closer to the actual value of visibility than the forecast value of TCN. After transfer learning, both RMSE and MAE decreased significantly, RMSE decreased from 9 km to 6.2 km, and MAE decreased from 5.2 km to 2 km. As shown in Table 2, the accuracy of the grading forecast has also improved. At the 0~1 km level, the TS score increased from 0.23 to 0.35. At the 1~5 km level, the TS score increased from 0.41 to 0.5, and at the 5~10 km level, the TS score was 0.52, increased to 0.67. And the accuracy rate increased from 0.64 to 0.76 at the 10~35 km level. It is worth noting that although the TS score has increased in all levels, there is still room for improvement. Through transfer learning, the match between the predicted value of visibility and the observed value has been significantly improved.

3.3.2. Comparison and Analysis of Different Model Results. In order to better test the prediction performance of TCN_TL for visibility in offshore areas, this paper compares its experimental results with the experimental results of the other three models without transfer learning. It should be noted that the following experiments used historical visibility forecast data from 2016 to 2018 to forecast the visibility from January to April 2018 in the Qiongzhou Strait region. Figure 8 shows the errors between the predicted and observed values for the next 24 h, 48 h, 72 h, and 96 h under different models. The forecast errors of each model are given in the figure, which are the TCN_TL and CUACE models, the forecast model based on BP neural network, the forecast model based on LSTM network, and the forecast model based on TCN. Among them, the TCN_TL model uses two parts of the historical visibility observation data of the source domain and the target domain during training, while other models use the historical visibility observation data of the target domain for training.

From the data shown in Figure 8, it can be found that no matter what kind of forecast model, the longer the forecast time, the greater the error between the model's forecast and the observed value. Among them, the performance of the forecast model based on TCN_TL is significantly better than other forecast models. Figure 8(a) shows the root-mean-square error of the predicted and observed visibility of the model. Taking the 24 hr forecast period of validity as an example, the RMSE of the CUACE model is higher than other models. The LSTM network prediction model has a lower RMSE than the BP neural network, while the TCN prediction

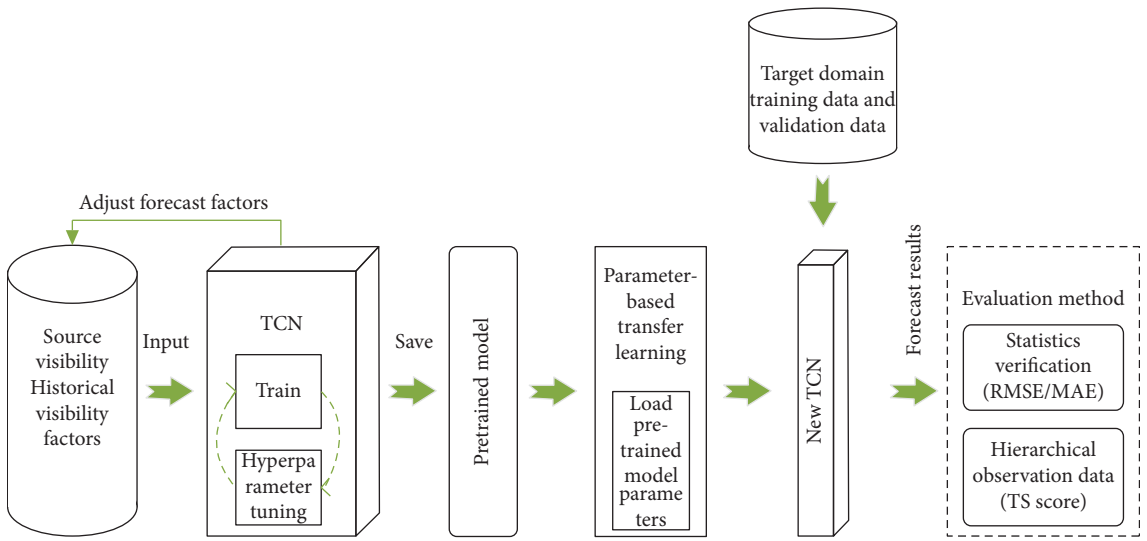


Figure 5: Technical process of visibility forecast based on TCN and transfer learning.

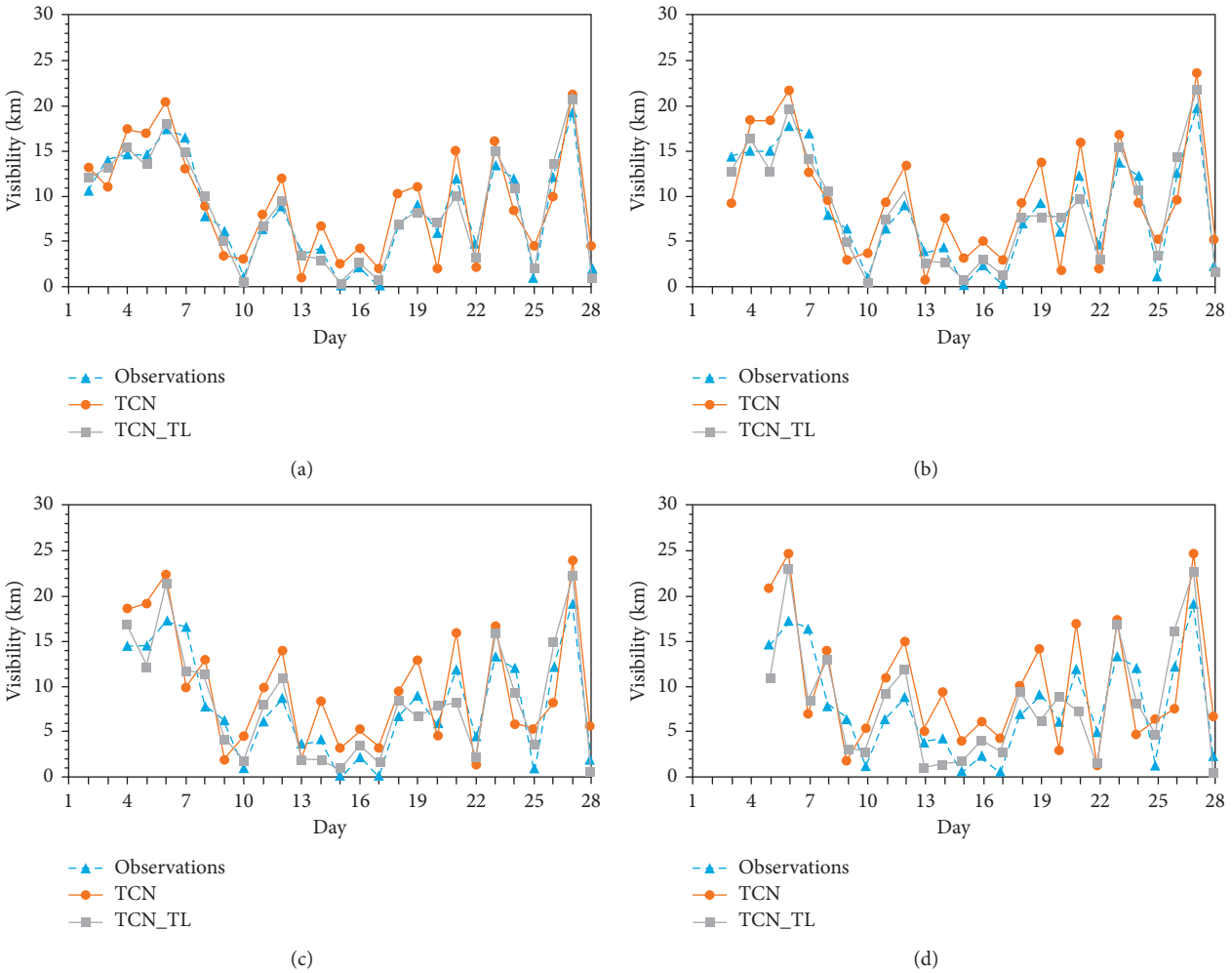


Figure 6: Daily changes of the visibility observation and forecast results of (a) 24 h, (b) 48 h, (c) 72 h, and (d) 96 h at the Haikou Station in February 2018.

Table 2: TS scores of visibility forecast in February 2018 at Haikou Station.

Classification (km)	0~24 h		24~48 h		48~72 h		72~96 h	
	TCN	TCN_TL	TCN	TCN_TL	TCN	TCN_TL	TCN	TCN_TL
[0, 1]	0.23	0.35	0.15	0.26	0.1	0.2	0.1	0.15
[1, 5]	0.41	0.5	0.36	0.54	0.3	0.45	0.23	0.31
[5, 10]	0.52	0.67	0.5	0.61	0.42	0.56	0.51	0.58
[10, 35]	0.64	0.76	0.6	0.7	0.56	0.7	0.63	0.63

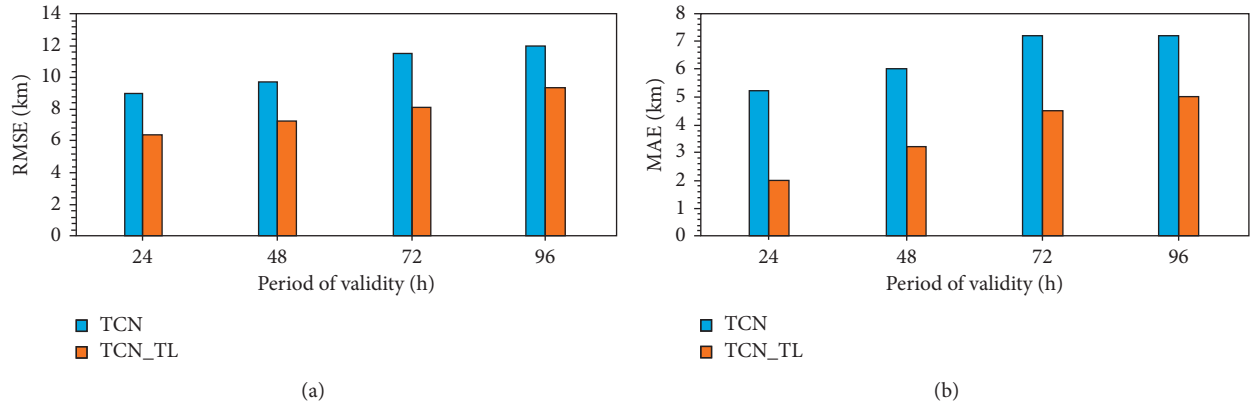


Figure 7: Statistical characteristics between the observed and forecasted visibility of February 2018 at Haikou Station. (a) RMSE. (b) MAE.

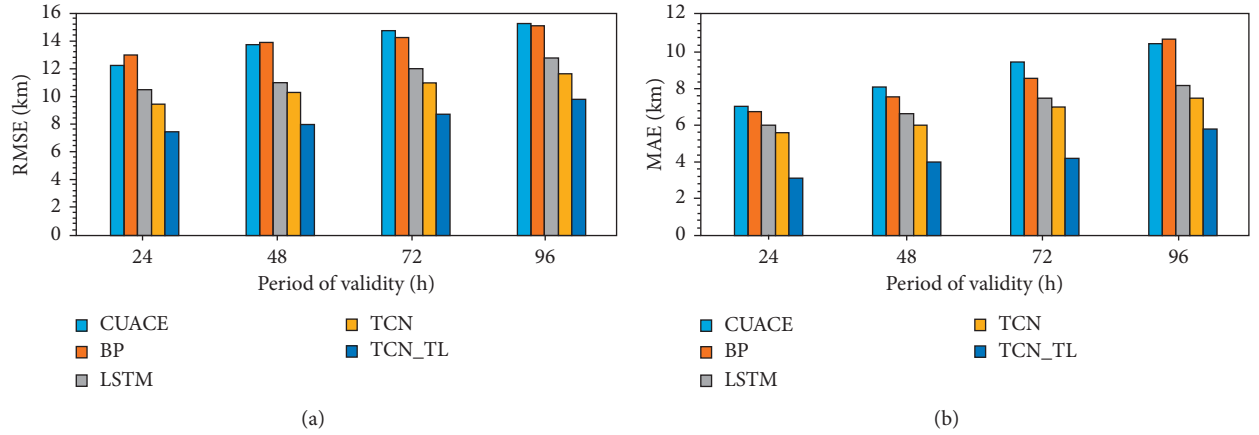


Figure 8: Statistical characteristics between the observed and forecasted visibility from February to April 2018 at Haikou Station. (a) RMSE. (b) MAE.

model has a slightly lower RMSE than LSTM. The RMSE of the TCN_TL prediction model using the transfer learning method is significantly lower than the RMSE of the TCN prediction model and other prediction models. The RMSE of the five forecasting models is 14.2 km, 14 km, 10.5 km, 9.4 km, and 7.4 km, respectively. Figure 8(b) shows the average absolute error between the predicted value and the observed value of the model. The MAE of the five forecasting models is 7 km, 6.7 km, 6 km, 5.6 km, and 3.1 km.

From the analysis in Figure 8, we can see that TCN_TL using the transfer learning method performs better than the other four models in both RMSE and MAE. Comparing TCN with the CUACE model, the BP neural network prediction model, and the LSTM prediction model, the performance of TCN is relatively good. From the comparison of the errors between the TCN and TCN_TL models, the use of transfer learning to compensate for the problem of small data volume significantly reduces the error between the predicted visibility and the observed visibility and improves the prediction performance of the model.

Generally speaking, the weather with low-visibility level, especially the weather with a level of 0~1 km, appears less frequently than ordinary weather, and the opportunities for model learning are also much less. It is relatively low, but improving the accuracy of this level of forecasting has practical guiding significance. As shown in Table 3, taking the 24 hr forecast as an example, when the visibility is less than 1 km, the BP neural network prediction model is only 0.18. The TS score of TCN is slightly higher than LSTM, reaching 0.25. TCN_TL has the highest score, reaching 0.36. In

Table 3: 24 hr time-efficient grading forecast TS score.

Classification (km)	CUACE	BP	LSTM	TCN	TCN_TL
[0, 1]	0.14	0.18	0.21	0.25	0.36
[1, 5]	0.33	0.35	0.4	0.45	0.52
[5, 10]	0.38	0.37	0.51	0.58	0.67
[10, 35]	0.47	0.48	0.62	0.68	0.78

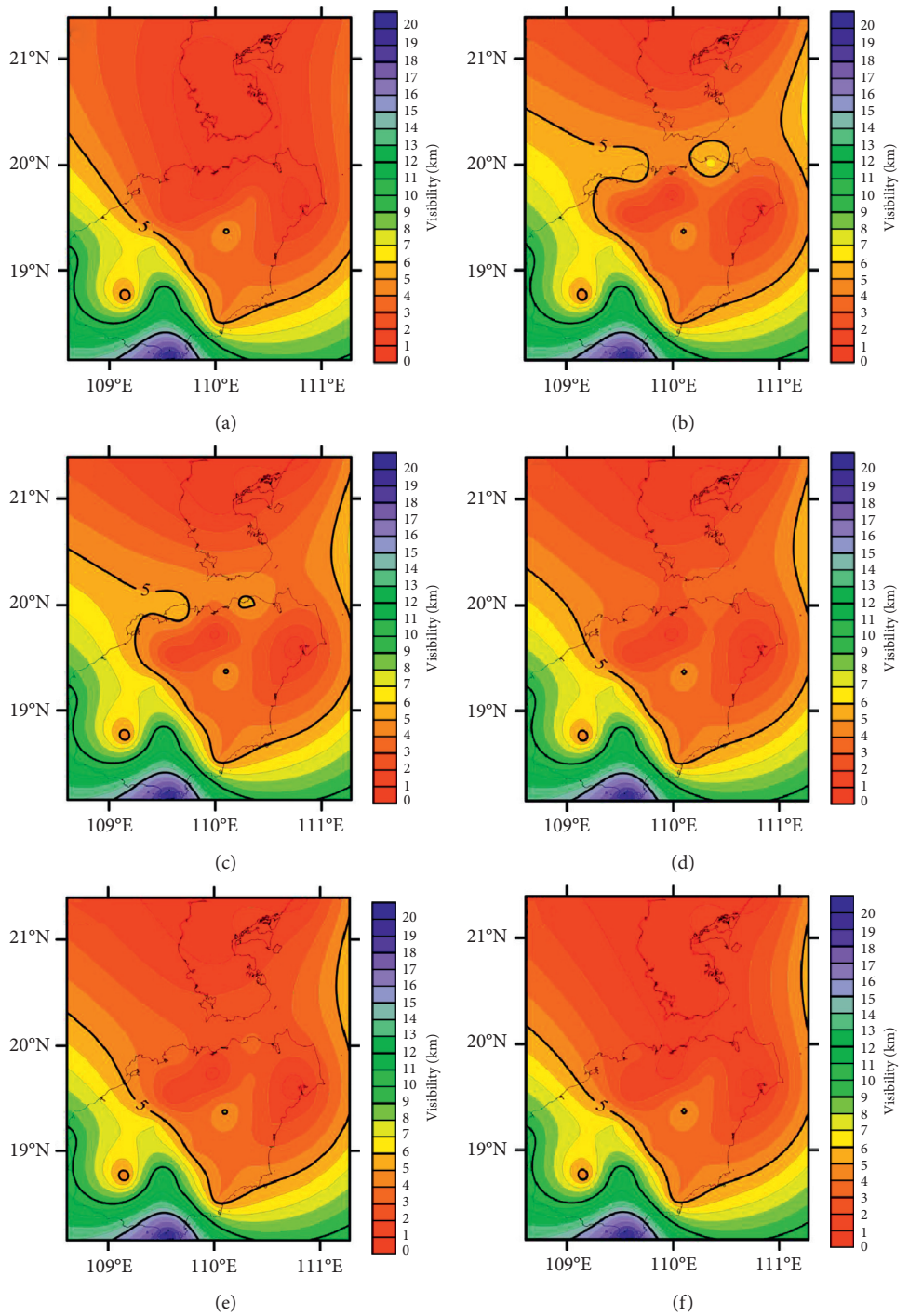


Figure 9: Visibility spatial distribution. (a) February 16th 8:00 measured map. (b) February 16th 8:00 CUACE model forecast map. (c) February 16th 8:00 BP model forecast map. (d) February 16th 8:00 LSTM model forecast map. (e) February 16th 8:00 TCN model forecast map. (f) February 16th 8:00 TCN_TL model forecast map.

addition, at the other three levels, the TS score of TCN_TL is also higher than the other four forecast models. Therefore, the results show that, compared with other traditional methods and machine learning methods, TCN_TL has more advantages in predicting the performance of offshore visibility.

3.4. Visibility Spatial Distribution Analysis. During the Spring Festival of 2018, the Qiongzhou Strait experienced persistent low-visibility weather. This section takes this event as an example and uses a 24 hr forecast model to perform forecast analysis. Figure 9 shows the visibility forecast of each model at 8 AM on February 16. Because the Kriging interpolation method considers the variation distribution of spatial attributes, it can effectively eliminate the errors caused by uneven sampling and make the results more in line with the actual situation [42]. Here, the Kriging interpolation method is used to interpolate the visibility spatial results of the Qiongzhou Strait, which better shows the spatial distribution of visibility. According to the spatial distribution of actual observation results of medium visibility in Figure 9(a), the Qiongzhou Strait is under low-visibility weather. Comparing the spatial distribution of visibility prediction results in Figures 9(b)–9(f), the overall spatial distribution state is gradually tending towards the actual spatial distribution state of visibility and the prediction result space of TCN_TL. The distribution is closest to the spatial distribution of actual observations. Comparing Figures 9(e) and 9(f), it can be found that the prediction result of TCN_TL is closer to the actual observation result than the prediction result of TCN. Therefore, the use of transfer learning improves the prediction accuracy of the model. Figure 9(b) is the spatial distribution of the CUACE forecast results. The forecast results are higher than the actual observation values, and the TCN_TL forecast results are more in line with the actual situation. Therefore, under the condition of small data set, the prediction performance of TCN_TL is better than that of CUACE.

4. Conclusions

The offshore visibility prediction method based on transfer learning proposed in this paper combines TCN and transfer learning. This paper uses TCN to establish a source domain forecast model and learns the knowledge of the source domain under the premise that the source domain has a large amount of data. And TCN_TL was used to forecast the visibility of the target domain offshore. The following conclusions can be obtained through experimental analysis:

(1) This paper compares the results of TCN and TCN_TL for forecasting offshore visibility. The experimental results show that the model of transfer learning can be used to learn the visibility knowledge of the source domain under the condition of a small amount of offshore meteorological observation data to improve the accuracy of the visibility forecast of the target domain.

(2) In this paper, the TCN network is used to learn the source domain data. Compared with LSTM and BP neural networks, the RMSE and MAE between TCN prediction and observation are smaller, and the TS score is relatively higher than others' in each visibility forecast level. Therefore, TCN is more advantageous for learning and predicting time series like visibility data than others.

(3) The 24-hour forecast of offshore visibility of Haikou Station from January to April 2018 was taken as an example. Under the conditions of small data sets, comparing the forecast results of TCN_TL and CUACE, the forecast error of TCN_TL is lower than that of CUACE, and the RMSE and the MAE are 6.8 km and 3.9 km. RMSE drops to 7.4 km, and MAE drops to 3.1 km. TS score of TCN_TL has also improved in each forecast level. At the level of 0~1 km, the TS score is 0.36, increased by 0.16. At the level of 1~5 km, the TS score is 0.52, increased by 0.19. At the level of 1~5 km, the TS score is 0.52, increased by 0.19. At the level of 5~10 km, the TS score is 0.67, increased by 0.29. At the level of greater than 10 km, the TS score is 0.78, increased by 0.39. Therefore, under the condition of a small data set, TCN_TL has an advantage in predicting the visibility of offshore waters than CUACE.

(4) It is worth noting that, compared with the different aging prediction of each model, no matter which model, the prediction error of the short aging is lower than that of the longer aging. Therefore, the model method proposed in this paper is more suitable for the short aging prediction of the offshore visibility.

Data Availability

The data of this research work can be found through open data set. The ECMWF model data were obtained from the ECMWF website (https://www.ecmwf.int./en/forecasts/datasets/browse-reanalysis-datasets). The daily visibility data and the CUACE model data were provided by the National Meteorological Center.

Conflicts of Interest

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

References

[1] G. Fu, X. Li, and N. Wei, "Atmospheric visibility study," *Periodical of Ocean University of China*, vol. 39, no. 5, pp. 855–862, 2009.

[2] A. D. Kappos, P. Bruckmann, T. Eikmann et al., "Health effects of particles in ambient air," *International Journal of Hygiene and Environmental Health*, vol. 207, no. 4, pp. 399–407, 2004.

[3] Q. Weihong, C. L. Jeremy, C. Youli et al., "Applying anomaly-based weather analysis to the prediction of low visibility associated with the coastal fog at Ningbo-Zhoushan port in East China," *Advances in Atmospheric Sciences*, vol. 36, no. 10, pp. 1060–1077, 2019.

[4] P. Li, S. Wang, K. Shang, B. Li, H. Zhu, and S. Zeng, "Visibility forecast in Beijing through artificial neural network based on hierarchical classification method," *Journal of Lanzhou University (Natural Sciences)*, vol. 48, no. 3, pp. 52–57, 2012.

[5] D. Xia, Z. Wu, H. Tan, Z. Yuan, L. Chen, and P. Huang, "Analysis and correction of visibility measured by automatic observing system in Guangdong," *Meteorological Science and Technology*, vol. 42, no. 1, pp. 68–72, 2014.

[6] F. Xia and C. Li, "Fog weather forecast experiments of Shandong province based three visibility schemes," *Journal of Meteorology and Environment*, vol. 34, no. 3, pp. 48–57, 2018.

[7] G. A. Grell, S. E. Peckham, R. Schmitz et al., "Fully coupled "online" chemistry within the WRF model," *Atmospheric Environment*, vol. 39, no. 37, pp. 6957–6975, 2005.

[8] K. Zhu, H. Zhang, and L. Zhang, "Multi-scale numerical modeling system of atmospheric environment and its application," *Journal of Agricultural Catastrophology*, vol. 4, no. 8, pp. 38–41, 2014.

[9] X. Deng, T. Deng, D. Wu, D. Jiang, H. Tan, and F. Li, "The numerical forecast model system of air quality and visibility in the Pearl River Delta," *Guangdong Meteorology*, vol. 32, no. 4, pp. 18–22, 2010.

[10] H. Liu, X. Rao, H. Zhang, M. Li, and Z. Zhang, "Comparative verification and analysis of environmental meteorology operational numerical prediction models in China," *Journal of Meteorology and Environment*, vol. 33, no. 5, pp. 17–24, 2017.

[11] X. Chen, Y. Wang, and X. Ren, "Simulation and evaluation of haze days in jiangsu province based on WRF/CMAQ in winter 2014," *Jounarl of Nanjing University (Natural Science)*, vol. 52, no. 6, pp. 961–976, 2016.

[12] Y. Wang, W. Zhao, N. Xing, Z. Fu, and H. Li, "Visibility forecast based on RMAPS-CHEM products in Beijing," *Meterology Monthly*, vol. 46, no. 3, pp. 403–411, 2020.

[13] Z. Kang, H. Gui, C. Hua et al., "China's national environmen-meteorological services and their developmental trend," *Advances in Meteorological Science and Technology*, vol. 6, no. 2, pp. 64–69, 2016.

[14] M. Li, Z. Zhang, S. Li, X. Yu, and C. Jv, "Verification of CUACE air quality forecast in urumqi," *Desert and Oasis Meteorology*, vol. 8, no. 5, pp. 63–68, 2014.

[15] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," in *Proceedings of the 9th International Conference on Artificial Neural Networks: ICANN*, pp. 850–855, Edinburgh, UK, September 1999.

[16] A. T. Müller, J. A. Hiss, and G. Schneider, "Recurrent neural network model for constructive peptide design," *Journal of Chemical Information and Modeling*, vol. 58, no. 2, pp. 472–479, 2018.

[17] L. Jiang, R. Hu, X. Wang, W. Tu, and M. Zhang, "Nonlinear prediction with deep recurrent neural networks for non-blind audio bandwidth extension," *China Communications*, vol. 15, no. 1, pp. 72–85, 2018.

[18] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks," *Renewable Energy*, vol. 133, pp. 422–432, 2019.

[19] Y. A. Farha and J. Gall, "MS-TCN: multi-Stage temporal convolutional network for action segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3575–3584, Long Beach, CA, USA, June 2019.

[20] C. Zhou, Z. Wu, and C. Liu, "A study on quality prediction for smart manufacturing based on the optimized BP-ada Boost model," in *Proceedings of the 2019 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering (SMILE)*, pp. 1–3, Hangzhou, China, January 2019.

[21] R. Wang, Y. Dai, C. Han, K. Xu, and L. Dong, "Application of DPO—BP in Strength Prediction of Concrete," in *Proceedingds of the IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1003–1006, Beijing, China, March 2017.

[22] J. Yao, S. M. Raffuse, M. Brauer, G. J. Williamson, D. M. J. S. Bowman, and F. H. Johnston, "Predicting the minimum height of forest fire smoke within the atmosphere using machine learning and data from the CALIPSO satellite," *Remote Sensing of Environment*, vol. 206, pp. 98–106, 2018.

[23] B. Cheng, L. Liu, Z. Qi, and H. Yang, "Prediction of continuous B-cell epitopes using long short term memory networks," in *Proceedings of the 2018 6th International Conference on Bioinformatics and Computational Biology (ICBCB)*, pp. 55–59, Chengdu, China, March 2018.

[24] Z. Liu, K. Cheng, H. Li, G. Cao, D. Wu, and Y. Shi, "Exploring the potential relationship between indoor air quality and the concentration of airborne culturable fungi: a combined experimental and neural network modeling study," *Environmental Science and Pollution Research*, vol. 25, no. 4, pp. 3510–3517, 2018.

[25] Y. Dai, Z. Lu, H. Zhang, T. Zhan, J. Lu, and P. Wang, "A correction method of environmental meteorological model based on long-short-term memory neural network," *Earth and Space Science*, vol. 6, no. 11, pp. 2214–2226, 2019.

[26] X. Song, J. Huang, and D. Song, "Air quality prediction based on LSTM-Kalman model," in *Proceedings of the IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 695–699, Chongqing, China, March 2019.

[27] W. Yao, P. Huang, and Z. Jia, "Multidimensional LSTM networks to predict wind speed," in *Proceedings of the 37th Chinese Control Conference (CCC)*, pp. 7493–7497, Wuhan, China, September 2018.

[28] X. Feng, Q. Li, Y. Zhu et al., "Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation," *Atmospheric Environment*, vol. 107, no. 4, pp. 118–128, 2015.

[29] H. Feng, Y. Chen, Y. Cheng, and K. Luo, "Research on low visibility weather forecast method for Shuangliu Airport," *Journal of Applied Meteorology Science*, vol. 17, no. 1, pp. 94–99, 2006.

[30] Z. Jiachen, F. Deng, Y. Cai, and J. Chen, "Long short-term memory-fully connected (LSTM-FC) neural network for PM2.5 concentration prediction," *Chemosphere*, vol. 220, pp. 486–492, 2019.

[31] T. Li, M. Hua, and X. Wu, "A hybrid CNN-LSTM model for forecasting particulate matter (PM2.5)," *IEEE Access*, vol. 8, pp. 26933–26940, 2020.

[32] H. Hu, H. Zhang, and Z. B. X. Chao, "Application analysis of neural network method in visibility forecast of coastal cities around Bohai Sea," *Journal of the Meteorological Sciences*, vol. 38, no. 6, pp. 798–805, 2018.

[33] R. Gao, X. Li, Z. Ren, and J. Wang, "Research on forecast model of Qingdao coastal sea fog decision tree," *Marine Forecasts*, vol. 33, no. 4, pp. 80–87, 2016.

[34] M. Talo, U. B. Baloglu, Ö. Yıldırım, and U. Rajendra Acharya, "Application of deep transfer learning for automated brain abnormality classification using MR images," *Cognitive Systems Research*, vol. 54, pp. 176–188, 2019.

[35] I. Ntinou, E. Sanchez, A. Bulat, M. Valstar, and G. Tzimiropoulos, "A transfer learning approach to heatmap regression for action unit intensity estimation," 2004, http://arxiv.org/abs/06657.

[36] C. Fan, Y. Sun, F. Xiao et al., "Statistical investigations of transfer learning-based methodology for short-term building energy predictions," *Applied Energy*, vol. 262, Article ID 114499, 2020.

[37] C. Raffel, N. Shazeer, A. Roberts et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, http://arxiv.org/abs/10683.

[38] W. Guo, C. Xu, Q. Xiao, and M. Li, "Bn parameter learning algorithm based on dynamic weighted transfer learning," *Application Research of Computers*, vol. 38, no. 1, pp. 1–6, 2019.

[39] X. U. Feng, X. Tian-Zhu, W. Hui, and L. Ke-Xiu, "Comparative analysis on different data of air-sea temperature difference and analysis on variation characteristics for south China sea in the past 35 years," *Journal of Tropical Meteorology*, vol. 23, no. 3, pp. 292–301, 2017.

[40] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, http://arxiv.org/abs/1803.

[41] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[42] W. Jiang, Guo, and C. Wang, "Temporal and spatial characteristics of visibility in Beijing from 2007 to 2015," *Journal of Meteorology and Environment*, vol. 35, no. 1, pp. 45–52, 2019.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8848363, 13 pages
https://doi.org/10.1155/2020/8848363

Research Article

Learning-Based Lane-Change Behaviour Detection for Intelligent and Connected Vehicles

Luyao Du,[1] **Wei Chen**,[1] **Zhonghui Pei**,[2] **Hongjiang Zheng**,[3,4] **Shuaizhi Fu**,[1] **Kang Chen**,[1] **and Di Wu**[5,6]

[1]*School of Automation, Wuhan University of Technology, Wuhan 430070, China*
[2]*School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China*
[3]*Shanghai Engineering Technology Research Center for Intelligent and Connected Vehicle Terminals, Shanghai 200030, China*
[4]*Shanghai PATEO Electronic Equipment Manufacturing Co., Ltd., Shanghai 200030, China*
[5]*Key Laboratory of Environment Change and Resources Use in Beibu Gulf, Nanning Normal University, Ministry of Education, Nanning 530001, China*
[6]*GNSS Research Center of Wuhan University, Wuhan 430000, China*

Correspondence should be addressed to Di Wu; 29649243@qq.com

Received 17 June 2020; Revised 17 August 2020; Accepted 31 August 2020; Published 1 October 2020

Academic Editor: Nian Zhang

Detection of lane-change behaviour is critical to driving safety, especially on highways. In this paper, we proposed a method and designed a learning-based detection model of lane-change behaviour in highway environment, which only needs the vehicle to be equipped with velocity and direction sensors or each section of the highway to have a video camera. First, based on the Next Generation Simulation (NGSIM) Interstate 80 Freeway Dataset, we analyzed the relevant features of lane-changing behaviour and preprocessed the data and then used machine learning algorithms to select the suitable features for lane-change detection. According to the result of feature selection, we chose the lateral velocity of the vehicle as the lane-change feature and used machine learning algorithms to learn the lane-change behaviour of the vehicle to detect it. From the dataset, continuous data of 14 vehicles with frequent lane changes were selected for experimental analysis. The experimental results show that the designed KNN lane-change detection model has the best performance with detection accuracy between 89.57% and 100% on the selected dataset, which can well complete the vehicle lane-change detection task.

1. Introduction

Over the past few years, with the rapid development of artificial intelligence and communication technology, intelligent vehicles based on intelligence and networking have become a major trend in the development of the automotive industry. From the perspective of technological development, intelligent vehicles are divided into three development directions: connected vehicle (CV), autonomous vehicle (AV), and the integration of the former two, namely, connected and automated vehicle (CAV) or intelligent and connected vehicle (ICV) [1].

ICVs play an important role in improving driving safety and reducing driver burden, contribute to energy conservation and environmental protection, and improve traffic efficiency. Research shows that, in the initial stage of ICVs, advanced intelligent driving assistance technology can help reduce traffic accidents by about 30%, improve traffic efficiency by 10%, and reduce fuel consumption and emissions by 5% [2]. Entering the ultimate stage of the ICVs, that is, the fully automatic driving stage, it could avoid traffic accidents, improve traffic efficiency, and finally liberate people from boring driving tasks.

Driving behaviour detection plays a significant role in ICVs' decision-making system. During the driving of the vehicle, due to the driver's lack of attention or the obstruction of the surrounding large vehicles, it is likely that the driver will not be able to fully understand the driving

conditions of the surrounding vehicles, thus causing great safety risks. Many methods of lane-change behaviour detection have been proposed by researchers in recent years, including hidden Markov model (HMM) [3–5], multi-view convolutional neural network model (MV-CNN) [6], and vision-based deep residual neural network (RNN) [7]. Detection of lane-changing behaviour in different scenarios, including highways [8–12] and signalized intersections [13–16], has also been studied by many researchers. Steering behaviour recognition [17] and prediction [18] methods have been proposed, too. Besides, some new deep learning and machine learning methods have also been proposed in recent years. Xie et al. comprehensively modeled lane-change using deep learning approaches including deep belief network and long short-term memory [19]. Xing et al. proposed an ensemble bi-directional LSTM model for driver intention inference [20]. Gao et al. proposed a data-driven lane-change detection system using deep learning techniques [21]. Zhang et al. modeled the car following and lane-changing behaviours simultaneously using hybrid retraining constrained long short-term memory neural networks [22]. Zhao et al. proposed a new quantitative discriminant model based on deep belief networks algorithm and the classification analysis method based on support vector machine [23]. Dang and Dai established a lane-change model based on improved Bayesian network [24]. These methods, however, need prior knowledge, or the structure is complex and the real-time performance can be improved. In practical application scenarios, there is usually lack of prior knowledge of data distribution, and a simpler classification method is easier to implement.

In this paper, we proposed a method and designed a learning-based detection model of lane-change behaviour on highways, which only needs the vehicle to be equipped with velocity and direction sensors or each section of the highway to have a video camera. The main contributions of this paper can be summarized as follows:

(1) Based on the NGSIM Interstate 80 Freeway Dataset, the vehicle lane-change behaviour characteristics were analyzed and selected, and the data, including non-lane-change, single lane-change, and sequential lane-change, was preprocessed and reconstructed.

(2) Based on the analysis of the vehicle lane-change process, and considering the real-time requirements in the application of ICVs, the vehicle lane-change detection model based on K-Nearest Neighbor (KNN) is proposed and compared with extra tree (ET) and random forest (RF).

(3) Through feature selection, the lateral speed, which is combined with speed and driving direction and is easy to be collected, is excavated as a feature for lane-change detection. The continuous data of 14 vehicles with frequent lane changes were tested and analyzed experimentally and performed well with accuracy between 89.57% and 100% on lane-change behaviour detection.

The rest of the paper is organized as follows. Section 2 explains the details of the dataset. Section 3 describes the methodology of lane-change behaviour detection, including feature selection and lane-change detection method. Section 4 presents the experiments and results of lane-change behaviour detection. Section 5 concludes this paper and discusses the future work.

2. Data Processing

In order to verify the lane-change detection method, NGSIM Interstate 80 Freeway Dataset initiated by the United States Department of Transportation (US DOT) Federal Highway Administration (FHWA), which is freely available at the NGSIM web site at http://ngsim.fhwa.dot.gov, is used and processed. The dataset contains 45 minutes, divided into three periods: 4:00 p.m. to 4:15 p.m.; 5:00 p.m. to 5:15 p.m.; and 5:15 p.m. to 5:30 p.m., which represent the buildup of congestion, the transition between uncongested and congested conditions, and full congestion during the peak period, respectively [25]. As shown in Figure 1, the six-lane study area with a length of 1650 feet is divided into seven sub-areas. In each sub-area, a video detector is installed on the high-rise building near the lane, and the traffic of the sub-area is photographed and recorded.

The original dataset contains many attributes, including some attributes that are not highly relevant to the lane-change detection. In order to establish a dataset suitable for vehicle lane-change detection, the attributes in the dataset that are not highly relevant to lane-change detection were deleted, Vehicle_ID, Lane_ID, V_Length, and V_Width remained the same as those in the original dataset, LX_m, LY_m, Vel_m/s, and Acc_m/s^2 changed the unit in the original dataset from feet to meters (1 foot = 0.3048 meters), the average lateral velocity of vehicle and instantaneous lateral acceleration of vehicle were, respectively, calculated by LX_m and Acc_m/s^2 and added to the dataset, and the lane-change behaviour of the vehicle was calculated by Lane_ID, forming a new dataset. In the Lane_changing attribute, 0 means to keep the current lane, 1 denotes a single lane change to the right, −1 stands for a single lane change to the left, 2 represents sequential lane change to the right, and −2 represents sequential lane change to the left. The composition of processed data is shown in Table 1.

The instantaneous lateral acceleration of vehicle Acc_X can be calculated as

$$\text{Acc\_X}_t = \text{Acc\_m/s}_t^2 * \sin\left[\operatorname{acr\,tan}\left(\frac{\text{LX\_m}_t - \text{LX\_m}_{t-1}}{\text{LY\_m}_t - \text{LY\_m}_{t-1}}\right)\right], \tag{1}$$

where Acc_X$_t$ represents the value of Acc_X at time t, Acc_m/s$_t^2$ denotes the value of Acc_m/s^2 at time t, LX_m$_t$ and LX_m$_{t-1}$, respectively, mean the value of LX_m at times t and $t-1$, while LY_m$_t$ and LY_m$_{t-1}$ stand for the value of LY_m at times t and $t-1$, respectively.

The average lateral velocity of vehicle Vel_X can be calculated as

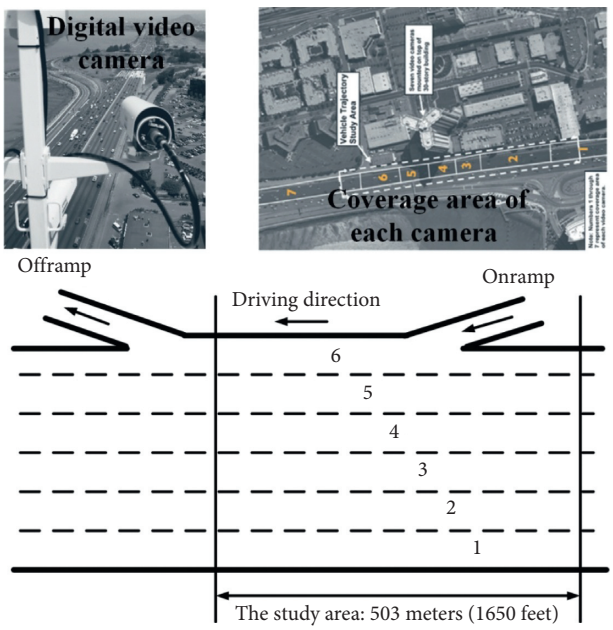


Figure 1: The collection scene description of data. The six-lane study area, which is divided into seven sub-areas, is photographed and recorded by digital video cameras.

Table 1: The composition of processed data.

Attribute label	Attribute definition
Vehicle_ID	Vehicle identification number.
LX_m	Lateral (X) coordinate of the front center of the vehicle in meter with respect to the left-most edge of the section in the direction of travel.
LY_m	Longitudinal (Y) coordinate of the front center of the vehicle in meter with respect to the entry edge of the section in the direction of travel.
V_Length	Length of vehicle in feet.
V_Width	Width of vehicle in feet.
Vel_m/s	Instantaneous velocity of vehicle in m/s.
Acc_m/s^2	Instantaneous acceleration of vehicle in m/s^2.
Acc_X	Instantaneous lateral acceleration of vehicle in m/s^2.
Vel_X	Average lateral velocity of vehicle in m/s.
Lane_ID	Current lane position of vehicle.
Lane_change	Current lane-change behaviour of vehicle.

$$\text{Vel\_X}_t = \frac{(\text{LX\_m}_t - \text{LX\_m}_{t-1})}{t}, \quad (2)$$

where Vel_X_t represents the value of Vel_X at time t, LX_m_t means the value of LX_m at time t, $\text{LX\_m}_{(t-1)}$ denotes the value of LX_m at time $t-1$, and t is the sampling period of the dataset, which is 0.1 seconds.

Current lane-change behaviour of vehicle Lane_changing can be calculated as

$$\text{Lane\_changing}_t = \text{Lane\_ID}_t - \text{Lane\_ID}_{t-1}, \quad (3)$$

where Lane_changing_t represents the value of Lane_changing at time t, Lane_ID_t means the value of Lane_ID at time t, and Lane_ID_t denotes the value of Lane_ID at time $t-1$.

3. Methodology

3.1. Feature Selection. In order to accurately detect vehicle lane changes, the relationship between various attributes and vehicle lane changes was analyzed. By analyzing the changing trends of various attributes when the vehicle changes lanes in the dataset, we found that the vehicle's lateral velocity has the most obvious correlation with the lane-change behaviour. The relationship between lateral velocity and lane change is shown in Figure 2, from which we can see that when the vehicle changes lanes, there will be a very obvious change in lateral velocity.

To further analyze and verify the relationship between each attribute and the lane-change behaviour of the vehicle, machine learning models, including KNN [26], extra trees

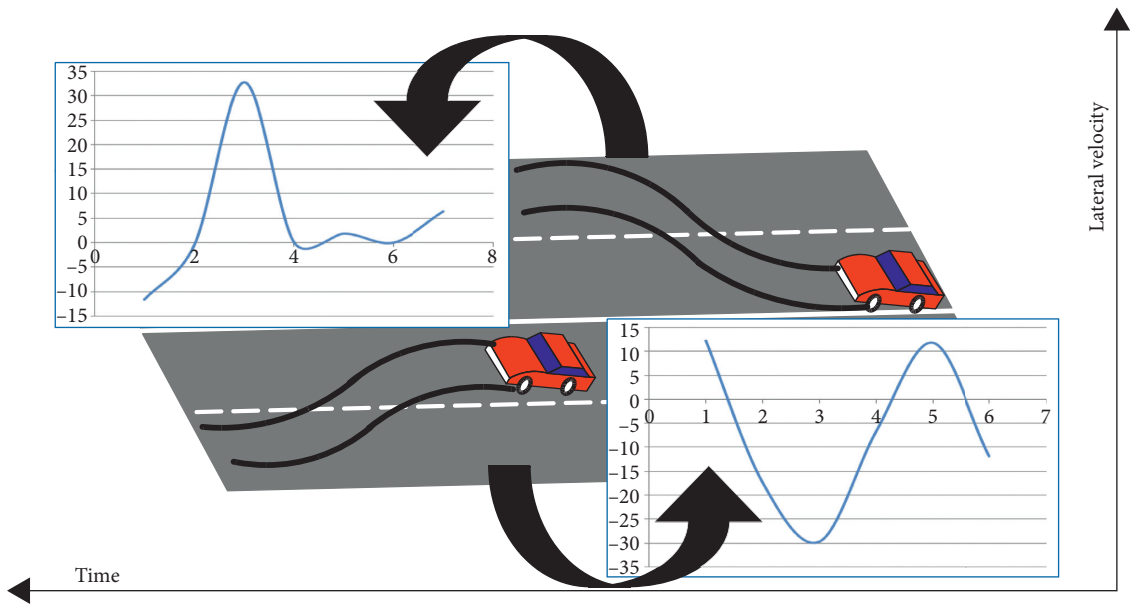


Figure 2: The relationship between lateral velocity and lane change.

[27], and random forest [28], were performed on each attribute. A total of 27287 lane-changing data were extracted from the dataset. At the same time, in order to balance the number of samples of lane-changing (the moment when the vehicle changes lanes) and non-lane-changing (the moment when the vehicle does not change lanes) data, 27287 non-lane-changing data were selected to form a feature selection dataset. The result of feature selection is shown in Table 2, from which we can see that the detection accuracy with Vel_X as the feature is significantly higher than other features, which can reach more than 90%. In the detection using Vel_X, the accuracy of KNN, extra tree, and random forest is 94.85%, 91.73%, and 92.31%, respectively; KNN has the highest accuracy.

In order to represent the contribution of each feature to the lane-change behaviour detection more intuitively, feature importance analysis, which can be applied to random forest and extra trees, was performed on the dataset. The Gini index was used to measure the feature importance, which can be defined as

$$G_m = \sum_{k=1}^{|K|} \sum_{k' \neq k} p_{mk}\, p_{mk'} = 1 - \sum_{k=1}^{|K|} p_{mk}^2, \tag{4}$$

where K means that there are K categories and p_{mk} denotes the proportion of category k in node m. VIM is used to represent the variable importance measures. The importance of feature X_j at node m, that is, the change in Gini index before and after the branch of node m, can be defined as

$$\mathrm{VIM}_{jm}^{(\mathrm{Gini})} = G_m - G_{\mathrm{l}} - G_{\mathrm{r}}, \tag{5}$$

where G_{l} and G_{r} represent the Gini index of the left and right nodes after the m branch. If the node where the feature X_j appears in the decision tree i is in the set M, then the importance of X_j in the i-th tree is

$$\mathrm{VIM}_{ij}^{(\mathrm{Gini})} = \sum_{m \in M} \mathrm{VIM}_{jm}^{(\mathrm{Gini})}. \tag{6}$$

Assuming there are n trees, then

$$\mathrm{VIM}_{j}^{(\mathrm{Gini})} = \sum_{i=1} \mathrm{VIM}_{ij}^{(\mathrm{Gini})}. \tag{7}$$

Finally, supposing there are c features, all the obtained importance scores are normalized:

$$\mathrm{VIM}_j = \frac{\mathrm{VIM}_j}{\sum_{i=1}^{c} \mathrm{VIM}_i}. \tag{8}$$

The feature ranking based on feature importance is shown in Table 3, which illustrates that Vel_X has the highest feature importance scores in both random forest and extra tree and is significantly higher than the other six features. Therefore, Vel_X can be selected as a feature of lane-change detection for ICVs.

3.2. Lane-Change Detection

3.2.1. Lane-Change Model. The data used in feature selection is discontinuous, so the learned features are relatively independent and have no relationship with the adjacent data. In the actual driving process, the data of vehicle lane-changing behaviour often only takes up a small part of the entire dataset. Therefore, in order to further analyze and establish a lane-change detection model during vehicle driving, we have selected continuous data from 14 vehicles with frequent lane changes for analysis, training, and testing.

Lane-change behaviour includes single lane change and sequential lane change. The single lane changes to the left and right are denoted as −1 and 1, while the sequential lane

Table 2: Precision result of feature selection.

Selected features	KNN	Extra trees (%)	Random forest (%)
LX_m; LY_m	60.28	70.63	74.28
V_Length; V_Width	62.38	65.26	65.29
Vel_m/s	49.14	50.26	50.67
Acc_m/s^2	43.28	43.62	43.52
Vel_X	94.85	91.73	92.31

Table 3: Features ranked based on importance.

Random forest			Extra trees		
Rank	Feature	Importance	Rank	Feature	Importance
1	Vel_X	0.712	1	Vel_X	0.788
2	Vel_m/s	0.094	2	Vel_m/s	0.061
3	Acc_m/s^2	0.052	3	Acc_m/s^2	0.042
4	LX_m	0.044	4	LX_m	0.032
5	LY_m	0.040	5	LY_m	0.030
6	V_Length	0.031	6	V_Length	0.024
7	V_Width	0.027	7	V_Width	0.022

changes to the left and right are denoted as −2 and 2, respectively. The lateral velocity of single lane change intercepted from selected data is shown in Figure 3, from which we can see that there is a significant peak/valley when the vehicle changes lanes, and the threshold of peak/valley can be learned to determine if the vehicle is changing lanes. The lateral velocity of sequential lane change intercepted from selected data is shown in Figure 4; similar to single lane change, there is also a significant peak/valley when the vehicle changes lanes. Besides, there will be a continuous peak/valley or a larger peak/valley of lateral velocity in the sequential lane change, which also can be learned to determine if the vehicle is changing lanes sequentially.

3.2.2. Detection Model. KNN is a simple classification method that can perform effective classification in the absence of prior knowledge of data distribution, and the same are the ET and RF.

RF is an algorithm that integrates multiple trees through the idea of integrated learning and its basic unit is the decision tree [29]. The construction process of the random forest model is mainly divided into four steps:

(1) First, a tree needs to be constructed. If there are N samples, there are randomly selected N samples to be replaced (randomly select one sample at a time, and then return to continue selection). The selected N samples are used to train a decision tree as the sample at the root node of the decision tree.

(2) Each sample has M attributes; when each node of the decision tree needs to be split, m attributes are randomly selected from the M attributes to satisfy the condition $m << M$. Then use some strategy such as information gain and Gini index from the m attributes to select one attribute as the split attribute of the node.

(3) Repeat step 2 until it can no longer split.

(4) Follow steps 1~3 to build a large number of decision trees to form a forest.

The ET is very similar to RF; they are both composed of many decision trees. The difference is that the RF obtains the best bifurcation attribute in a random subset, while ET obtains the bifurcation value completely randomly, so as to achieve the fork of the decision tree [30].

From the results of feature selection, we can see that when lateral velocity is used as a feature for lane-change detection, KNN has achieved good result, which is better than ET and RF. Therefore, the KNN model is designed using lateral velocity as feature for lane-change detection and the result is compared with ET and RF.

KNN makes predictions using the training dataset directly. Predictions are made for a new data point by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances.

To determine which of the K instances in the training dataset are most similar to a new input, a distance measure is used. For real-valued input variables, the Euclidean distance is used as a distance measure method. Euclidean distance is calculated as the square root of the sum of the squared differences between a point a and point b across all input attributes i.

$$\text{Euclidean distance}(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}. \quad (9)$$

When KNN is used for classification, the output can be calculated as the class with the highest frequency from the K-most similar instances. Each instance in essence votes for their class and the class with the most votes is taken as the prediction.

Class probabilities can be calculated as the normalized frequency of samples that belong to each class in the set of K most similar instances for a new data instance. For example, in a binary classification problem (class is 0 or 1),

$$p(\text{class} = 0) = \frac{\text{count}(\text{class} = 0)}{\text{count}(\text{class} = 0) + \text{count}(\text{class} = 1)}. \quad (10)$$

The KNN algorithm can be described as follows:

(1) Initialize training sets and categories

(2) Calculate the Euclidean distance between the test set sample and the training set sample

(3) Sort the training set samples in ascending order according to the Euclidean distance

(4) Select the first K training samples with the smallest Euclidean distance and count their frequency in each category

(5) The category with the highest return frequency, that is, the test set sample, belongs to this category

Table 4 shows the step of KNN algorithm, in which the list I_z of its nearest neighbors is determined by calculating the similarity distance between the training object $(x, y) \in I$

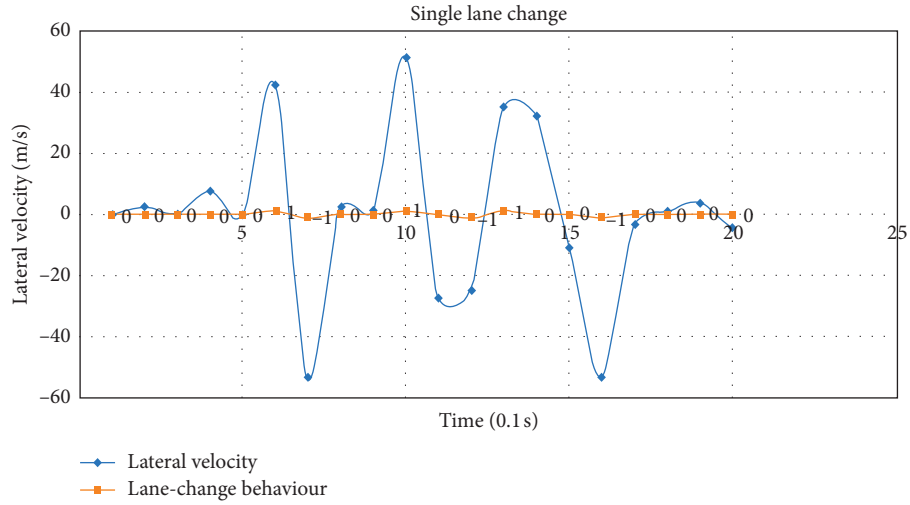


Figure 3: The lateral velocity of single lane change.

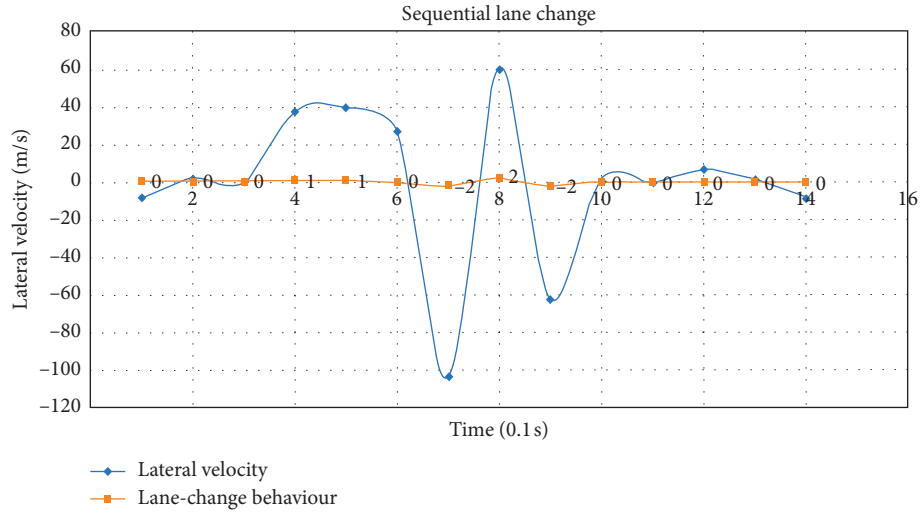


Figure 4: The lateral velocity of sequential lane change.

Table 4: Description of KNN algorithm.

Input: Training object $(x, y) \in I$ and test object $z = (\hat{x}, \hat{y})$
Processing: Compute distance $d = (\hat{x}, \hat{y})$ between z and every object $(x, y) \in I$. Select $I_z \subseteq I$, the set of k closest training objects to z.
Output: $\hat{y} = \arg_v \max \sum_{(x_i, y_i) \in I_z} F(v = y_i)$

and the test object $z = (\hat{x}, \hat{y})$, where x represents the training object, y represents the class to which it belongs, and $\hat{x}$ and $\hat{y}$ represent the test object and the class to which it belongs.

4. Experimental Results

4.1. Evaluation Indicators. When performing machine learning, the confusion matrix of prediction results can be described as shown in Table 5.

The number of pairs of samples divided by the number of all samples is the accuracy (ACC), which can be defined as

$$\text{ACC} = \frac{\text{TP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{11}$$

Generally, the higher the accuracy, the better the classifier. However, in the case of imbalance between positive and negative samples, the accuracy is a big flaw as an evaluation indicator. It is not scientific and comprehensive to evaluate a model based on accuracy alone.

To evaluate the performance of machine learning more scientifically and comprehensively, precision (P), recall (R), and $F1$ score can be used.

Table 5: Confusion matrix of prediction results.

Prediction truth	Positive	Negative
True	True positive (TP)	True negative (TN)
False	False positive (FP)	False negative (FN)

The precision, which represents the proportion of positive examples that are actually classified as positive examples, can be defined as

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{12}$$

The recall, which measures how many positive examples are classified as positive examples, can be defined as

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{13}$$

P and R indicators sometimes conflict, so they need to be considered comprehensively. The most common method is F-Measure, which is the weighted harmonic average of P and R, and can be defined as

$$F = \frac{\left(\alpha^2 + 1\right)P * R}{\alpha^2 (P + R)}. \tag{14}$$

When the parameter $\alpha = 1$, it is $F1$:

$$F1 = \frac{2 * P * R}{P + R}. \tag{15}$$

$F1$ combines the results of P and R. When $F1$ is higher, it can indicate that the model is more effective.

In addition, receiver operating characteristic (ROC) curve, in which the abscissa is False Positive Rate (FPR) and the ordinate is True Positive Rate (TPR), is also an important evaluation indicator. The definition of TPR is the same with P, and the FPR can be defined as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FN}}. \tag{16}$$

The area under the ROC curve is called AUC. The prediction effect of a classification model can be evaluated based on the AUC value; the larger the AUC value, the better the performance of the model.

4.2. Experiments and Analysis. The model is built on vscode using python and uses the scikit-learn framework. The experiments were performed on a server with a single-core CPU, 2.6 GHz, 2G memory, and Ubuntu 18.04.

KNN was performed on the selected dataset first. In order to choose the most appropriate number of neighbors, we trained and tested different numbers of neighbors on the dataset consisting of the data of all 14 vehicles and obtained their accuracy, respectively. The result of varying number of KNN neighbors is shown in Figure 5, from which we can see that the accuracy of the training set decreases as the number of neighbors increases, and at the same time the accuracy of the test set increases as the number of neighbors increases. When the number of neighbors increases to 9, the accuracy of both the training and test sets remains stable. Therefore, 9 is appropriate to be determined as the number of neighbors.

After determining the number of neighbors, KNN model was designed and performed on the dataset consisting of the data of all 14 vehicles, compared with ET and RF. The ROC of designed models including KNN, ET, and RF is shown in Figure 6 and the AUC values of the three models are shown in Table 6. Obviously, from the ROC curves, the performance of KNN is better than RF and obviously better than ET. It can be seen more clearly in Table 6 that the AUC value of KNN is 97.73%, while the AUC values of ET and RF are 92.55% and 96.69%, respectively, showing that the performance of KNN is the best in these three models.

After experimental testing on the dataset of all 14 vehicles, in order to analyze the effect of continuous lane-change detection of vehicles in real scenes, the designed KNN model was used to perform experiments on the respective datasets of the 14 vehicles and compared with ET and RF.

The detailed sample sizes of lane-change behaviour on 14 selected vehicles are shown in Table 7. In the dataset, we can see that, in the real scene, the number of lane-keep samples is generally larger than the number of left and right lane-change samples. Among the 14 vehicles, only the total number of the lane changes to left (LCL) and the lane changes to right (LCR) of the vehicle numbered 2791 is greater than the number of lane-keep (LK), and the number of each item is still less than the number of LK. Besides, there are continuous lane changes in vehicles numbered 2795 and 2825. A single lane change to the left and right is recorded as LCL-1 and LCR-1, while a continuous lane change to the left and right is recorded as LCL-2 and LCR-2, respectively.

The dataset combining all 14 selected vehicles is divided into training and testing sets according to a ratio of 0.75 to 0.25. The confusion matrix of the detection results is shown in Figure 7, which illustrates that the KNN model has the highest detection accuracy, followed by RF, and ET has the lowest detection accuracy. In addition, detection errors mainly occur between the non-lane-change and the single lane-change behaviour, the probability of false detection between sequential lane-change behaviour and other behaviours is small, and the probability of false detection between left lane-change and right lane-change behaviour is also small. It is worth noting that, in the KNN and RF models, there is no misdetection between the left lane-change, right lane-change, and non-lane-change behaviour.

The experimental results of lane-change detection performed on 14 selected vehicles are shown in Table 8, in which the evaluation indicator mACC denotes mean accuracy of the detection model on all lane-change behaviours. From the experimental results, KNN performed best in the lane-change detection results of all 14 vehicles, while in the lane-change detection results of 14 vehicles, ET performed better than RF in 4 vehicles, RF performed better than ET in 7 vehicles, and ET and RF performed the same in the remaining 3 vehicles. KNN's lane-change detection accuracy ranges from 89.57% to 100%, ET's

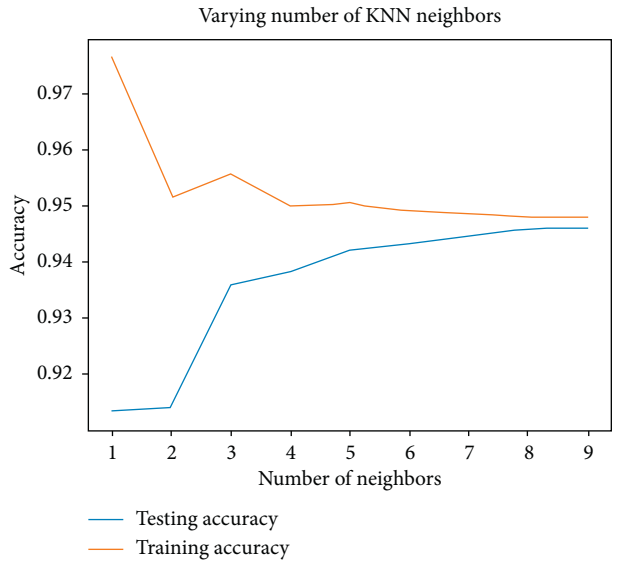


Figure 5: Result of varying number of KNN neighbors.

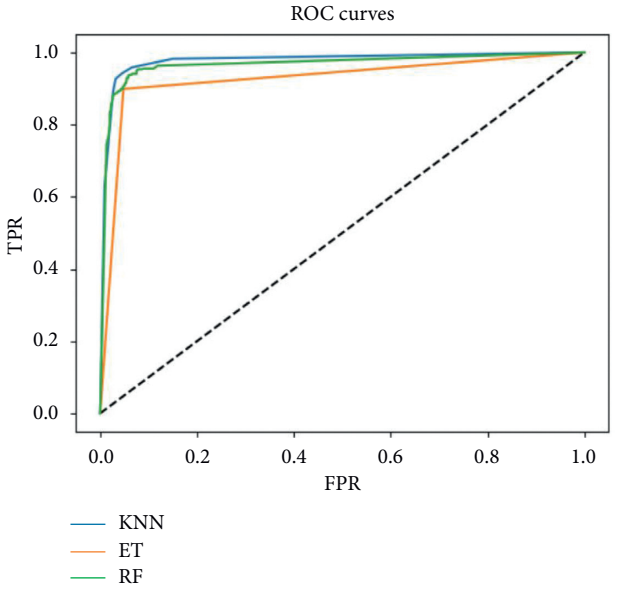


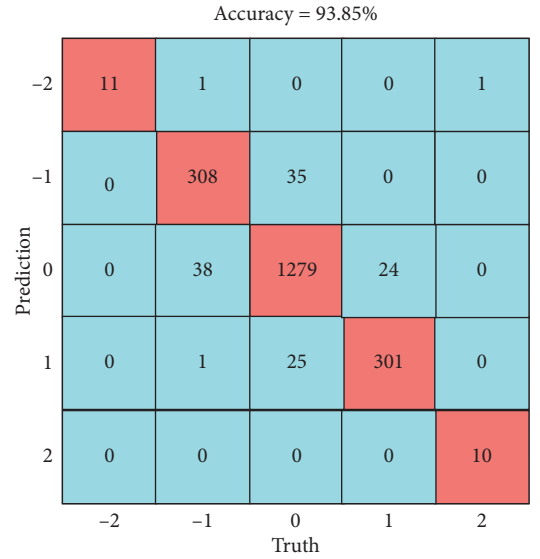
Figure 6: The ROC curves of designed models.

Table 6: AUC values of designed models.

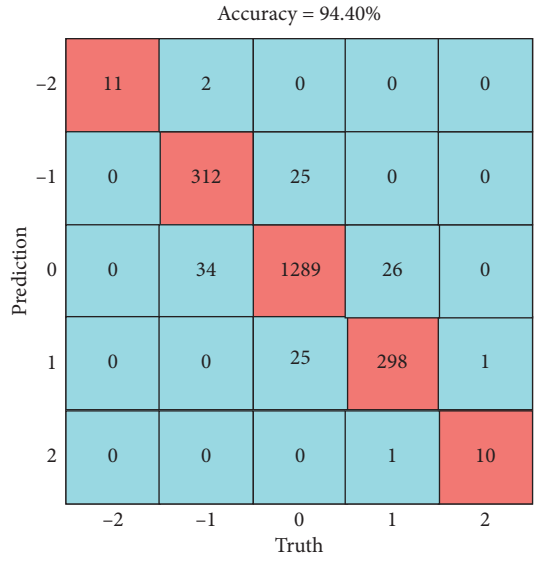
	KNN (%)	ET (%)	RF (%)
AUC values	97.73	92.55	96.69

Table 7: Sample size of lane-change behaviour on selected vehicles.

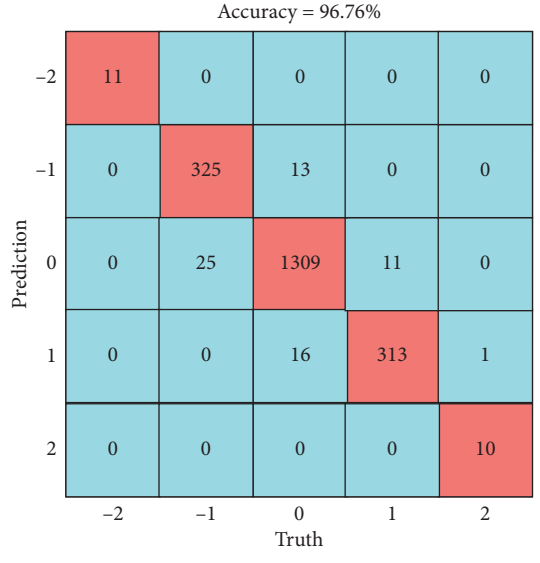
ID	Behaviour	Sample size
3365	LCL	131
	LK	444
	LCR	133
3362	LCL	79
	LK	299
	LCR	79
2826	LCL	39
	LK	123
	LCR	39
2804	LCL	45
	LK	267
	LCR	45
2795	LCL-2	36
	LCL-1	46
	LK	613
	LCR-1	48
	LCR-2	35
2782	LCL	152
	LK	935
	LCR	151
2778	LCL	91
	LK	620
	LCR	91
3363	LCL	127
	LK	334
	LCR	127
3063	LCL	23
	LK	99
	LCR	22
2791	LCL	207
	LK	379
	LCR	206
2800	LCL	100
	LK	300
	LCR	99
2825	LCL-2	12
	LCL-1	13
	LK	123
	LCR-1	15
	LCR-2	11
2779	LCL	134
	LK	526
	LCR	135
2774	LCL	100
	LK	379
	LCR	102



(a)



(b)



(c)

Figure 7: The confusion matrix of detection results: (a) ET, (b) RF, and (c) KNN.

Table 8: Experimental results of lane-change detection on selected vehicles.

ID	Model	Behaviour	P (%)	R (%)	$F1$ (%)	mACC (%)
3365	ET	LCL	91	94	93	97.18
		LK	98	97	98	
		LCR	100	100	100	
	RF	LCL	91	94	93	96.61
		LK	97	97	97	
		LCR	100	97	99	
	KNN	LCL	97	97	97	98.87
		LK	99	99	99	
		LCR	100	100	100	
3362	ET	LCL	68	89	77	84.35
		LK	89	84	86	
		LCR	88	82	85	
	RF	LCL	71	89	79	86.09
		LK	91	85	88	
		LCR	89	86	87	
	KNN	LCL	76	100	86	89.57
		LK	97	85	91	
		LCR	87	93	90	
2826	ET	LCL	89	89	89	94.12
		LK	94	97	96	
		LCR	100	86	92	
	RF	LCL	80	89	84	90.20
		LK	92	94	93	
		LCR	100	71	83	
	KNN	LCL	89	89	89	96.08
		LK	97	97	97	
		LCR	100	100	100	
2804	ET	LCL	73	80	76	93.33
		LK	97	94	96	
		LCR	90	100	95	
	RF	LCL	89	80	84	94.44
		LK	97	96	96	
		LCR	82	100	90	
	KNN	LCL	90	90	90	95.55
		LK	97	97	97	
		LCR	89	89	89	
2795	ET	LCL-2	91	100	95	94.36
		LCL-1	67	77	71	
		LK	99	95	97	
		LCR-1	81	100	90	
		LCR-2	100	83	91	
	RF	LCL-2	91	100	95	96.41
		LCL-1	91	77	83	
		LK	99	98	98	
		LCR-1	81	100	90	
		LCR-2	100	83	91	
	KNN	LCL-2	100	100	100	98.46
		LCL-1	100	77	87	
		LK	98	100	99	
		LCR-1	100	100	100	
		LCR-2	100	100	100	

Table 8: Continued.

ID	Model	Behaviour	*P* (%)	*R* (%)	*F*1 (%)	mACC (%)
		LCL	100	98	99	
	ET	LK	99	100	99	99.03
		LCR	100	95	97	
		LCL	100	98	99	
2782	RF	LK	98	100	99	98.71
		LCR	100	92	96	
		LCL	100	98	99	
	KNN	LK	100	100	100	99.68
		LCR	100	100	100	
		LCL	96	100	98	
	ET	LK	100	97	99	98.01
		LCR	88	100	94	
		LCL	96	100	98	
2778	RF	LK	100	97	99	98.01
		LCR	88	100	94	
		LCL	100	100	100	
	KNN	LK	99	100	99	99.00
		LCR	100	91	95	
		LCL	87	77	82	
	ET	LK	87	90	89	87.07
		LCR	86	89	88	
		LCL	90	74	81	
3363	RF	LK	86	90	88	86.39
		LCR	83	89	86	
		LCL	96	77	86	
	KNN	LK	90	95	92	91.16
		LCR	90	96	93	
		LCL	50	33	40	
	ET	LK	84	96	90	83.33
		LCR	100	40	57	
		LCL	50	33	40	
3063	RF	LK	87	96	92	86.11
		LCR	100	60	75	
		LCL	100	67	80	
	KNN	LK	97	100	98	97.22
		LCR	100	100	100	
		LCL	100	77	87	
	ET	LK	82	98	89	88.89
		LCR	95	87	91	
		LCL	100	84	91	
2791	RF	LK	84	97	90	89.90
		LCR	93	84	88	
		LCL	96	85	91	
	KNN	LK	89	96	92	92.42
		LCR	96	96	96	
		LCL	88	93	90	
	ET	LK	94	85	90	88.00
		LCR	72	90	80	
		LCL	91	97	94	
2800	RF	LK	96	91	93	92.00
		LCR	82	90	86	
		LCL	97	97	97	
	KNN	LK	97	95	96	95.20
		LCR	86	95	90	

Table 8: Continued.

ID	Model	Behaviour	P (%)	R (%)	$F1$ (%)	mACC (%)
		LCL-2	50	100	67	
		LCL-1	50	67	57	
	ET	LK	100	91	95	90.91
		LCR-1	67	100	80	
		LCR-2	100	100	100	
		LCL-2	50	100	67	
		LCL-1	67	67	67	
2825	RF	LK	100	94	97	90.91
		LCR-1	50	50	50	
		LCR-2	83	100	91	
		LCL-2	100	100	100	
		LCL-1	100	100	100	
	KNN	LK	100	97	98	95.45
		LCR-1	50	100	67	
		LCR-2	100	80	89	
		LCL	98	100	99	
	ET	LK	100	99	100	99.50
		LCR	100	100	100	
		LCL	98	100	99	
2779	RF	LK	100	99	100	99.50
		LCR	100	100	100	
		LCL	100	100	100	
	KNN	LK	100	100	100	100
		LCR	100	100	100	
		LCL	91	91	91	
	ET	LK	96	96	96	94.52
		LCR	94	94	94	
		LCL	95	91	93	
2774	RF	LK	96	97	96	95.21
		LCR	94	94	94	
		LCL	100	100	100	
	KNN	LK	98	98	98	97.26
		LCR	94	94	94	

lane-change detection accuracy ranges from 83.33% to 99.50%, while RF's lane-change detection accuracy ranges from 86.09% to 99.50%. Besides, combined with the sample sizes, in the vehicles with the numbers of 3063 and 2825, the number of samples of the lane-change left and the lane-change right is small, and the detection results are relatively poor, indicating that too few samples will affect the accuracy of the lane-change detection.

5. Conclusions

This paper proposed a lane-change detection method for intelligent and connected vehicles. Based on the feature selection of vehicle lane-change behaviour, the detection model based on machine learning was designed, and the effect verification and comparison were performed on the selected dataset. The dataset based on NGSIM Interstate 80 Freeway Dataset was processed for lane-change detection first. After that, feature selection for lane-change detection was performed on the processed dataset, and the lateral velocity was selected as the feature for lane-change detection. Then, the lane-change model was analyzed based on the real data in the processed dataset and the detection model was designed. Finally, the number of KNN neighbors was determined based on experiment, and the performance of KNN, ET, and RF was analyzed by the evaluation indicators. From the experimental results, the designed KNN model performed best in all datasets of the selected 14 vehicles, with detection accuracy ranging from 89.57% to 100%, indicating that it can well complete the task of lane-change behaviour detection for ICVs.

As for future work, the lane-changing scene can be extended by the measured data from the vehicle sensors to establish a more widely adaptable dataset, and the detection model can be further optimized and then implemented on embedded hardware to achieve a lane-change real-time detection system for ICVs.

Data Availability

The related data are available online at https://github.com/WHUT-DLY/Processed-data-based-on-the-Next-Generation-Simulation-NGSIM-Interstate-80-Freeway-Dataset.

Conflicts of Interest

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

Acknowledgments

The research work was supported in part by the National Key R&D Program of China under Grant 2018YFB0105205, in part by Major Technological Innovation Project of Hubei Province under Grant 2019AAA025, and in part by the Fundamental Research Funds for the Central Universities (WUT: 2019-JL-023).

References

[1] K. Q. Li, Y. F. Dai, S. B. Li, and M. Y. Bian, "State-of-the-art and technical trends of intelligent and connected vehicles," *Journal of Automotive Safety and Energy*, vol. 8, no. 1, pp. 1–14, 2017.

[2] S. A. E. China-, "Technology Roadmap for Energy Saving and New Energy Vehicles," Mechanical Industry Press, Beijing, China, 2016, in Chinese.

[3] R. Hamada, T. Kubo, K. Ikeda et al., "Modeling and prediction of driving behaviors using a nonparametric Bayesian method with AR models," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 131–138, 2016.

[4] Q. Deng and D. Soffker, "Improved driving behaviours prediction based on fuzzy logic-hidden Markov model (FL-HMM)," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2003–2008, Suzhou, China, June 2018.

[5] E. Yurtsever, S. Yamazaki, C. Miyajima et al., "Integrating driving behavior and traffic context through signal symbolization for data reduction and risky lane change detection," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 242–253, 2018.

[6] Y. Zhang, J. Li, Y. Guo, C. Xu, J. Bao, and Y. Song, "Vehicle driving behavior recognition based on multi-view convolutional neural network with joint data augmentation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4223–4234, 2019.

[7] Z. S. Wei, C. Wang, P. Hao, and M. J. Barth, "Vision-based lane-changing behaviour detection using deep residual neural network," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3108–3113, Auckland, New Zealand, October 2019.

[8] X. Y. Xu, J. D. Yu, Y. M. Zhu, Z. C. Wu, J. D. Li, and M. L. Li, "Leveraging smartphones for vehicle lane-level localization on highways," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1894–1907, 2018.

[9] T. Rehder, A. Koenig, M. Goehl, L. Louis, and D. Schramm, "Lane change intention awareness for assisted and automated driving on highways," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 265–276, 2019.

[10] N. Motamedidehkordi, S. Amini, S. Hoffmann, F. Busch, and M. R. Fitriyanti, "Modeling tactical lane-change behaviour for automated vehicles: a supervised machine learning approach," in *Proceedings of the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 268–273, Naples, Italy, June 2017.

[11] D. Augustin, M. Hofmann, and U. Konigorski, "Prediction of highway lane changes based on prototype trajectories," *Forschung im Ingenieurwesen*, vol. 83, no. 2, pp. 149–161, 2019.

[12] S. Seelam, M. Arpan, V. Praveen, and K. G. Naga, "Simulation of traffic flow to analyze lane changes on multi-lane highways under non-lane discipline," *Periodica Polytechnica: Transportation Engineering*, vol. 48, no. 2, pp. 109–116, 2018.

[13] T. Wang, L. J. Xu, G. J. Chen, and W. Zhao, "A guidence method for lane change detection at signalized intersections in connected vehicle environment," in *Proceedings of the 2019 5th International Conference on Transportation Information and Safety*, pp. 32–38, Liverpool, UK, July 2019.

[14] Q. Gao, J. Zhang, Z. Sheng, and L. Dong, "A lane-changing BML model considering the influence of both lane information and turn signals," *Lixue Xuebao/Chinese Journal of Theoretical and Applied Mechanics*, vol. 52, no. 1, pp. 283–291, 2020.

[15] Y. Xing, J. Wang, W. Liu, L. Sun, and F. Chong, "Study on the characteristics of vehicle lane-changing in the intersection," *Advances in Intelligent Systems and Computing*, vol. 890, pp. 371–381, 2019.

[16] F. L. Wei, Z. Y. Wang, and J. Liu, "Exploring factors contributing to lane changes during left turns on quadruple left-turn lanes at signalized intersections," *Advances in Mechanical Engineering*, vol. 9, no. 5, pp. 171–179, 2018.

[17] Z. Ouyang, J. Niu, and M. Guizani, "Improved vehicle steering pattern recognition by using selected sensor data," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1383–1396, 2018.

[18] L. X. Li and P. H. Li, "Analysis of drivers steering behaviour for lane change prediction," in *Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pp. 71–75, Hangzhou, China, August 2019.

[19] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 41–60, 2019.

[20] Y. Xing, C. Lv, H. J. Wang, D. P. Cao, and E. Velenis, "An ensemble deep learning approach for driver lane change intention inference," *Transportation Research: Part C*, vol. 115, 2020.

[21] J. Gao, Y. L. Murphey, J. G. Yi, and H. H. Zhu, "A data-driven lane-changing behavior detection system based on sequence learning," *Transportmetrica B: Transport Dynamics*, 2020.

[22] X. Zhang, J. Sun, X. Qi, and J. Sun, "Simultaneous modeling of car-following and lane-changing behaviors using deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 287–304, 2019.

[23] W. Zhao, L. J. Xu, B. Ran, and J. Z. Wang, "Dangerous lane-change detecting model on highway based on deep learning DBN algorithm," *Journal of Southeast University (Natural Science Edition)*, vol. 47, no. 4, pp. 832–838, 2017.

[24] T. Dang and F. Dai, "Vehicle lane change model based on improved bayesian network structure learning," *International Journal of Intelligent Technologies and Applied Statistics*, vol. 11, no. 4, pp. 255–270, 2018.

[25] Federal Highway Administration (FHWA), "Next Generation Simulation (NGSIM) Interstate 80 Freeway Dataset," U.S. Department of Transportation Intelligent Transportation Systems Joint Program Office (JPO), Washington, D.C., USA, 2016.

[26] Z. Yu, H. Chen, J. Liuxs, J. You, H. Leung, and G. Han, "Hybrid," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1263–1275, 2016.

[27] V. John, Z. Liu, C. Z. Guo, S. Mita, and K. Kidono, "Real-time lane estimation using deep features and extra trees regression," *Image and Video Technology*, vol. 9431, pp. 721–733, 2016.

[28] G. Robin, J. M. Poggi, T. M. Christine, and V. V. Nathalie, "Random forests for big data," *Big Data Research*, vol. 9, pp. 28–46, 2017.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8823906, 9 pages
https://doi.org/10.1155/2020/8823906

Review Article

Extracting Parallel Sentences from Nonparallel Corpora Using Parallel Hierarchical Attention Network

Shaolin Zhu,[1] Yong Yang,[2] and Chun Xu[3]

[1]*Zhengzhou University of Light Industry, Zhengzhou 453000, China*
[2]*Xinjiang Normal University, Urmqi 830011, China*
[3]*Xinjiang University of Finance and Economics, Urmqi 830011, China*

Correspondence should be addressed to Yong Yang; 68523593@qq.com

Received 5 June 2020; Accepted 31 July 2020; Published 1 September 2020

Academic Editor: Nian Zhang

Collecting parallel sentences from nonparallel data is a long-standing natural language processing research problem. In particular, parallel training sentences are very important for the quality of machine translation systems. While many existing methods have shown encouraging results, they cannot learn various alignment weights in parallel sentences. To address this issue, we propose a novel parallel hierarchical attention neural network which encodes monolingual sentences versus bilingual sentences and construct a classifier to extract parallel sentences. In particular, our attention mechanism structure can learn different alignment weights of words in parallel sentences. Experimental results show that our model can obtain state-of-the-art performance on the English-French, English-German, and English-Chinese dataset of BUCC 2017 shared task about parallel sentences' extraction.

1. Introduction

Parallel sentences are a very important linguistic resource which comprises much text in the parallel translation of different languages. A large parallel corpus is crucial to train machine translation systems which can produce good quality translations. As is well known, the major bottleneck of statistical machine translation (SMT) and neural machine translation (NMT) is the scarceness of parallel sentences in many language pairs [1–3]. With an increasing amount of comparable corpora on the World Wide Web, a potential solution that alleviates the parallel data sparsity is to extract parallel sentences from comparable corpora. Previous research has shown that this bottleneck can be relieved by extracting parallel sentences from comparable corpora [4–11].

As collecting parallel sentences is important for improving the quality of machine translation systems, many works try to mine parallel sentences from comparable corpora in the last two decades. Their success has a great contribution to the development of this research. Traditional systems developed to extract parallel sentences from comparable corpora typically rely on multiple features or metadata from comparable corpora structure. Bouamor and Sajjad [12] proposed to use a hybrid approach pairing multilingual sentence-level embedding and supervised classifier to identify parallel sentence pairs. They used features such as source-target punctuation marks features and morphosyntactic features to build a support vector machine binary classifier. Although feature engineering is an effective strategy to filter parallel sentences, it usually suffers from the language diversity issue. For example, the named entity is an important feature to measure source-target candidate parallel sentences. However, the named entity has various processes in different languages. For English, CoreNLP (https://stanfordnlp.github.io/CoreNLP/) can be implemented to extract English persons, locations, and organizations, while there are no open-source tools to deal with other lingual named entities such as Uyghur. To address those issues, many methods extracted parallel sentences without feature engineering. More recent approaches used deep learning, such as convolutional neural networks [13] and recurrent neural networks based on long short-term memory (LSTM) [1, 14, 15] to learn an end-to-end network classifier to filter parallel sentences.

Although mining parallel sentences using neural-network-based approaches has been quite effective, we use the better representations that can be obtained by incorporating knowledge of context information in the model of sentence architecture in this paper. As we all know, not all parts of a sentence are equally relevant for representing parallel sentences (as an example in Figure 1, unmarked words do not affect detecting parallel sentences). That is, different words have various important weights for detecting parallel sentences.

To address those issues, this paper proposes a parallel hierarchical attention network (PHAN) that learns parallel sentence representations. The PHAN first avoids employing a lot of manual operation to carry out feature engineering. At the same time, compared with current neural networks, the PHAN can effectively learn language differences and the various weights of alignments. As illustrated in Figure 2, the process can be as follows: (1) It first uses one-hot word representations as inputs without feature engineering. (2) Since parallel sentence pairs have different hierarchical components (words form sentences, two monolingual sentences form a parallel sentence pair), the model first encodes monolingual contexts to learn language differences. (3) Then, it inputs those monolingual encodings into a top network to encode a parallel sentence representation. The reason for using this network is that different words in a sentence are different. Moreover, the importance of words is highly context-dependent; that is, the same word may be differentially important in different contexts [2, 16, 17]. (4) Finally, we aggregate the outputs of the neural network into the classification layer to identify parallel sentences. The classification layer adopts the softmax function to implement a binary classification.

Our experimental results show that our method achieves significant and consistent performance compared with all baseline methods in filtering parallel sentences task. In our work, we remove feature engineering and additional computing resources. In particular, we extract parallel sentences from Wikipedia articles. Then, we use the parallel sentences to test the machine translation system and show that the extracting parallel sentences can improve machine translation.

This paper first introduces the main research content. Section 2 presents a detailed description of the model. Section 3 presents experiments and settings. Section 4 gives the detailed results of our experiment. Finally, it is the conclusion of this paper.

2. Parallel Hierarchical Attention Network

In this section, we propose a parallel hierarchical attention network (PHAN) to identify parallel sentence pairs. Figure 1 shows the structure of the PHAN. We consider a training parallel dataset $D = \{(S_i^s, S_i^t: l_i), i = 1, \ldots, N\}$ made of N pairs of sentences $\{(S_i^s, S_i^t)\}$ with labels $l_i \in \{0, 1\}$. If a pair of sentences is parallel, the label is marked as $\{1\}$, otherwise as $\{0\}$. For example, we set the label of two sentences $\{''\text{I love the motherland}'', ''\text{wo ai zuguo}''\}$ as $\{1\}$.

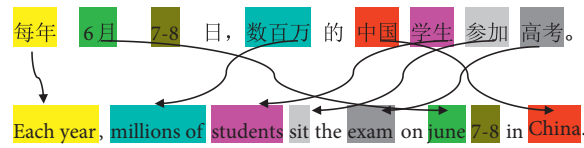


Figure 1: Not all parts of a sentence are equally relevant for representing parallel sentences.

The network takes a pair of sentences $\{(S_i^s, S_i^t)\}$ as input and output is a label of a pair of sentences $\{l_i\}$. It has two levels, monolingual sentences versus bilingual sentences. The level of monolingual sentences is made of source language encoder and target language encoder. The monolingual encoder is made of two bidirectional GRU (Gated Recurrent Unit) networks with parameters H_w and an attention model with parameters a_w, while the bilingual encoder level similarly includes a network and an attention model. The monolingual level mainly encodes monolingual sentence context and dependency. The bilingual level mainly encodes parallel sentence pair interactive context and dependency. The classification layer uses the output $p(s|t)$ to determine a label $\{l_i\}$.

2.1. Word Layers. In natural language processing, continuous word embeddings [18] are often used as the input of the neural network. However, in this task, we use the one-hot vectors instead of continuous embeddings. The reason for using one-hot vectors is that one-hot vectors can help to encode the context of a sentence. In the first step, to compare source and target sentences in the mathematical sense, we need to project them into one-hot n-dimensional space. Each word is converted into a one-hot representation. Although words are often converted into continuous word embeddings, the one-hot representation is more suitable to capture context information.

In order to get this one-hot vector, we define a lexicon $V = \{w_1, w_2, \ldots, w_m\}$, where m is the number of words of source or target sentences. A one-hot of the word w_i is an array as $[0, 0, \ldots, 1, \ldots, 0]$, and we set the number of the word in the lexicon as 1. For example, for a sentence "she is the king," the lexicon is $[''\text{she}'', ''\text{is}'', ''\text{the}'', ''\text{king}'']$. Then, the one-hot of "the" is $[0, 0, 1, 0]$. The one-hot representation of j^{th} word in the i^{th} sentence is defined as

$$\omega_{i,j}^s = \text{Embedding}\left(w_{i,j}^s\right), \tag{1}$$

where $w_{i,j}^s$ is j^{th} word in the i^{th} sentence. E^T is a pretrained embedding matrix, where Embedding( ) is a linear transformational function to embed a word to a one-hot vector. The source language has the same definition.

2.2. Encoder Layers. In the above section, we convert words into one-hot word vectors that can be calculated in the neural network. Next, we use a stream-dependent word encoder to encode each word representation to learn the near context information in a sentence.

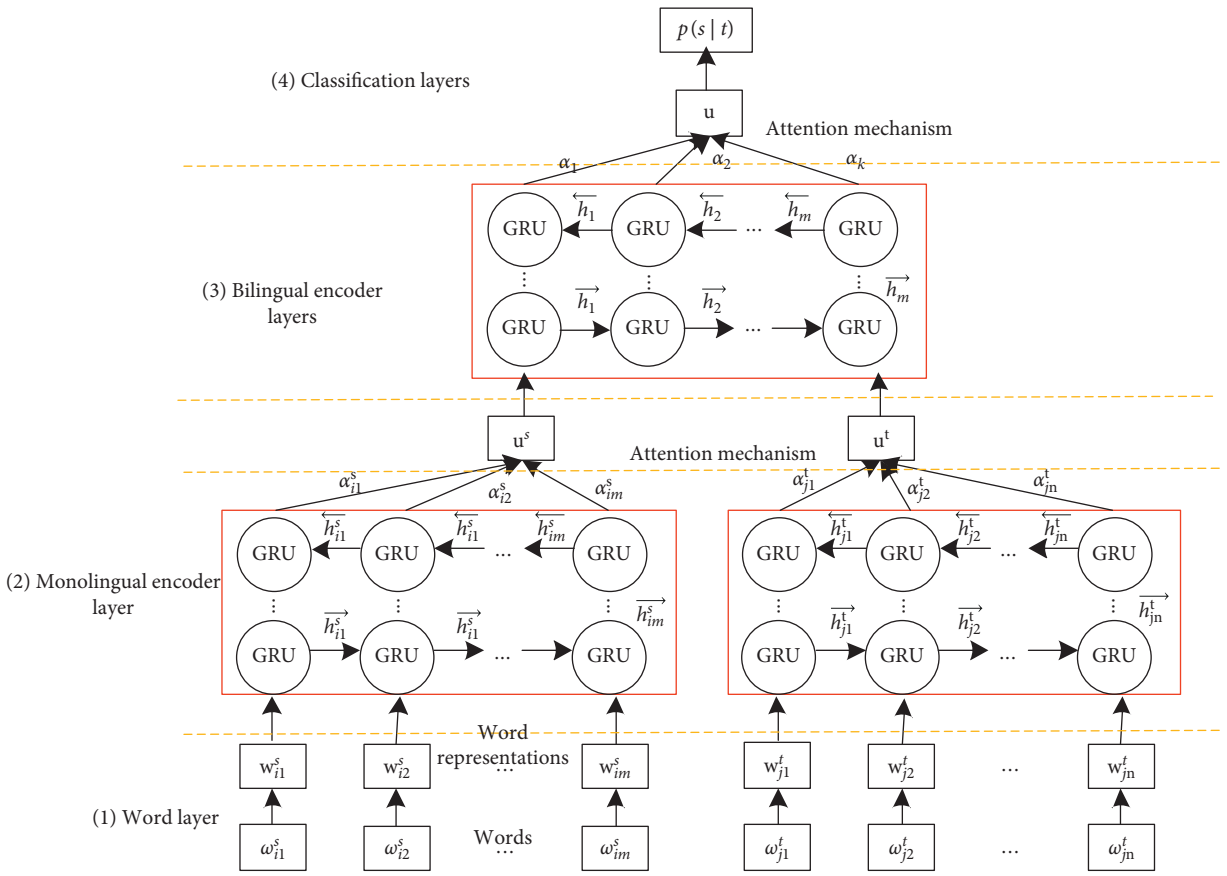


Figure 2: Hierarchical attention neural networks for modeling and selecting parallel sentences.

The traditional recurrent neural network (RNN) is affected by short-term memory. If a sequence is too long, it will be difficult to transfer information into a long step. Therefore, it will miss some important information when we process a long text. For example, when we watch a movie, we may only remember the words such as "amazing" and "excellent" and do not care about the words such as "this," "is," and "a" in the next day. The GRU can effectively achieve the above process. It can only keep some relevant information and forget useless data when we obtain parallel sentences. At the monolingual level, in order to learn the information from both directions of words, this paper uses bidirectional GRU to learn the context in a sentence. The GRU used a gating mechanism to track the state of sequences without using separate memory cells. There are two types of gates: the reset gate r_t and the update gate z_t. They together control how information is updated to the state. At the time t, the GRU computes the new state as follows:

$$h_t = (1 - z_t)\Theta h_{t-1} + z_t \Theta \widehat{h}_t, \tag{2}$$

which is the linear interpolation between the previous state h_{t-1} and the state $\widehat{h}_t$ computed with new sequence information. We use the two states to learn the context information in monolingual sentences. The gate z_t decides how much past context information is kept and how much new context information is added. This operation can effectively learn longer context information. z_t is updated as follows:

$$z_t = \sigma(w_t x_t + u_t h_{t-1} + b_z), \tag{3}$$

where x_t is the input state sequence vector with time t. The other state $\widehat{h}_t$ is computed in a similar way. $\widehat{h}_t$ is a corresponding weight that maintains a constant state.

$$\widehat{h}_t = \tanh(w_h x_t + r_t \Theta(u_t h_{t-1}) + b_h). \tag{4}$$

In fact, r_t is the reset gate which controls how much the past state information contributes to the sentences. If r_t is zero, then it forgets the previous state. We use the following equation to update the reset gate:

$$r_t = \sigma(w_r x_t + u_r h_{t-1} + b_r). \tag{5}$$

In the process, we use $w^s_{i,j}$ to represent a word in a source sentence, $t \epsilon [0, T]$. In order to encode the context information of a sentence, we use the following formula to calculate the hidden representation state for the t^{th} time in the source language:

$$\begin{aligned} \overrightarrow{h^s_{i,j}} &= \overrightarrow{\text{GRU}}\left(w^s_{i,j} : \theta^s_{r,t}\right), \\ \overleftarrow{h^s_{i,j}} &= \overleftarrow{\text{GRU}}\left(w^s_{i,j} : \theta^s_{r,t}\right), \\ h^s_{i,j} &= \left[\overrightarrow{h^s_{i,j}}, t\overleftarrow{h^s_{i,j}}\right]^T, \end{aligned} \tag{6}$$

where $\overrightarrow{h^s_{i,j}}$ and $\overleftarrow{h^s_{i,j}}$ are forward GRU functions and backward GRU functions and $\theta^s_{r,t}$ is the model parameter for word GRUs.

We obtain the context information for a given word $w_{i,j}^s$ by concatenating the forward hidden state $\overrightarrow{h_{i,j}^s}$ and $\overleftarrow{h_{i,j}^s}$, $h_{i,j}^s = [\overrightarrow{h_{i,j}^s}, \overleftarrow{h_{i,j}^s}]^T$, which summarizes information of the whole sentence. Target sentences are encoded like source sentences with an additional neural network layer, which helps the encoder to recognize the most relevant features by emphasizing critical points of the target sentence given by each source sentence.

From the example of Figure 2, we can observe that not all words contribute equally to the representation of the sentence meaning, especially when distinguishing whether two sentences are parallel. Therefore, we introduce an attention mechanism to learn this information that different words have various weights in distinguishing parallel sentences.

$$u_{i,j}^s = \tanh\left(h_{i,j}^s : \theta_w^s\right) = \tanh\left(w_w h_{i,t}^s + b_w\right)$$

$$\alpha_{i,t}^s = \frac{\exp\left(u_{i,t}^{sT} u_w\right)}{\sum_{i=1}^t \exp\left(u_{i,t}^{sT} u_w\right)}, \tag{7}$$

$$u^s = \sum_1^t \alpha_{i,t} h_{i,t}.$$

In the attention process, we first use a one-full-layer perception to learn $u_{i,t}^s$ as a hidden representation of $h_{i,t}^s$. Then, in order to learn the importance of a word in a sentence, we calculate the similarity of $h_{i,t}^s$ with a level context vector u_w. Next, we use a softmax function to get a normalized importance weight. Note that u_w is a model parameter in the attention mechanism. The context vector u_w can be seen as a high-level representation that selects which word is more important for a sentence. After that, we get a state u^s by a weighted sum of the word annotations based on the weights. We can get a target vector u^t by the same method.

At the bilingual level, after combining the intermediate vectors u^s and u^t, the function networks encode sequence vectors. We concatenate the forward GRU and the backward GRU to obtain the hidden states for each input vector.

2.3. Classification Parallel Sentence. In this section, we should detect whether a sentence pair is parallel or not from the top neural network. In order to achieve this goal, we employ a softmax layer to classify parallel sentences. The basic process is that it maps the multiple outputs of the encode layer into an interval (0, 1). In this paper, we treat the classifying parallel sentence as a binary classification problem. We input the source and target sentences into the encode layer. The encoder layer outputs a state vector u into the classification layer. For the classification layer, we use the following formula that maps the input into the interval (0, 1). It is obvious that the output of the classification layer is a probability.

$$l_i' = P\left(t_i | s_i\right) = \frac{1}{1 + e^{-\left(W_c u + b_c\right)}} \epsilon (0, 1), \tag{8}$$

where W_c is a value matrix and b_c is the bias term for the classification layer. For the classification problem, we usually use the cross-entropy as a loss.

$$l(\theta) = -\frac{1}{N} \sum_{i=1}^N \phi\left(l_i, l_i'\right). \tag{9}$$

We use ϕ to stand for the binary cross-entropy. Then, we use the gold label l_i and predicted label l_i' for a pair of a sentence i to optimize the loss. The final objective can be minimized with stochastic gradient descent (SGD) or variants such as Adam to maximize classification.

3. Experiments and Setup

In this section, we assess the effectiveness of our model. We compare our method with multiple settings. As we want to improve the performance of our model, we artificially construct negative samples.

3.1. Negative Examples. Hangya and Fraser [19] showed that a training model only using parallel sentences is not enough. There are many sentence pairs where the overall meaning is similar, but they are not parallel sentences. So, we need to generate negative examples with similar words but different meanings. Therefore, we generate synthetic noisy data from good parallel sentences. We follow [20] to generate our negative examples that have similar words but different meanings.

Gregoire and Langlais [14] showed that obtaining parallel sentences from nonparallel corpora in practice is an unbalanced classification task in which nonparallel sentences represent the majority class. Although an unbalanced training set is not desired since a classifier trained on such data typically tends to predict the majority class and has a poor precision, the overall impact on the performance of our model is not clear. So, we train a total of 10 models with $k \epsilon \{0, 1, \ldots, 9\}$, such that with $k = 0$ and $k = 9$, a model is respectively trained on the dataset with a positive to negative sentence pairs ratio of 100% and 10%.

3.2. Data. To implement experiments, we use the BUCC'17 English-French, English-Chinese, and English-German parallel datasets (https://comparable.limsi.fr/bucc2017/cgi-bin/download-data.cgi) to train our model. For test sets, we use the BUCC'17 English-French, English-Chinese, and English-German datasets (https://comparable.limsi.fr/bucc2017/cgi-bin/download-test-data.cgi). Each testing dataset contains two monolingual corpora. The monolingual corpora contain about 100 k–550 k sentences and 2,000–14,000 sentences are parallel. For the convenience of researchers, BUCC 2017 provided us with an evaluation script and a gold standard data to calculate the precision, recall, and *F*-score. For Chinese, we use OpenCC (https://github.com/BYVoid/OpenCC) to normalize characters to be simplified and then perform Chinese word segmentation and POS tagging with THULAC (http://thulac.thunlp.org). The preprocessing of English, French, and German involves tokenization, POS tagging, lemmatization, and lower casing which we carry out with the NLTK (http://www.nltk.org)

toolkit. The statistics of the preprocessed corpora are given in Table 1.

3.3. Training Settings. We use 256-dimensional GRUs for all RNNs in our model. To prevent the neural network from overfitting, we give the drop-out as 0.5 for the last layer in each module. In order to enhance our model, we add some new negative parallel sentences into training data by sampling {0, 1, . . ., 9} negative sentence pairs for each parallel sentence pair. For the system, we use TensorFlow to realize our models. All those parameters introduced earlier are based on manual analysis of the data and nonexhaustive tuning on the development set.

3.4. Baselines. We compare our model to four baselines (the parameters of the baselines follow their authors):

(1) Maximum entropy classifier (ME) [3]

(2) Multilingual sentence embeddings (MSE) [12]

(3) Dual conditional cross-entropy (DCCE) [21]

(4) An LSTM recurrent neural network (LSTM) [14]

The first baseline (ME) is the traditional statistics-based approach that is conventionally considered as alignment features between two sentences. The alignment features mainly conclude the number of connected words, the top three largest fertilities, and the length of the longest connected substring. We use those features to construct a maximum entropy classifier according to Munteanu et al. This method mainly relied on feature engineering. Feature engineering usually suffers from the language diversity issue.

The second baseline (MSE) is an important contribution of this type to approach that mentioned in [22]. First, they used a continuous vector representation of each source-target sentence pair which is learned using a bilingual distributed representation model to reduce the size and noise of the candidate sentence pairs. Then, they filtered source-target sentence pairs by feature engineering and built a support vector machine (SVM) binary classifier to identify parallel sentences. This method also relied on feature engineering.

The third baseline (DCCE): this work proposed dual conditional cross-entropy to extract parallel sentences. This work used the computed cross-entropy scores based on training two inverse translation models on parallel sentences. This method requires additional computational resources to train the translation model.

The final baseline (LSTM) is based on bidirectional recurrent neural networks that can learn sentence representations in a shared vector space by explicitly maximizing the similarity between parallel sentences. This method does not distinguish the various weights of words in detecting parallel sentences. These end-to-end network models do not add attention to encode and do not learn complex mappings and alignments to quantify parallel information.

Compared to the baselines, the PHAN first is independent of feature engineering. It makes the PHAN universal and is easy to apply the PHAN into multiple languages. Moreover, the PHAN uses a parallel hierarchical attention mechanism to capture the deep representation of monolingual and parallel bilingual sentences.

Table 1: Training and test set statistics.

Type	Language		Number
Training data	English-French		229,000
	English-Chinese		287,000
	English-German		237,000
Test data	English-French	English	38,069
		French	21,497
	English-Chinese	English	88,860
		Chinese	94,637
	English-German	English	40,354
		German	32,594

4. Results and Discussion

4.1. Model Evaluation. In this section, we first give the overall performance of different models. Table 2 shows precision, recall, and F_1 scores of three language pairs.

From Table 2, we can observe that the two methods of ME and MSE get very poor performance compared with ours. The performance is stable no matter in English-French, English-Chinese, and English-German. As the two methods of ME and MSE rely on feature engineering, alignment and bilingual words need a lot of manual annotation. However, manual annotation only covers limited language information and the high cost of manual annotation makes it difficult to obtain large-scale annotation corpus in many languages or domains. The work of [21] for the WMT18 task performed sentence pairs' extraction, was not feature-based, and gave very good results. We also verify the performance of our method by contrasting [21]. Junczys-Dowmunt [21] trained a multilingual translation model to enforce the agreement of cross-entropy scores. However, they need to train a good machine translation system to improve performance. The trained machine translation system heavily affects the performance of required parallel sentences. From Table 2, we can observe that the results of English-Chinese are not as good as English-French and English-German. As we all know, English-Chinese machine translation is not good as English-French and English-German on the same scale corpus and translation method. The reason is that English-French and English-German are similar languages, but English-Chinese is distant languages. In addition to LSTM, which does not use a parallel attention mechanism, we show a significant increase in our proposed method. Our PHAN outperforms LSTM in three language pairs. We analyze the performance of ours and LSTM; the main difference is that we treat the same words that may be differentially important in different sentences. So, we use two parallel networks and attention mechanism to learn different context information. However, LSTM does not learn this context information as it does not add an effective attention mechanism. Our model uses a parallel attention mechanism

Table 2: The precision (P), recall (R), and F_1 scores of extracting parallel sentences.

Model	En-Fr			En-De			En-Zh		
	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
ME	88.37	83.12	86.72	87.83	82.25	86.02	83.58	80.61	83.90
MSE	93.75	88.43	91.28	92.89	88.05	91.17	90.36	86.93	89.52
DCCE	94.13	89.09	92.45	92.87	89.35	91.78	90.86	87.04	89.82
LSTM	93.89	88.71	92.03	93.05	87.93	91.67	91.83	87.16	90.06
PHAN	94.27	90.03	92.63	93.16	89.73	92.06	92.07	89.37	91.23

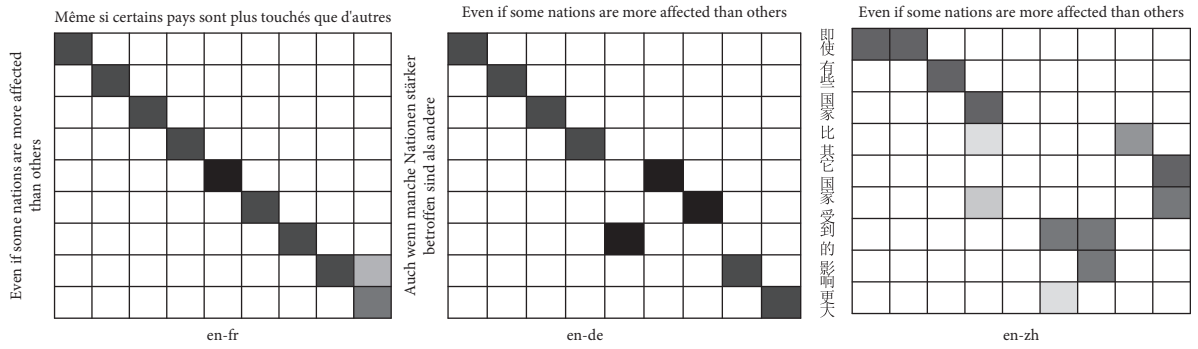


Figure 3: Our results are three alignments in three language pairs.

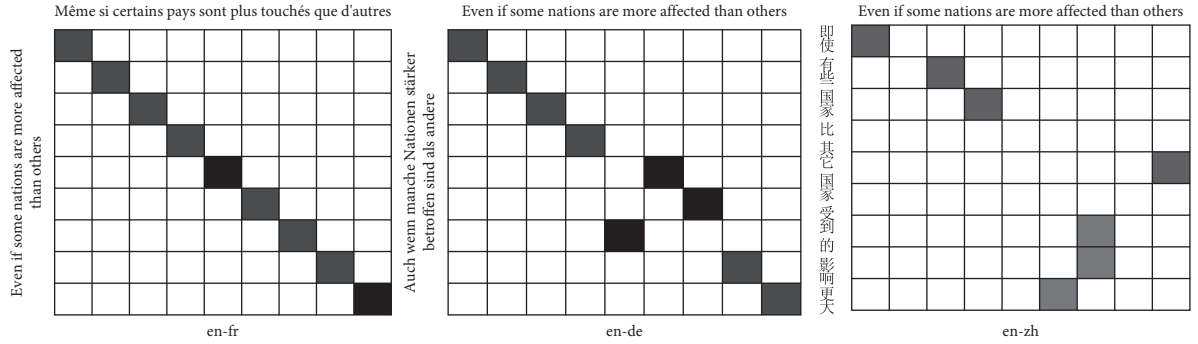


Figure 4: LSTM results are three alignments in three language pairs.

to mine more context information to improve performance. In the next section, we will carry out two experiments to further analyze our model.

4.2. Qualitative Analysis. We further analyze the performance of PHAN to observe which model can make it perform better than that without the attention mechanism. Alignment is an important factor in identifying parallel sentences. If the weights of alignment are not important, the neural network without attention mechanism may also effectively detect parallel sentences since all alignments have the same contribution. However, the alignment deeply depends on linguistics and context [23–25]. For example, the English word "bearing" means multiple Chinese words such as "chengzhou," "baochi," and "zhoucheng" in a different context.

We can visualize alignments for some sample sentences and observed translation quality as an indication of an attention model. In order to test that our model is able to mine various informative alignments in parallel sentences, we use this method to make the analysis. To test whether our model can better capture alignments than LSTM without a parallel attention mechanism, we plot the distribution of the attention weights of the words in three language bilingual sentences. The results are shown in Figures 3 and 4. The two figures show that our attention model can obtain a better-visualized alignment. From the two figures, we can find that our model can obtain various alignment weights in three language pairs. For example, our model can distinguish one-to-many alignment in English-Chinese. We can find that LSTM forces the alignment to one-to-one; if a word does not capture alignment, it will not align any words. However, we can observe the alignments of three language pairs; we find that one-to-many occurs more in English-Chinese than English-French and English-German. This may be the main factor that our model gets a bigger improvement in English-Chinese than English-French and English-German. In order to verify this hypothesis, we count the proportion of the number of words in three language sentence pairs. The

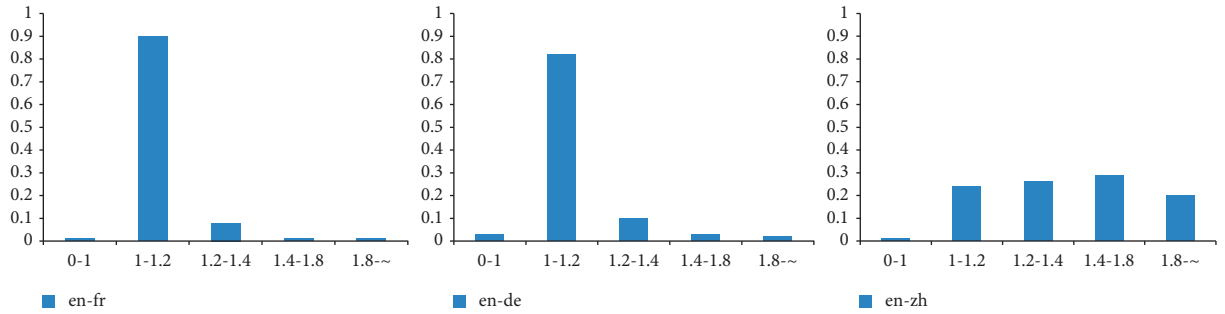


Figure 5: The ratio of the number of words in three language sentence pairs.

1.	"The Art of Eating Spaghetti (意大利面条)" caught my eye.	English
	"吃意大利面的艺术" 这一条 映入我的眼帘。	Chinese
	Chi yi da li mian de yi shu shi yi tiao ying ru wo de yan lian	
2.	"The Art of Eating Spaghetti (意大利面条)" caught my eye.	English
	"L'art de manger des spaghettis (spaghettis)" a attiré mon attention.	French

Figure 6: Different languages have different alignments for the same English sentence.

results are shown in Figure 5. We can observe that English sentences are often longer than Chinese sentences, and the other language pairs have not this situation. This makes one-to-many often occur in English-Chinese. It makes semantic confusion and affects the classification of parallel sentences. This is also an important reason why different language pairs have various accuracies in the classification of parallel sentences.

We further explore the language differences and their impact on detecting parallel sentences. We manually extract English-Chinese and English-French parallel sentences to discuss language differences. Example 1 is extracted by the PHAN, but the other baselines miss it. From Figure 6, we can observe that the English phrase "caught my eye" and the Chinese phrase "ying ru wo de yan lian" are not a suitable translation regardless of context information. According to the bilingual lexicon, "Zhua zhu wo de yan jing" is the right translation of the English phrase. However, if we use the translation "Zhua zhu wo de yan jing" to replace the phrase "ying ru wo de yan lian" in the Chinese sentence, the new sentence is wrong. Although the translation is right, it is a wrong collocation in Chinese. The ME, MSE, and DCCE need the lexicon to learn the bilingual signal, which leads to the fact that the word pairs that are not in bilingual lexicon affect detecting parallel sentences. As LSTM has no parallel attention mechanism to effectively encode monolingual information, LSTM cannot encode a monolingual context to distinguish alignments. In fact, language differences and their impact are very important in machine translation. In building machine translation systems, many works add attention to improve machine translation [26]. Example 2 is obtained by all systems. The English phrase "caught my eye" and the French phrase "attiré mon attention" are very right translations in English-French lexicon. From the above, we can conclude that our method can consider language differences by encoding the monolingual context. It can lead to a better result in detecting parallel sentences.

4.3. Performance in Machine Translation. In this paper, we hope to obtain parallel sentences and improve the performance of the machine translation system. In the training machine translation system, we use the BUCC'17 English-French, English-Chinese, and English-German parallel datasets as baselines. We use our model to extract parallel sentences from Wikipedia (https://linguatools.org/tools/corpora/wikipedia-comparable-corpora/) corpus. Then, we add the obtained parallel sentences into the three original training data as the new training set for machine translation. To evaluate the translation performance of machine translation, we use the well-known BLEU score. We use phrase-based systems that are trained with Moses for the SMT system. To train the NMT systems, we use OpenNMT (https://github.com/OpenNMT/OpenNMT-py) system.

We trained 48 machine translation systems for each SMT (http://www.statmt.org/moses/) and NMT (https://opennmt.net/) approaches. The baseline systems are trained with BUCC'17 English-French, English-Chinese, and English-German parallel sentences. For the remaining compared systems, we sort the extracted parallel sentence pairs by an extraction system in descending order according to the threshold values and append the top of {20000, 50000, . . ., 500000} and append the extracted parallel sentence pairs to the original training dataset. We change different numbers of extracted parallel sentences to train the machine translation system to test the stable performance of our model.

Table 3 shows BLEU scores in machine translation systems of SMT and NMT approaches. We can observe that adding the parallel sentences extracted by our model can lead to significant improvement compared to the baseline systems. Therefore, we know that parallel training sentences heavily affect the performance of the machine translation system. This improvement can be observed in three language machine translation systems. The table shows different gains of BLEU scores compared

Table 3: The precision (P), recall (R), and F_1 scores of extracting parallel sentences.

Data	En-Fr		En-De		En-Zh	
	SMT	NMT	SMT	NMT	SMT	NMT
Baseline	23.71	22.32	21.62	21.35	21.1	17.32
Top20K	24.84 (+1.13)	25.42 (+3.1)	23.38 (+1.76)	25.06 (+3.71)	23.21 (+2.11)	24.56 (+7.24)
Top50K	26.16 (+2.45)	26.35 (+4.8)	24.63 (+3.01)	26.42 (+5.07)	24.66 (+3.56)	25.89 (+8.57)
Top100K	28.31 (+3.6)	27.48 (+5.03)	25.72 (+4.1)	27.67 (+6.32)	25.78 (+4.68)	27.02 (+9.7)
Top200K	29.37 (+4.66)	29.51 (+6.06)	26.76 (+5.14)	28.73 (+7.38)	26.86 (+5.76)	28.13 (+10.81)
Top300K	30.39 (+5.68)	30.55 (+8.10)	27.79 (+6.17)	29.80 (+8.45)	27.91 (+6.81)	29.18 (+11.86)
Top400K	30.41 (+6.70)	30.57 (+9.12)	28.83 (+7.21)	30.82 (+9.47)	28.92 (+7.82)	30.21 (+12.89)
Top500K	31.56 (+7.85)	31.58 (+10.13)	30.14 (+8.52)	31.85 (+10.50)	29.93 (+8.83)	31.22 (+13.9)

to the baseline systems. When we get Top20K, we add extracted parallel sentence pairs to improve the BLEU score of SMT and NMT systems by 1.13 and 3.1 in English-French, and we also find this improvement in other language pairs. Then, we observe that when we get Top500K, the translation system trained on extracted parallel sentences has better BLEU than Top20K. This means that our model can effectively extract parallel sentences so that it can improve BLEU. We know that adding parallel training sentences can improve the performance of machine translation. These results confirm the quality of extracted sentence pairs and the effectiveness of our model. Hence, we can conclude that our approach could be applied to extract parallel sentences from comparable corpora and improve the performance of machine translation.

5. Conclusions

In this paper, we explore a new parallel hierarchical attention network to extract parallel sentences. Our system is able to obtain state-of-the-art performance in filtering parallel sentences while using less feature engineering and preprocessing. Additionally, our model can make full use of monolingual and bilingual sentences. Moreover, we propose a parallel attention mechanism to learn various alignment weights in parallel sentences. In the experiments, we show that our model obtains a state-of-the-art result on the BUCC2017 shared task. In particular, the effectiveness of our model in using the obtained parallel sentences to implement machine translation tasks is demonstrated.

In the future, we will explore the following directions:

(1) BPE and similar methods can effectively help us solve the out-of-vocabulary issue. We will use BPE to improve its performance

(2) Our model needs parallel sentences to be trained, which can be problematic in low-resource language pairs. In order to lessen the need for parallel sentences, identifying parallel sentences via minimum supervision is a promising avenue, especially in low-resource language pairs

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Shaolin Zhu expresses his gratitude to his supervisors Yating Yang and Xiao Li (The Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences), who guided throughout the research. This work was supported by the Chinese National Natural Science Foundation (No. 61975187) and Xinjiang Autonomous Region University Research Plan (No. XJEDU2017M027).

References

[1] A. Aghaebrahimian, "Deep neural networks at the service of multilingual parallel sentence extraction," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1372–1383, Santa Fe, NM, USA, August 2018.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the ICLR*, pp. 1–15, San Diego, CA, USA, May 2015.

[3] D. S. Munteanu and D. Marcu, "Improving machine translation performance by exploiting non-parallel corpora," *Computational Linguistics*, vol. 31, no. 4, pp. 477–504, 2005.

[4] C. Chu, T. Nakazawa, and S. Kurohashi, "Constructing a Chinese-Japanese parallel corpus from wikipedia," in *Proceedings of the LREC*, pp. 642–647, Reykjavik, Iceland, May 2014.

[5] C. Espana-Bonet, A. C. Varga, A. Barron-Cedeno, and J. van Genabith, "An empirical analysis of NMT-derived interlingual embeddings and their use in parallel sentence identification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1340–1350, 2017.

[6] J. Gonzalez-Rubio, "Webinterpret submission to the WMT2019 shared task on parallel corpus filtering," in *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pp. 271–276, Florence, Italy, August 2019.

[7] T. Kajiwara and M. Komachi, "Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings," in *Proceedings of the COLING 2016: The 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1147–1158, Osaka, Japan, December 2016.

[8] J. Lu, X. Lv, Y. Shi, and B. Chen, "Alibaba submission to the WMT18 parallel corpus filtering task," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 917–922, Brussels, Belgium, October 2018.

[9] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in*

Natural Language Processing, pp. 1412–1421, Lisbon, Portugal, September 2015.

[10] S. Mahata, D. Das, and S. Bandyopadhyay, "BUCC2017: a hybrid approach for identifying parallel sentences in comparable corpora," in *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pp. 56–59, Vancouver, Canada, August 2017.

[11] B. Marie and A. Fujita, "Efficient extraction of pseudo-parallel sentences from raw monolingual data using word embeddings," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 392–398, Vancouver, Canada, July 2017.

[12] H. Bouamor and H. Sajjad, "H2@BUCC18: parallel sentence extraction from comparable corpora using multilingual sentence embeddings," in *Proceedings of the Workshop on Building and Using Comparable Corpora*, pp. 1–5, Miyazaki, Japan, May 2018.

[13] J. Grover and P. Mitra, "Bilingual word embeddings with bucketed CNN for parallel sentence extraction," in *Proceedings of ACL 2017, Student Research Workshop*, pp. 11–16, Vancouver, Canada, July 2017.

[14] F. Gregoire and P. Langlais, "BUCC 2017 shared task: a first attempt toward a deep learning framework for identifying parallel sentences in comparable corpora," in *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pp. 46–50, Vancouver, Canada, August 2017.

[15] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, Seattle, WA, USA, October 2013.

[16] A. Klementiev, I. Titov, and B. Bhattarai, "Inducing cross-lingual distributed representations of words," in *Proceedings of the COLING*, pp. 1459–1474, Mumbai, India, December 2012.

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

[18] P. Littell, S. Larkin, D. Stewart, M. Simard, C. Goutte, and C.-K. Lo, "Measuring sentence parallelism using mahalanobis distances: the NRC unsupervised submissions to the WMT18 parallel corpus filtering shared task," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 900–907, Brussels, Belgium, October 2018.

[19] V. Hangya and A. Fraser, "An unsupervised system for parallel corpus filtering," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 882–887, Brussels, Belgium, October 2018.

[20] M. Guo, Q. Shen, Y. Yang et al., "Effective parallel corpus mining using bilingual sentence embeddings," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 165–176, Brussels, Belgium, October 2018.

[21] M. Junczys-Dowmunt, "Dual conditional cross-entropy filtering of noisy parallel corpora," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 888–895, Brussels, Belgium, October 2018.

[22] P. Koehn, H. Khayrallah, K. Heafield, and M. L. Forcada, "Findings of the WMT 2018 shared task on parallel corpus filtering," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 726–739, Brussels, Belgium, October 2018.

[23] D. Alvarez-Melis and T. S. Jaakkola, "GROMOV-wasserstein alignment of word embedding spaces," in *Proceedings of the Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1881–1890, Brussels, Belgium, October 2018.

[24] J. Chen, J. He, Y. Shen et al., "End-to-end learning of LDA by mirror-descent back propagation over a deep architecture," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1765–1773, Montreal, Canada, December 2015.

[25] K. Cho, B. Van Merri¨enboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.

[26] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: open-source toolkit for neural machine translation," in *Proceedings of the ACL 2017, System Demonstrations*, pp. 67–72, Vancouver, Canada, July 2017.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8858588, 8 pages
https://doi.org/10.1155/2020/8858588

Research Article

A Radar Signal Recognition Approach via IIF-Net Deep Learning Models

Ji Li,[1] Huiqiang Zhang,[1] Jianping Ou,[2] and Wei Wang[1]

[1]*School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2]*ATR Key Laboratory, National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Jianping Ou; oujianping@nudt.edu.cn and Wei Wang; wangwei@csust.edu.cn

Received 17 July 2020; Revised 31 July 2020; Accepted 15 August 2020; Published 28 August 2020

Academic Editor: Nian Zhang

In the increasingly complex electromagnetic environment of modern battlefields, how to quickly and accurately identify radar signals is a hotspot in the field of electronic countermeasures. In this paper, USRP N210, USRP-LW N210, and other general software radio peripherals are used to simulate the transmitting and receiving process of radar signals, and a total of 8 radar signals, namely, Barker, Frank, chaotic, P1, P2, P3, P4, and OFDM, are produced. The signal obtains time-frequency images (TFIs) through the Choi–Williams distribution function (CWD). According to the characteristics of the radar signal TFI, a global feature balance extraction module (GFBE) is designed. Then, a new IIF-Net convolutional neural network with fewer network parameters and less computation cost has been proposed. The signal-to-noise ratio (SNR) range is −10 to 6 dB in the experiments. The experiments show that when the SNR is higher than −2 dB, the signal recognition rate of IIF-Net is as high as 99.74%, and the signal recognition accuracy is still 92.36% when the SNR is −10 dB. Compared with other methods, IIF-Net has higher recognition rate and better robustness under low SNR.

1. Introduction

Radar signal recognition is a key technology in the field of radar electronic countermeasures. When receiving a radar signal, it is crucial to demodulate the signal to obtain useful information, and how to identify the signal type is the key. The accuracy of signal recognition in a complex electromagnetic environment determines the pros and cons of electronic reconnaissance systems. Due to the emergence of complex electromagnetic environments and various new system radars in modern warfare, electronic reconnaissance and electronic countermeasure systems have brought serious challenges. How to identify the type of radar signal more quickly and accurately is the key and difficult point of radar signal recognition technology.

Traditional radar signal recognition technologies include support vector machine learning (SVM) and traditional five-parameter feature matching algorithm. Li and Ying [1] achieved the purpose of identifying and classifying radar signals by extracting different entropy features. Ying and Xing [2] proposed an improved semisupervised SVM algorithm for radar signal recognition which has high accuracy. Li et al. [3] proposed a deep joint learning method, including deep representation and low-dimensional discrimination, to enhance feature stability and environmental adaptability. The approach achieved a high recognition rate for multiple radar signals under low SNR. Li [4] proposed an SKLEARN system based on automatic machine learning. Through the automatic solution algorithm of the SKLEARN system and the optimization of hyperparameters, the accuracy of radar signal recognition is improved and the stability is more reliable. Feng B et al. [5] proposed a manifold method to reduce dimensionality in high dimensions, extract features, and set an appropriate threshold as a classifier. This method had good accuracy, but did not have good generalization performance. Guo et al. [6] proposed a frequency domain analysis method and an identification method based on the Fast Correlation-based Filter Solution (FCBF) and adaboosting (AdaBoost). Under low SNR conditions, this method is more efficient than manually

extracting features for classification. Zhang et al. [7] proposed a machine learning method based on Tree-based Pipeline Optimization Too (TPOT) and Local Interpretable Model-agnostic Explanations (LIME) and used genetic algorithms to optimize the pipeline structure and related parameters. This method can not only optimize the machine learning process for different data sets but also determine the type of radar signal according to the interpretability of the radar signal when there are indistinguishable radar signals in the dataset.

However, traditional radar signal recognition technology requires artificial design of more complex features extraction algorithms and classifiers, which are more difficult to implement and have poor generalization performance. With the development of artificial intelligence (AI), the application fields of deep learning are getting wider and wider. In the field of image recognition, Convolutional Neural Networks (CNNs) is a hotspot in many researches. Its network has ability to represent learning, that is, it can extract high-order features from input information, and can respond to the translation of input features. Denaturation, which can identify similar features in different positions in space, is widely used in computer visualization, natural language processing, and other fields. Qu et al. [8] proposed a multilabel classification network based on the Deep Q-learning Network (DQN), which can be recognized under low SNR. Through the radar signal preprocessing and feature extraction of the convolutional neural network, the network can identify random overlapping radar signals under low SNR. Cai et al. [9] proposed a radar signal modulation and recognition algorithm based on an improved CNN model. In this model, a dense connection block layer and a global pooling layer were added to identify 8 radar signals. Limin et al. [10] proposed a radar signal recognition method based on an improved AlexNet model. At low SNR, they performed smooth pseudo-Wingner time-frequency analysis on a variety of signals using an improved AlexNet model, resulting in a high overall recognition rate.

In this paper, USRP N210 and USRP-LW N210 Universal Software Radio Peripheral (Universal Software Radio Peripheral) are used to simulate the radar signal transmission and reception process, and a total of 8 classes of radar signals, namely, Barker, Frank, chaotic, P1, P2, P3, P4, and OFDM, are produced with the SNR between −10~6 dB. Then, all classes of signals were distributed through the Choi–Williams distribution function (CWD) transformation to generate two-dimensional time-frequency images (TFIs). As the TFI information location distribution of different radar signals is quite different, some signal information is concentrated in the central area, and some signal information is distributed at the edge. Aiming at the abovementioned problems, this paper designed a global feature balance extraction module (GFBE) and a new IIF-Net convolutional neural network structure which has strong recognition ability for radar signals. By improving the classifier, IIF-Net has reduced the number of parameters and computation and has better identification accuracy and reliability.

2. GFBE Module and IIF-Nets

2.1. GFBE Module. The traditional radar signal recognition method is based on the conventional 5 parameters: carrier frequency (RF), angle of arrival (DOA), pulse arrival time (TOA), pulse amplitude (PA), and pulse width (PW). However, most of the signal parameters are external features, which are easy to be interfered by the external environment. The external interference will cause the distortion and loss of the signal and reduce the recognition accuracy. CNNs can adaptively learn image features for recognition, which can improve the accuracy of radar signal recognition.

With the development of computer hardware, CNN is widely used in various fields. In the article of the development of convolutional neural network and its application in image classification, Wang et al. [11] analyzed the application and development of CNN in detail. In 2012, Hinton and Alex Krizhevsky proposed AlexNet [12] and successfully applied ReLU [13], Dropout [14], and LRN [13] in CNN for the first time. Visual geometry group networks (VGG-Nets) [15] proposed a 3×3 small convolution filter, which deepened the network to 19 layers. With the increase of the network depth, the problem of network degradation appeared. After enough training times, the accuracy rate on the training set will be saturated or even decreased, and the problem of gradient and information disappearance also hinders the increase of the network depth. Residual net (ResNet) [16] solved this problem by using short skip connection and continued to increase the network depth. In image recognition, in order to extract features better, the image can be reconstructed with super resolution [17]. The improved lightweight network [18] also achieves a good classification effect.

Different convolutional layers of CNN can extract different features of the target. The shallow convolutional layer extracts the features of the target such as texture and contour, while the deep convolutional layer extracts the abstract features of the target and contains richer semantic information. However, with the deepening of the network layers, there will be problems such as information loss, gradient disappearance, and degradation. The location distribution of TFI information for different classes of radar signals is different, so this paper designed a global feature balance extraction module (GFBE), as shown in Figure 1. In Figure 1, "Conv1," "Conv3," and "Conv5" represent 1×1, 3×3, and 5×5 convolution kernels, respectively, and "Maxpool (3)" represents a 3×3 pooling layer with a stride of 1. The module contains multiple sizes of convolution kernels. The short skip connection layer of the module is composed of two "Conv1" and "Conv3". Through the short skip connection, it can prevent information loss, increase the network depth, and solve the problem of network degradation to a certain extent. The first Conv1 is used to reduce the dimension, and the second Conv1 is used to increase the dimension. The main purpose is to reduce the number of parameters and increase the nonlinear learning ability of the network. The next is the parallel convolution structure and point convolution layer, which contains convolution kernels of various sizes: "Conv5," "Conv3," "Conv1" and 3×3 MaxPool.

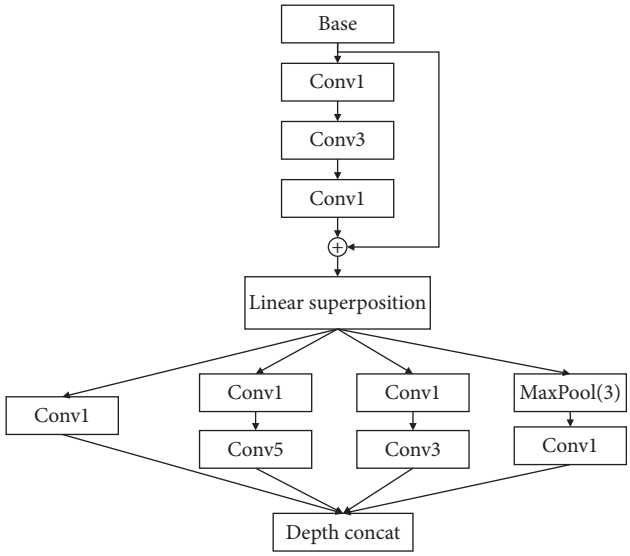


Figure 1: GFBE structure.

For TFI of different radar signals, larger convolution kernel are used for images with more dispersed information distribution, while a smaller convolution kernel is used for images with more local information distribution, which can ensure balanced extraction of image features.

2.2. IIF-Nets Structures. Based on the GFBE module, 3 IIF-Net deep CNN structures are proposed: IIF-Net56, IIF-Net107, and IIF-Net Net158. In these networks, a GFBE structure has 5 layers, where a "Conv" is a composite structure containing "convolution," "batch standardization," and "activation function". The network structure is shown in Table 1.

Radar signal recognition technology requires high real-time performance, and recognition must be made immediately when the signal is captured. The network is required to have less parameters and low calculation cost to reduce the consumption of hardware, so the global average pooling (GAP) [19] is used as the classifier of IIF-Net. This classification method does not require a fully connected layer, which can greatly reduce the number of parameters and can avoid overfitting under certain conditions.

2.3. Network Complexity. When different classifiers are used to identify 8 classes of radar signals, the network parameters and calculations are different. Suppose the size of the output feature map of the last layer is $H \times W \times D$, when using three fully connected layers, the number of parameters in the classifier is $16,818,184 + 4096 \times H \times W \times D$. When a single-layer fully connected layer is used, the parameters in the classifier are $H \times W \times D \times 8 + 8$. When using GAP, since the pooling layer has no parameters, the number of parameters can be further reduced to $D \times 8 + 8$.

The number of parameters for different networks is shown in Figure 2, and the number of calculation is shown in Figure 3.

It can be seen from Figure 2 that IIF-Net slowly increases the parameter amount with the increase of the network depth, and the network depth has little effect on the parameter amount. The VGG16 network has only 16 layers, but the amount of parameters is 5.44 times that of IIF-Net56, 3.11 times that of IIF-Net107, and 2.30 times that of IIF-Net158. IIF-Net has 6 more layers than ResNet, but the number of parameters is reduced by about 110,000. The radar system requires high real-time performance, but the small equipment, such as bombs, has insufficient memory, and its hardware is hard to support too many parameter quantities. IIF-Net is relatively small in parameter quantity, which is a kind of a better choice.

According to Figure 3, the calculation of the VGG network is very huge. The floating-point operations per second (FLOPs) of VGG16 is as high as 15.583 billion, which is 2.94 times that of the 56-layer IIF-Net. Network structure and network depth have a great impact on the amount of computation. IIF-Net is deeper than ResNet, so the amount of calculation is increased. The number of layers of IIF-Net107 is 1.80 times that of IIF-Net56, so the amount of calculation is 1.71 times that of IIF-Net56. The amount of IIF-Net158 is 2.42 times that of Net56, which is very huge. Therefore, when the difference in the signal recognition rate is not large, IIF-Net56 has the highest cost performance.

3. Experimental Results

3.1. Dataset. The dataset is generated by USRP N210, USRP-LW N210 simulating the process of real radar signal transmission and reception. The generated signal is transformed by CWD to obtain TFI. Unlike SAR images [20] in radar target recognition and high-resolution radar target images [21], TFI is a digital image with low image information loss, which is convenient for computer processing and analysis.

There are many methods of time-frequency analysis, including short-time Fourier transform (STFT), continuous wavelet transform (CWT), bilinear models including Wigner–ville distribution, pseudosmooth (WVD), CWD, adaptive parameter models (such as the ARMA model, time-frequency rearrangement model (RS), and synchronous extraction model SET). But, they have some shortcomings. For example, the time-frequency resolution of STFT and CWT is insufficient. The effect of WVD on multicomponent signal interference is poor. The RS complexity is too high; SST and SET are very advantageous for instantaneous frequency extraction and signal reconstruction, but the signal energy is too compressed, resulting in only one line at the frequency point. In this paper, high definition CWD transform is adopted, and an appropriate mask function is selected to avoid the cross-term problem, which improves the recognition performance of the radar signal.

The Choi–Williams distribution function is one of a series of Cohen's class distribution functions. The distribution uses an exponential core function to filter out cross terms. The core function of the Choi–Williams distribution does not increase with the increase of μ and τ, so it can filter out the cross terms with different frequencies and time centers.

Table 1: IIF-Net configuration.

IIF-Net56		IIF-Net107		IIF-Net158	
Conv7-64, stride: 2, padding: 3 × 3 Maxpool, stride: 2, padding: 1					
Conv1-64 Conv3-64 Conv1-256	×2	Conv1-64 Conv3-64 Conv1-256	×2	Conv1-64 Conv3-64 Conv1-256	×2
GBFE-256					
Conv1-128 Conv3-128 Conv1-512	×3	Conv1-128 Conv3-128 Conv1-512	×3	Conv1-128 Conv3-128 Conv1-512	×7
GBFE-512					
Conv1-256 Conv3-256 Conv1-1024	×5	Conv1-256 Conv3-256 Conv1-1024	×22	Conv1-256 Conv3-256 Conv1-1024	×35
GBFE-1024					
Conv1-512 Conv3-512 Conv1-2048	×3	Conv1-512 Conv3-512 Conv1-2048	×3	Conv1-512 Conv3-512 Conv1-2048	×3
GAP Classifier, Soft-max					

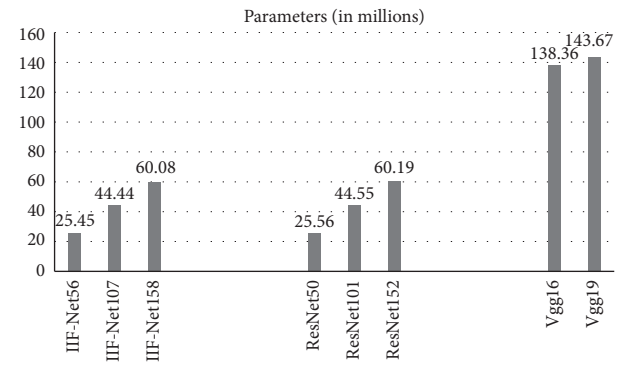


Figure 2: Parameters.

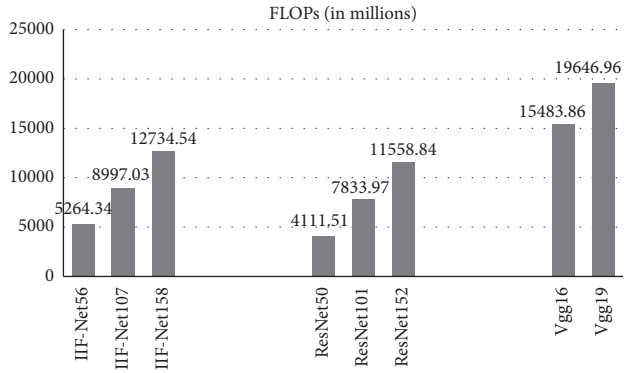


Figure 3: FLOPs.

$$C_x(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}A_x(\mu,\tau)\varphi(\mu,\tau)\exp(j2\pi(\mu t-\tau f))\mathrm{d}\mu\mathrm{d}\tau, \tag{1}$$

where $A_x(\mu,\tau)=\int_{-\infty}^{\infty}x((t+\tau)/2)x^*((t-\tau)/2)e^{-j2\pi t\mu}\mathrm{d}t$ is the fuzzy function, μ and τ are, respectively, the frequency offset and delay, and $x(t)$ is the received signal.

The core function $\varphi(\mu,\tau)=\exp[-\alpha(\mu\tau)^2]$ is a Gaussian function, where α is an adjustable parameter.

In the radar signal dataset, there are 8 types of signals. Each class of signal generates 2592 TFIs, and the SNR is −10~6 dB. Each class of signal has a total of 20,736 samples, and every 2 dB contains 288 samples. Figure 4 shows the TFI of the signal after passing through CWD.

It can be seen from the images that the distribution of different signal information is different: the distribution of chaotic code information is relatively concentrated, the distribution of OFDM signal information is relatively scattered, and the information distributions of P1–P4, Barker, and Frank are below the center, with irregular signal characteristics.

3.2. Preprocessing. In the experiments, we downsample the samples of the training set and the test set to a fixed resolution of 224 × 224 and, then, expand the data: randomly flip the image horizontally, randomly flip vertically, and randomly rotate 90°. The data set is expanded by 3 times to prevent the network from overfitting.

In order to maintain the unity of the experiments, the experiments are conducted on the same platform. The platform of signal generation is shown in Table 2.

During the experiment, the parameters were set up, the learning rate is 0.001, the momentum is 0.9, the weight decay is $5e-4$, and the batch size is 10. The experimental platform configuration is shown in Table 3.

3.3. Experimental Results. In order to make the radar signal recognition more authentic and simulate the interference of a complex external environment, noises with an SNR of −10~6 dB are added to the signal. The real radar signal transmission and reception process is simulated by USRP N210 and USRP-LW N210. The generated signals are transformed by CWD to obtain TFI for radar signal

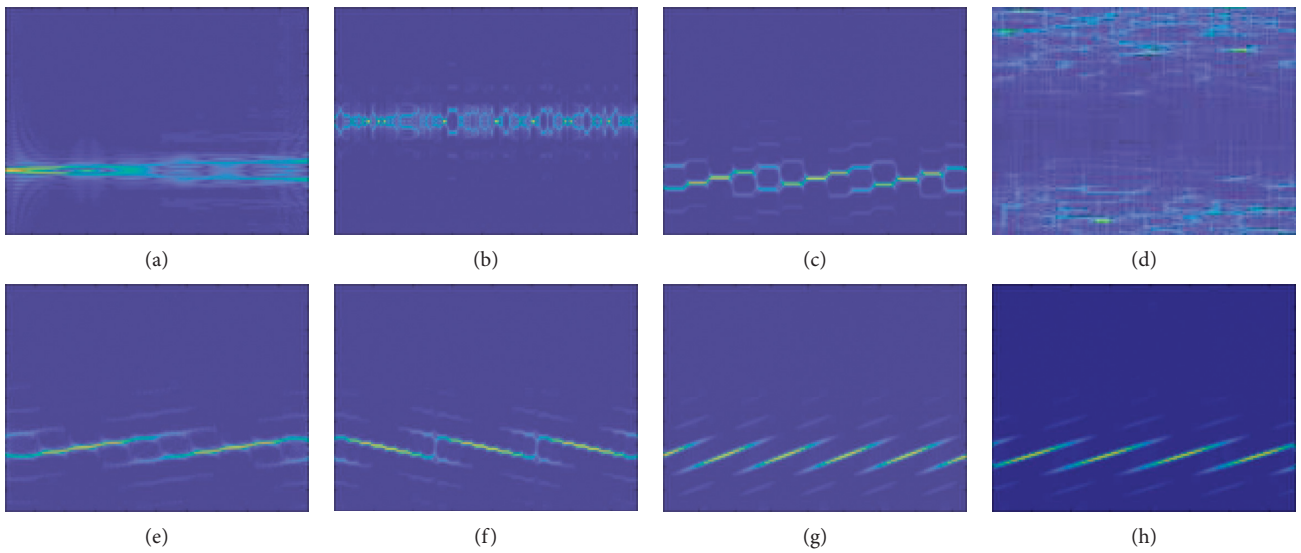


Figure 4: TFI of various radar signals. (a) Barker, (b) Frank, (c) chaotic, (d) OFDM, (e) P1, (f) P2, (g) P3, and (h) P4.

Table 2: Signal generation platform configuration.

Parameter	USRP N210/USRP-LW N210
REF IN	15 dBm
PPS IN	5 V
Power	6 V, 3 A
ADC sampling rate	100 MS/s
DAC sampling rate	400 MS/s
LO accuracy	2.5 ppm

Table 3: Experimental platform configuration.

Attributes	Configuration information
Operating system	Ubuntu 14.04.5 LTS
CPU	Intel (R) Xeon (R) CPU E5-2670 v3 @ 2.30 GHz
GPU	GeForce GTX TITAN X
CUDNN	CUDNN 6.0.21
CUDA	CUDA 8.0.61
Frame	PyTorch

identification. Under the same training set and test set, we use different depths of IIF-Net to identify radar signals under different SNRs. The experimental results are shown in Table 4.

According to Table 4, the signal recognition rate of IIF-Net56 is 99.36% and in the case of SNR is −4 dB. When the SNR is −10 dB, the noise causes a lot of interference, but the recognition rate is still higher than 92%. The results indicate that the IIF-Net networks are robust. The recognition rate of IIF-Net56 is about 1% lower than that of the other 2 networks. It shows that, with the deepening of network depth, there is no obvious difference in the extraction of signal features. The parameter amount of IIF-Net158 and IIF-Net107 is 2.36 times and 1.75 times of that of IIF-Net56, and the calculation amount is 2.42 times and 1.71 times of that of IIF-Net56. Based on the experimental results, we found that

Table 4: IIF-Net recognition accuracies at different depths (%).

SNR (dB)	IIF-Net56	IIF-Net107	IIF-Net158
−10	92.36	92.54	92.85
−8	94.55	95.56	95.64
−6	96.53	96.73	97.52
−4	99.36	99.48	99.53
−2	99.74	100	100
0	100	100	100
2	100	100	100
4	100	100	100
6	100	100	100

IIF-Net158 had the best recognition performance, but the network parameters and calculation amount increased greatly. Therefore, based on the abovementioned analysis, IIF-Net56 has the highest cost-performance ratio.

Under the same training set and test set, we also compare IIF-Net56 with other networks. Experimental results of other CNN networks are shown in Table 5.

According to Table 5, various classic CNNs have a good recognition rate for radar signals when the SNR is above 0 dB. However, when the SNR is between −10 dB and 0 dB, IIF-Net has the highest recognition performance. Compared with IIF-Net, the signal recognition rate of VGG-Net is about 6% lower than that of IIF-Net. Because of VGG-Net's shallow network, it cannot fully extract the features of the image, resulting in low signal recognition rate. Moreover, VGG-Net has too large parameters and calculation and requires too much hardware equipment and more calculation time. Therefore, VGG-Net is not suitable for the radar electronic countermeasure field which needs high real-time performance.

The signal recognition rate of ResNet is close to IIF-Net, which is about 2% lower. Because ResNet uses short skip connection, it can deepen the network and solve the problem of "network degradation" to a certain extent. It can also prevent information loss during network transmission.

Table 5: Recognition accuracy rates of other CNNs (%).

SNR (dB)	ResNet50	ResNet101	ResNet152	VGG16	VGG19	IIF-Net56
−10	90.49	90.85	91.24	86.85	88.59	92.36
−8	92.68	93.79	94.46	89.26	90.27	94.55
−6	94.65	95.15	96.31	92.57	94.16	96.53
−4	97.47	97.83	98.52	95.61	96.54	99.36
−2	98.87	99.26	99.49	98.42	99.62	99.74
0	99.51	100	100	99.53	99.75	100
2	100	100	100	100	100	100
4	100	100	100	100	100	100
6	100	100	100	100	100	100

However, the distribution of TFI feature information of a radar signal is irregular, and ResNet mostly uses small convolution kernel of 3×3, which has good recognition effect for images with concentrated information distribution and has low recognition effect for TFI features of radar signal. The GFBE module proposed in this paper solves this problem to a certain extent. For images with different information distribution, it can extract image features in a global and balanced way, improve signal recognition rate, and enhance generalization.

We further compare IIF-Net56 with other radar signal recognition methods, and the results are shown in Table 6.

According to Table 6, the signal recognition rate of the DQN network at −6 dB is higher than that of IIF-Net56, which is 1.05% higher, but at −10 dB, the recognition rate is much lower than that of IIF-Net56, which is reduced by 4.81%. This indicates that high-intensity noise has little influence on IIF-Net, and IIF-Net can still fully extract image information, obtain high signal recognition rate, and have good robustness. It can also be seen from the table that when the SNR is above −6 dB, the signal recognition rate obtained by I-CNN has little difference from that of IIF-Net, and both of them have good recognition effect. When the SNR is −10 dB and −8 dB, the signal recognition rate of IIF-Net is much higher than that of I-CNN, which shows that IIF-Net has strong anti-interference ability and can extract image features in a balanced and sufficient way. Fusion Image uses transfer learning and a cascaded automatic encoder based on self-learning to extract the effective information of the fused image, thereby ensuring the recognition performance. Meanwhile, Fusion Image adopts multifeature Fusion algorithm to fuse features, which reduces redundant information of features, but its recognition rate is 1.03% lower than that of IIF-Net56 at −6 dB. FCBF-AdaBoost and Entropy are traditional image classification methods, which are mostly designed for certain classes of image features. Their recognition rates are relatively poor in multitask and low SNR environments.

Under the same training set and test set, the recognition rates of IIF-Nets proposed in this paper under different SNRs are shown in Table 7.

It can be seen from Table 7 that, under the environment of low SNR (−10 dB), 3 IIF-Net networks have little difference in the recognition effect of different radar signals. The deepening of the network depth has a significant effect on the recognition rate of various radar signals. The influence range of network depth on the recognition rate of various radar signals is between 1% and 2%. This indicates that when the network depth reaches a certain degree, the signal feature information can be fully extracted. Further deepening of the network has little impact on the recognition effect of signals, but the recognition effects of different classes of radar signals under the same network are greatly different. Among them, Barker has the best recognition effect, over 97%. chaotic, Frank, OFDM, P2, and P3 receive the next highest recognition rates, with accuracy rates of over 94 percent, while P1 and P4 have relatively poor recognition effects, at about 80 percent. According to the TFI of the radar signal, P1 and P4 are very similar. Under the environment of −10 dB, the energy of noise is much greater than that of the signal, and the information features of the signal are drowned by the noise, which makes P1 and P4 more similar and greatly increases the difficulty of identification. However, IIF-Net56 has a comprehensive recognition rate of 92.36% under −10 dB, and its recognition performance is higher than that of other methods.

The IIF-Net proposed in this paper can extract information globally for images with irregular information distribution, which has a good recognition effect. Other traditional methods are mostly designed for specific classes of images. When the image changes greatly, their recognition effects are poor. The artificially designed feature extraction algorithm is also relatively complex, and its generalization performance is low. Compared with other CNNs, IIF-Net still has a recognition rate of 92.36% under −10 dB, which is higher than that of those other CNNs.

3.4. Experiments Analysis. This paper proposes 3 IIF-Net structures, namely, IIF-Net56, IIF-Net107, and IIF-Net158. According to the experimental results, their signal recognition rates are above 99.74% when the SNR is higher than −2 dB. At −10 dB, the recognition rates are as high as 92.36%. When deepening the networks, the differences between the recognition rates of the three networks are within 1%, but the parameters and calculations have increased significantly. Therefore, IIF-Net56 has the best overall performance.

The information characteristic distribution of the radar TFI signal is irregular. Therefore, the distribution characteristics and irregularity of image information should be taken into account when extracting image features. A parallel convolutional layer can be used to extract different types

Table 6: Recognition accuracy rate of other methods (%).

Method	−10	−8	−6	−4	−2	0	2	4	6
DQN [8]	87.55	—	97.58	—	—	—	100	100	100
Entropy [1]	66.50	—	—	—	—	100	—	—	—
FCBF-AdaBoost [6]	—	—	—	—	—	94.46	96.86	98.75	98.52
Fusion Image [22]	—	—	95.50	—	—	—	—	—	—
I–CNN [23]	55	80	96.10	—	—	100	100	100	100
IIF-Net56	92.36	94.55	96.53	99.36	99.74	100	100	100	100

Table 7: Recognition accuracy of the same signal in different networks (–10 dB) (%).

Signal	IIF-Net56	IIF-Net107	IIF-Net158
Barker	100.00	97.22	100.00
Chaotic	96.56	100.00	97.35
Frank	95.83	98.61	96.26
OFDM	96.54	100.00	98.85
P1	81.67	79.17	80.37
P2	95.44	94.44	94.68
P3	94.72	97.22	95.84
P4	80.52	80.56	81.41

of image information. The network depth should be kept moderate. It is difficult to fully extract image features when the network is too shallow, but the recognition rate cannot be significantly improved when the network is too deep. If the network is too deep, the degradation problem may occur, and the amount of parameters and calculation will increase greatly. To a certain extent, the problem of network degradation can be solved by using a short skip connection mode, while the integrity of image information can be maintained. The classifier can choose GAP to reduce the number of network parameters and calculations. The GFBE module includes Conv1, Conv3, Conv5, and MaxPoo(3) to deepen the network through short skip connection to prevent the loss of image information and uses Conv3, Conv5, and the MaxPool(3) parallel convolutional layer to extract global information. At the same time, it controls the dimensions of the network through Conv1 and improves the nonlinear learning ability of the network.

4. Conclusions

In this paper, USRP N210 and USRP-LW N210 are used to simulate the transmitting and receiving process of radar signals to generate near-real radar signals. Then, CWD is used to get the radar TFI. According to the irregular information distribution characteristics of radar signal TFI, we designed a GFBE module. Based on this module, three network structures, IIF-Net56, IIF-Net107, and IIF-Net158, are proposed. Through analysis, we conclude that IIF-Net56 has the best comprehensive performance. The network has a recognition rate of 92.36% at a low SNR of −10 dB. GAP is added into the network, and the number of parameters and calculation amount are relatively less, which reduces the requirement for hardware equipment. IIF-Net56 uses a GAP layer to reduce the amount of parameters and calculation and reduces the requirements of hardware equipment. Therefore, the network proposed in this paper has a good application prospect in the field of high real-time radar electronic countermeasures. In the field of radar electronic countermeasures, transmitting jamming signals for electronic countermeasures is a common method. In the future, we will do further research on radar jamming signal recognition.

Data Availability

The dataset in the paper can be obtained by contacting Huiqiang Zhang (hqzhang9013@163.com).

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China under Grant 61471370, Scientific Research Fund of Hunan Provincial Education Department under Grant 17C0043, Hunan Provincial Natural Science Fund under Grant 2019JJ80105, Changsha Science and Technology Project "Intelligent processing method and system of remote sensing information for water environment monitoring in Changsha", and Hunan Graduate Scientific Research Innovation Project under Grant CX20200882.

References

[1] J. Li and Y. Ying, "Radar signal recognition algorithm based on entropy theory," in *Proceedings of the 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, pp. 718–723, Shanghai, China, November 2014.

[2] F. Ying and W. Xing, "Radar signal recognition based on modified semi-supervised SVM algorithm," in *Proceedings of the IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2336–2340, Chongqing, China, March 2017.

[3] D. Li, R. Yang, X. Li, and S. Zhu, "Radar signal modulation recognition based on deep joint learning," *IEEE Access*, vol. 8, pp. 48515–48528, 2020.

[4] P. Li, "Research on radar signal recognition based on automatic machine learning," *Neural Computing and Applications*, vol. 32, no. 7, pp. 1959–1969, 2020.

[5] B. Feng and Y. Lin, "Radar signal recognition based on manifold learning method," *International Journal of Control and Automation*, vol. 7, no. 12, pp. 399–406, 2014.

[6] J. Guo, P. Ge, W. Jin, and W. Zhang, "Radar signal recognition based on FCBF and Adaboost algorithm," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 4185–4190, Wuhan, China, July 2018.

[7] W. Zhang, P. Ge, W. Jin, and J. Guo, "Radar signal recognition based on TPOT and LIME," in *Proceedings of the 2018 37th Chinese Control Conference (CCC)*, pp. 4158–4163, Wuhan, China, July 2018.

[8] Z. Qu, C. Hou, C. Hou, and W. Wang, "Radar signal intra-pulse modulation recognition based on convolutional neural network and deep Q-learning network," *IEEE Access*, vol. 8, pp. 49125–49136, 2020.

[9] J. Cai, C. Li, and H. Zhang, "Modulation recognition of radar signal based on an improved CNN model," in *Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 293–297, Dalian, China, October 2019.

[10] G. Limin, X. Chen, and C. Tao, "Radar signal modulation type recognition based on AlexNet model," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 49, no. 3, pp. 1000–1008, 2019, in chinese.

[11] W. Wang, Y. Yang, X. Wang, W. Wang, and L. I. Ji, "The development of convolution neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, p. 040901, 2019.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1097–1105, Doha, Qatar, November 2012.

[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the International Conference on Machine Learning*, pp. 807–814, Haifa, Israel, June 2010.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky et al., "Improving neural networks by preventing co-adaptation of feature detectors," 2012, http://arxiv.org/abs/1207.0580.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.

[16] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[17] W. Wei, J. Yongbin, L. Yanhong, L. Ji, W. Xin, and Z. Tong, "An advanced deep residual dense network (DRDN) approach for image super-resolution," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592–1601, 2019.

[18] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-MobileNet models," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.

[19] M. Lin, Q. Chen, S. Yan et al., "Network in network," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.

[20] W. Wang, C. Zhang, J. Tian et al., "A SAR image targets recognition approach via novel SSF-net modelse," *Computational Intelligence and Neuroscienc*, vol. 2020, Article ID 8859172, 9 pages, 2020.

[21] W. Wang, C. Zhang, J. Tian et al., "High resolution radar targets recognition via inception-based VGG (IVGG) networks," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8893419, 11 pages, 2020.

[22] L. Gao, X. Zhang, J. Gao, and S. You, "Fusion image based radar signal feature extraction and modulation recognition," *IEEE Access*, vol. 7, pp. 13135–13148, 2019.

[23] Z. Qu, X. Mao, and Z. Deng, "Radar signal intra-pulse modulation recognition based on convolutional neural network," *IEEE Access*, vol. 6, pp. 43874–43884, 2018.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8887453, 8 pages
https://doi.org/10.1155/2020/8887453

Research Article

Image Target Recognition via Mixed Feature-Based Joint Sparse Representation

Xin Wang,[1] Can Tang,[1] Ji Li,[1] Peng Zhang,[2] and Wei Wang[1]

[1]*School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2]*School of Electronics and Communications Engineering, Sun Yat-sen University, Shenzhen 518107, China*

Correspondence should be addressed to Peng Zhang; zhangpeng5@mail.sysu.edu.cn and Wei Wang; wangwei@csust.edu.cn

Received 5 July 2020; Revised 14 July 2020; Accepted 23 July 2020; Published 10 August 2020

Academic Editor: Nian Zhang

An image target recognition approach based on mixed features and adaptive weighted joint sparse representation is proposed in this paper. This method is robust to the illumination variation, deformation, and rotation of the target image. It is a data-lightweight classification framework, which can recognize targets well with few training samples. First, Gabor wavelet transform and convolutional neural network (CNN) are used to extract the Gabor wavelet features and deep features of training samples and test samples, respectively. Then, the contribution weights of the Gabor wavelet feature vector and the deep feature vector are calculated. After adaptive weighted reconstruction, we can form the mixed features and obtain the training sample feature set and test sample feature set. Aiming at the high-dimensional problem of mixed features, we use principal component analysis (PCA) to reduce the dimensions. Lastly, the public features and private features of images are extracted from the training sample feature set so as to construct the joint feature dictionary. Based on joint feature dictionary, the sparse representation based classifier (SRC) is used to recognize the targets. The experiments on different datasets show that this approach is superior to some other advanced methods.

1. Introduction

In recent years, sparse representation classification (SRC) approach has successfully been used in the field of image recognition. Compared with other methods, SRC is robust to illumination, occlusion, and noise. In the feature extraction stage, the traditional image recognition methods based on sparse representation usually use the original samples directly or the low-dimensional samples after dimensionality reduction as the atoms to construct the dictionary. However, the dictionary constructed in this way cannot effectively represent the test samples, and it is difficult to make full use of the information hidden between the training samples. So, many scholars began to study the use of various features in the construction of dictionaries.

Gabor transform is a windowed Fourier transform, first proposed by Lee [1]. Later, Gabor wavelet transform was put forward by combining Gabor transform with wavelet transform. Different from the traditional Fourier transform, Gabor wavelet transform can easily adjust the frequency and direction of the filter, so the signal features obtained by Gabor wavelet transform have good discrimination in the time-space domain and the frequency domain. Using Gabor wavelet transform to extract the features of the original samples for sparse representation classification can avoid the problems caused by the direct construction of dictionaries from the original samples to some extent. Lu and Zhang proposed a face recognition method based on discriminant dictionary learning, which obtained the Gabor amplitude images of the faces through Gabor filter. Then, they used the Gabor amplitude images to construct a new dictionary for sparse representation classification, which improved the recognition rate of the face images in the uncontrolled environment [2].

As a popular image classification and recognition framework, convolutional neural network (CNN) has attracted a great deal of scholarly attention. However, CNN needs a large number of samples for training. In reality,

many samples are not easily obtained, and the cost of CNN parameters adjustment is also large. CNN can extract a variety of features, such as texture, shape, color, and topology at the same time, so it is also very suitable to be used as a tool to extract image features [3, 4]. Zhang et al. proposed a CNN-GRNN model for image classification and recognition [5]. The model used CNN to extract image features and then used general regression neural network (GRNN) for classification and recognition. The deep features extracted by CNN enabled the method to have a good recognition effect. In order to better extract the features, the image superresolution can be applied for the image reconstruction first [6].

When Gabor wavelet transform is used to extract features for target recognition, the impact of light condition transformation on recognition can be reduced. At the same time, it has better robustness for image deformation and rotation to some extent. Therefore, this paper proposes an image target recognition method based on mixed features and joint sparse representation (M-JSR). The Gabor wavelet feature extracted by Gabor wavelet transform and the deep feature extracted by CNN were combined to form the hybrid feature and carry out adaptive weighting and PCA dimensionally reduction for mixed features and finally combined with the joint sparse model for classification recognition. The problem of poor representation ability of the original dictionary is avoided by building the dictionary with mixed features instead of the original sample. Compared with using CNN for classification recognition, M-JSR does not require a large number of training samples nor does it need a lot of time to adjust parameters. Moreover, the joint sparsity model divides the dictionary into the public features part and the private features part, so that the dictionary has better discrimination ability, and thus improves the recognition accuracy.

2. Feature Extraction

2.1. Gabor Wavelet Feature Extraction. Gabor wavelet transform has unique advantages in the representation, and analysis of image signals for images can be processed in different scales and directions. In simple terms, Gabor wavelet transform is used to convolve a set of Gabor filter functions with a given image signal.

In general, the two-dimensional Gabor function can be expressed as [1]

$$\psi_{u,v}(m,n) = \frac{k^2}{\sigma^2}\exp\left(-\frac{k^2(m+n)^2}{2\sigma^2}\right) \cdot \left[\exp\left(ik\cdot\left(\frac{m}{n}\right)\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right], \quad (1)$$

where $k = k_v(\cos\theta, \sin\theta)^T$, $\theta = \pi u/8$ represents the direction of the filter, $k_v = k_{\max}/f^v$, $k_{\max}$ represents the maximum frequency, f is the interval factor of the kernel function in the frequency domain, and u and v represent the direction and scale of Gabor wavelet, respectively. Researches show that using 5 scales ($v = 0, 1, 2, 3,$ and 4) and 8 directions ($u = 0, 1, 2, 3, 4, 5, 6,$ and 7) can get the best effect [7]. m and n represent the spatial coordinates of the image, σ is the radius of the Gaussian function (which is the size of the two-dimensional Gabor wavelet) and i is a complex number operator.

Assume the input image is $I = (m, n)$, then

$$F_{u,v}(m,n) = I(m,n) \otimes \psi_{u,v}(m,n), \quad (2)$$

where $F_{u,v}(m,n)$ represents the Gabor wavelet features of the image $I = (m, n)$.

2.2. Deep Feature Extraction. Convolutional neural network (CNN) [8] is a feedforward neural network, which is essentially a multilayer perceptron. A complete convolutional neural network consists of the input layer, the convolutional layer, the subsampling layer (pooling layer), and the fully connected layer. The convolution layer is used to extract the features of the input data, and it generally contains multiple convolution kernels. The pooling layer mainly compresses the features which are extracted by the convolution layer to decrease the complexity of network computing and improve the robustness. The full connection layer combines the previously extracted features nonlinearly and sends the output value to the classifier, such as softmax classifier. Therefore, in addition to image classification, CNN can also be used as a tool to extract image features.

For extracting sparse features, we draw on the viewpoint of the literature [9–11] about network design. Visual geometry group networks (VGGNets) proposed by Simonyan and Zisserman have significantly improved image recognition performance by deepening the network to 19 layers. VGG19 network is used to extract deep features, and its structure is shown in Figure 1. In VGG19, the convolution filters are set to 3×3, and the max pooling is 2×2 with stride 2. VGG19 has better performance than other convolutional network models in extracting target features. As shown in Figure 1, the number of convolution kernels at the next layer is doubled when the size of the feature map is reduced by half through the max pooling layer. VGG19 ends with three fully connected layers and softmax function.

The convolution kernel of CNN convolutional layer can automatically extract complex global and local features from the image. The convolution kernels of shallow layers in the CNN network extract mostly texture and detail features. Relatively speaking, the deeper the layers are, the more representative the extracted features will be, while the resolution of the feature maps will become lower. As shown in Figure 2, the middle part is the original figure, the left side is the feature extracted by the convolution layer of the first part of VGG19 network, and the right side is the feature extracted by the convolution layer of the second part of VGG19 network.

3. Joint Sparsity Model

3.1. Joint Sparsity Model. The joint sparsity model (JSM) was originally used for the coding of multiple related signals in distributed compressed sensing scenes [12]. In JSM,

Figure 1: Structure of VGG19.

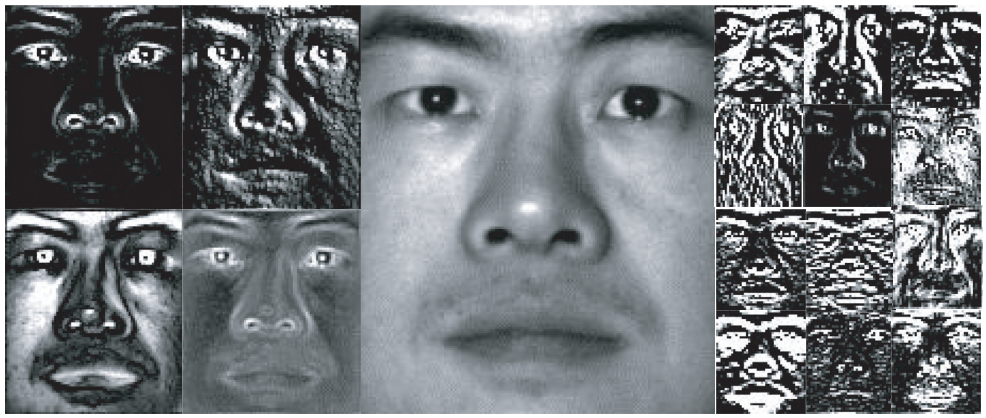


Figure 2: Samples of deep features.

according to the intrasignal and the intersignal correlation, a group of related signals can be regarded as a signal set. Then, each signal in the signal set can be jointly represented by the public feature of this type of signal and its own private feature, such as formula (3). Both public and private features can be sparsely represented on the same sparse basis.

$$y_j = z_c + z_j, \quad j \in \{1, 2, 3, \ldots, J\}, \tag{3}$$

where y_j is the jth signal in a certain type of signal, z_c represents the public feature of this type of signal, and z_j represents the private feature of the jth signal.

If all the samples can be classified into K categories, and each containing J samples, the jth sample of class i can be represented as $y_{i,j}$. After putting all the samples of class i into one set, we can represent it as $y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,J}]^T$. Then, as shown in formula (4), the jth sample of class i can be represented by a combination of public and private features, thus greatly reducing the required storage space:

$$y_{i,j} = z_i^c + z_{i,j}^i, \tag{4}$$

where z_i^c is the public feature of all samples in class i and $z_{i,j}^i$ is the private feature of the jth sample of class i [13]. Assuming that the samples can be sparsely represented on the orthogonal basis $\Psi \in R^{N \times N}$, formula (4) can be expressed as

$$\theta_{i,j} = \Psi y_{i,j} = \Psi z_i^c + \Psi z_{i,j}^i = \theta_i^c + \theta_{i,j}^i, \tag{5}$$

where $\theta_i^c = \Psi z_i^c$ represents the sparse representation of the public part on Ψ and $\theta_{i,j}^i = \Psi z_{i,j}^i$ represents the sparse representation of the private part on Ψ. Through left multiplying Ψ^T, $\Psi^T \theta_{i,j} = \Psi^T \theta_i^c + \Psi^T \theta_{i,j}^i = z_i^c + z_{i,j}^i = y_{i,j}$, the images of class i can be represented as

$$\begin{bmatrix} y_{i,1} \\ y_{i,2} \\ \vdots \\ \vdots \\ y_{i,J} \end{bmatrix} = \begin{bmatrix} \Psi^T & \Psi^T & 0 & \cdots & 0 \\ \Psi^T & 0 & \Psi^T & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ \Psi^T & 0 & \cdots & 0 & \Psi^T \end{bmatrix} \cdot \begin{bmatrix} \theta_i^c \\ \theta_{i,1}^i \\ \theta_{i,2}^i \\ \vdots \\ \vdots \\ \theta_{i,J}^i \end{bmatrix}. \tag{6}$$

After simplifying, formula (6) can be expressed as

$$y_i = \tilde{\Psi} W_i, \tag{7}$$

where $y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,j}]^T$, $W_i = \left[\theta_i^c \ \theta_{i,1}^i \ \theta_{i,2}^i \ldots \theta_{i,j}^i\right]^T$, and $\tilde{\Psi} = [A, B]$ represents an overcomplete dictionary that contains two parts: $A = \left[\Psi^T \ \Psi^T \ \ldots \ \Psi^T\right]$ and $B = \mathrm{diag}(A)$. W_i can be obtained by solving the l_1 minimization problem as follows:

$$\begin{aligned} W_i &= \arg\min \|W_i\|_1, \\ \text{s.t.} \quad y_i &= \tilde{\Psi} W_i. \end{aligned} \tag{8}$$

After obtaining W_i, according to the inverse transformation, the public features of all images of class i and the private features of each image in the Ψ domain can be obtained as

$$\begin{aligned} z_i^c &= \Psi^T \theta_i^c, \\ z_{i,j}^i &= \Psi^T \theta_{i,j}^i. \end{aligned} \tag{9}$$

Combining all public and private features can get the joint feature dictionary D:

$$D = \left[z_1^c, z_2^c, \ldots, z_K^c, z_{1,1}^1, \ldots, z_{1,J}^1, z_{2,1}^2, \ldots, z_{2,J}^2, \ldots, \ldots, z_{K,1}^K, \ldots z_{K,J}^K\right]. \tag{10}$$

Finally, according to the sparse representation classification method, the target can be classified by the following formula:

$$\text{class}(i) = \arg\min_i \left\| y - D\delta_i(x') \right\|_2, \tag{11}$$

where x' represents the sparse coefficient vector that can be reconstructed from y with the dictionary.

3.2. Adaptive Weighted Reconstruction. When using SRC, the information carried by atoms in different dictionaries is mainly used to sparse reconstruction. Therefore, in order to improve the recognition accuracy, the atoms with more target information can be screened out by calculating the variance or standard deviation. And, the contribution ability of these atoms can be artificially improved to make the dictionary more discriminant [14].

Suppose $F = [F_1, F_2, \ldots, F_n]^T$ is a vector which extracted from an image, and then it can be modified by the following formula:

$$F' = \left[F_1' = \frac{\left|F_1 - \overline{F}\right|}{\overline{F}} F_1, F_2' = \frac{\left|F_2 - \overline{F}\right|}{\overline{F}} F_2, \cdots, F_n' = \frac{\left|F_n - \overline{F}\right|}{\overline{F}} F_n \right], \tag{12}$$

where $\overline{F} = (F_1 + F_2 + \cdots + F_n)/n$, F_i'represents the *ith* feature after weighted reconstruction. After the above processing, the variance between the feature vectors will increase to a certain extent. The feature dictionary contains more recognition information, which can improve the discrimination ability of the dictionary.

4. Framework of Mixed Feature-Based Joint Sparse Representation (M-JCR)

The algorithm framework is shown in Figure 3. First, Gabor wavelet features and deep features are combined into mixed features. Then, the joint sparsity model is used to extract public feature and private feature to build joint dictionary, and the test samples are sparse reconstructed. Finally, the target can be identified on the basis of the minimum reconstruction error criterion.

The specific steps of M-JSR are as follows:

(1) Gabor wavelet transform is used to extract Gabor wavelet features of training images and test images, and CNN is used to extract deep features of training images and test images.

(2) The Gabor wavelet feature and deep feature are adaptively weighted to form the mixed feature set, and the mixed feature is dimensionally reduced by PCA.

(3) The public feature of each class and the private feature of each image are extracted from the training image feature set. The public features are formed into a matrix M, and all private features are arranged into a matrix N to form a joint feature dictionary$D = [M, N]$, as shown in Formula (10).

(4) The mixed feature vector of the test image is sparsely represented on the joint feature dictionary to get the sparse coefficient$x\prime$, and the mixed feature vector of the test image is reconstructed.

(5) Finally, the recognition result is obtained through Formula (11).

5. Experiments and Analysis

In this paper, M-JSR is verified on face images, AR data set, and remote sensing images, respectively. The platform used in the experiment is Matlab R2017a. The computer is configured as Intel Core i5-3210M@2.5 GHz, and the memory is 4 GB. The experimental results are the average values of 10 experiments.

5.1. Face Image Recognition. In this part, two face datasets of AR [15] and Extended YaleB [16] are selected, and our experiment results are compared with SRC [17], extended SRC (ESRC) [18], low-rank matrix recovery method (LR) [19], discriminative low-rank representation method (DLRR) [20], sparse dictionary decomposition method (SDD) [21], adaptive weighting joint sparse representation method (AJSR) [14], and deep feature-based adaptive joint sparse representation(D-AJSR) [22], respectively.

5.1.1. AR Dataset. The AR dataset contains more than 4000 positive images, belonging to 126 individuals, with the image size of 120×165. In the experiments, we use a subset of 100 people, 50 men and 50 women, and there are 26 positive images of each person. Among them, 14 images are no blocking images with only changes in expression or light. 6 people wear sunglasses, and 6 people wear scarves. Therefore, the dataset can be divided into two separate parts, and each part contains 13 pictures (7 positive pictures with no blocking and only changes in expression or light, 3 facial pictures with sunglasses, and 3 positive pictures with scarves). Figure 4 shows some sample images in the AR dataset. We randomly select one part for training and the other for testing. The Gabor wavelet features used in the experiments include 5 scales and 40 features in 8 directions. The deep features used are from the convolution layer in the second part of VGG19, and the number is 128. After PCA dimension reduction, the feature dimensions are 25, 50, 75, 100, and 150.

The experimental results are shown in Table 1. The bold number in each column represents the highest recognition rate under the same condition. Although the recognition rate of M-JSR is not the highest when the dimension is 25, it also remains at the average level. When the dimension is above 50, the recognition rate of M-JSR is higher than that of other methods.

5.1.2. Extended YaleB Dataset. The Extended YaleB dataset consists of 2,414 positive images of size 168×192, in which there are 38 people under different lighting conditions.

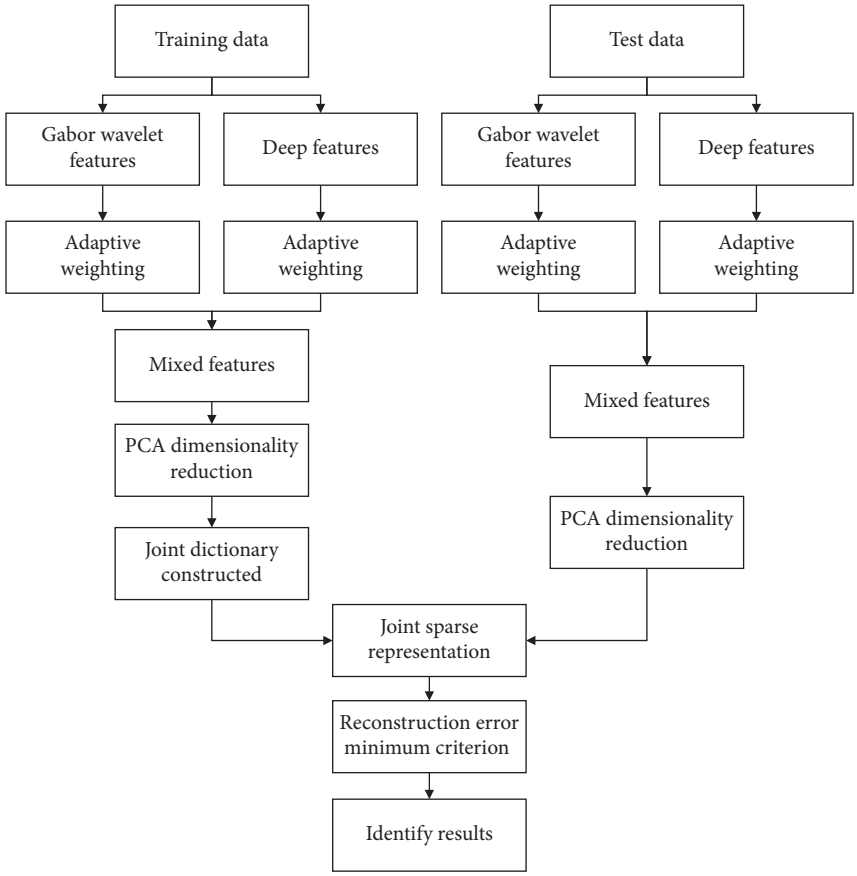


Figure 3: The algorithm framework of M-JSR.

Figure 4: Samples in the AR dataset.

Table 1: Recognition rates (%) on the AR dataset.

Dimensions	25	50	75	100	150
SRC [17]	64.29	81.29	88.43	89.29	90.29
ESRC [18]	63.14	80.43	85.43	86.14	87.29
LR [19]	68.57	84.14	86.00	88.71	88.00
DLRR [20]	75.71	88.14	89.43	91.00	91.86
SDD [21]	75.86	87.29	89.71	91.71	93.00
D-AJSR [22]	67.10	86.00	90.70	94.10	95.10
M-JSR	71.00	88.20	94.60	96.00	96.80

Figure 5 shows some sample images from the Extended YaleB dataset. In the experiments, we randomly selected 16 images of each person for training and the rest for testing. The Gabor wavelet features used in the experiment include 5 scales and 40 features in 8 directions. The deep features used are from the convolution layer in the second part of VGG19, and the number is 128. After PCA dimension reduction, the feature dimensions are also 25, 50, 75, 100, and 150.

The experimental results are shown in Table 2. The bold number in each column represents the highest recognition rate under the same condition. The M-JSR method maintains high accuracy rates in all dimensions, only slightly lower than D-AJSR in 50 and 75 dimensions. Compared with the AR dataset, the recognition rates are relatively higher because there is no image with sunglasses and scarf.

5.2. Remote Sensing Image Recognition Experiments. In this part, we download the remote sensing aircraft images of different shooting times and locations on Google Earth 7.1.8 as the experimental dataset. In the dataset, 375 remote sensing images are classified to 15 aircraft types, as shown in Figure 6. 10 images in each aircraft type are randomly

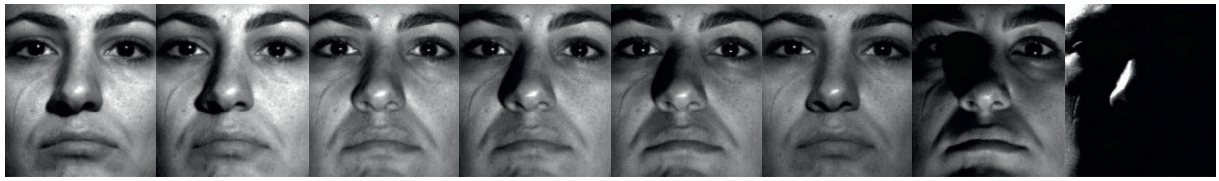


Figure 5: Samples in extended YaleB dataset.

Table 2: Recognition rates (%) on extended YaleB dataset.

Dimensions	25	50	75	100	150
SRC [17]	72.98	85.22	88.43	90.48	92.30
ESRC [18]	73.86	85.33	88.37	90.20	91.20
LR [19]	75.97	84.39	88.21	89.09	91.14
DLRR [20]	85.44	89.81	89.92	92.25	93.05
SDD [21]	89.70	92.03	92.41	92.69	92.75
D-AJSR [22]	93.16	96.05	96.84	96.58	97.37
M-JSR	93.42	95.00	96.E68	97.36	97.63

Figure 6: Examples of remote sensing aircraft images.

selected for training and 15 for testing. The image size is 170×170. The Gabor wavelet features used in the experiment include 5 scales and 40 features in 8 directions. The deep features used are from the first part of VGG19, and the number is 64. After PCA dimension reduction, the feature dimensions are also 25, 50, 75, and 100. The experiment results are shown in Table 3. The bold number in each column represents the highest recognition rate under the same condition.

It can be seen from Table 3 that M-JSR has better effect than other methods. This is because the addition of Gabor wavelet feature can provide more information in different directions. However, compared with the recognition rates of face images, the recognition rates are relatively lower. It is mainly because many planes leave shadows on the side due to the slanting sun. As a result, the contour of two planes will appear on the feature map when the image feature is extracted, which has great interference to the subsequent recognition.

Table 3: Recognition rate (%) of remote sensing aircraft images.

Dimensions	25	50	75	100
SRC [17]	62.00	63.56	65.33	66.00
AJRC [14]	70.62	72.00	76.67	78.67
D-AJSR [22]	71.33	75.53	77.33	80.65
M-JSR	74.25	78.67	82.00	82.67

5.3. Comprehensive Analysis of Experiments. In the experiment, when PCA was used in dimensionality reduction, the cumulative variance contribution rates of the 3 datasets were

Table 4: Cumulative variance contribution rates (%) on different datasets.

Dimensions	25	50	75	100	150
AR [15]	45.32	55.16	57.42	61.20	69.04
Extended YaleB [16]	42.90	59.58	67.91	73.84	82.37
Remote sensing data set	43.42	61.80	75.03	85.45	—

Table 5: Training efficiency (s)of different datasets.

Dimensions	25	50	75	100	150
AR [15]	609.150	689.515	813.090	1077.03	1420.16
Extended YaleB [16]	326.109	366.662	409.921	519.172	645.442

Table 6: Test efficiency (s) of different datasets.

Dimensions	25	50	75	100	150
AR [15]	1105.84	1273.50	1497.33	1817.91	2899.68
Extended YaleB [16]	642.385	674.836	694.840	749.198	850.541

also different, as shown in Table 4. It can be seen that the cumulative variance contribution rates of M-JSR on all datasets is low. The reason is M-JSR uses the mixed features which composed of Gabor wavelet features and deep features, so the energy of feature vectors would not be concentrated during PCA dimensionality reduction. Relatively speaking, the fewer principal components are selected, the lower the cumulative variance contribution rate will be. At the same time, the recognition rates of M-JSR are also low when the feature dimension is low.

In addition to the contribution rates of the cumulative variance, the time efficiency of M-JSR is also calculated on 3 datasets, respectively. The training efficiency results of AR dataset and Extended YaleB dataset are shown in Table 5, and the test efficiency results are shown in Table 6. The unit of time is seconds (s). In these experiments, the images of the AR dataset is more than those of the YaleB dataset, so that the training time and test time required for the AR dataset are more than that of the Extended YaleB dataset.

On the remote sensing dataset, the time efficiency of M-JSR is compared with that of SRC, AJRC, and D-AJSR. The training efficiency results are shown in Table 7, and the test efficiency results are shown in Table 8. The unit of time is seconds (s). As can be seen from Table 7 and Table 8, since M-JSR needs to extract two types of features, it takes more training time and more testing time than the other methods. However, considering the recognition rate, we still think the M-JSR method has its own advantages.

It can be seen from the previous experiments that M-JSR has a good robustness for the illumination change and rotation of the image because of the combination of Gabor wavelet features and deep features. Moreover, when the dataset is small, satisfactory recognition results can also be obtained. In many cases, it is difficult to obtain a large number of target images, and the image quality is generally poor due to the influence of dim light, distortion, and other factors. In this case, M-JSR can also provide accurate identification results.

Table 7: Training efficiency (s) of different methods on remote sensing dataset.

Dimensions	25	50	75	100
SRC [17]	1.2649	1.2901	1.2758	1.2833
AJRC [14]	49.734	58.775	78.598	115.08
D-AJSR [22]	63.104	72.078	94.864	128.94
M-JSR	74.053	82.471	101.49	136.11

Table 8: Test efficiency (s)of different methods on remote sensing dataset.

Dimensions	25	50	75	100
SRC [17]	4.1456	7.4306	8.1706	9.4669
AJRC [14]	105.14	108.93	113.11	117.32
D-AJSR [22]	121.00	131.29	132.51	134.70
M-JSR	135.62	138.54	142.97	146.77

6. Conclusions

For the application requirements of image target recognition, Gabor wavelet features and deep features are introduced into JSR in this paper. The classification framework (M-JSR) has good robustness for deformation, rotation, and light and shade change and can get relatively accurate recognition results with only a few training samples. In M-JSR, two kinds of features are composed into mixed features, in which the weights can be adjusted adaptively. The joint sparse model divides the feature dictionary into public part and private part, which reduces the required storage space and improves the recognition accuracy of the image target. However, because M-JSR needs to extract two characteristics, it takes more time than other methods. Therefore, in the future research, how to take into account the feature expressiveness and extraction speed is a problem that needs to be paid attention. Using lightweight networks [23] for feature extraction is an effective approach.

Data Availability

All datasets in this article are public datasets and can be found on public websites.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the National Defense Pre-Research Foundation of China under Grant 9140A01060314KG01018, National Natural Science Foundation of China under Grant 61471370, Equipment Exploration Research Project of China under Grant 71314092, Scientific Research Fund of Hunan Provincial Education Department under Grant 17C0043,

Hunan Provincial Natural Science Fund under Grant 2019JJ80105, Changsha Science and Technology Project under Grant 29312, and Hunan Graduate Scientific Research Innovation Project under Grant CX20200882.

References

[1] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[2] Z. Lu and L. L. Zhang, "Face recognition algorithm based on discriminative dictionary learning and sparse representation," *Neurocomputing*, vol. 174, pp. 749–755, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, vol. 60, pp. 1097–1105, Lake Tahoe, Nevada, USA, January 2012.

[4] S. Ren, K. He, R. Girshick, J. Sun, and R.-C. N. N. Faster, "Towards real-time object detection with region proposal networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, vol. 39, pp. 91–99, Montreal, Canada, 2015.

[5] J. Zhang, S. Kun, and L. Xing, "Small sample image recognition using improved Convolutional Neural Network," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 640–647, 2018.

[6] W. Wei, J. Yongbin, L. Yanhong, L. Ji, W. Xin, and Z. Tong, "An advanced deep residual dense network (DRDN) approach for image super-resolution," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592–1601, 2019.

[7] C. Wang, L. Yun, and Z. Li, "Algorithm research of face image gender classification based on2-D gabor wavelet transform and SVM international symposium on computer science and computational Technology," in *Proceedings of 2008 International Symposium on Computer Science and Computational Technology*, pp. 312–315, IEEE, Shanghai, China, December 2008.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] C. Zhang and J. Tian, "A SAR image targets recognition approach via novel SSF-net models," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8859172, 9 pages, 2020.

[10] W. Wang, Y. Yang, and X. Wang, "Development of convolutional neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, Article ID 040901, 2019.

[11] W. Wang, C. Zhang, and J. Tian, "High resolution radar targets recognition via inception-based VGG (IVGG) networks," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8893419, 11 pages, 2020.

[12] D. Baron, M. F. Duarte, and M. B. Wakin, "Distributed compressive sensing," in *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, IEEE, Taipei, Taiwan, April 2009.

[13] P. Nagesh and B. Li, "A compressive sensing approach for expression-invariant face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1518–1525, IEEE, Miami, FL, USA, June 2009.

[14] W. Wang, J. Chen, J. Li, and X. Wang, "Remote targets recognition based on adaptive weighting feature dictionaries and joint sparse representations," *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 11, pp. 1863–1870, 2018.

[15] A. Martínez and R. Benavente, "The AR face database," *CVC Technical Report*, vol. 24, 1998.

[16] K. C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[17] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[18] W. Deng, J. Hu, and J. Guo, "Extended SRC: undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–1870, 2012.

[19] C. Chen, C. Wei, and Y. Wang, "Low-rank matrix recovery with structural incoherence for robust face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2618–2625, IEEE, Providence, RI, USA, June 2012.

[20] J. Chen and Z. Yi, "Sparse representation for face recognition by discriminative low-rank matrix recovery," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 763–773, 2014.

[21] F. Cao, X. Feng, and J. Zhao, "Sparse representation for robust face recognition by dictionary decomposition," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 260–268, 2017.

[22] W. Wei, T. Can, W. Xin, L. Yanhong, H. Yongle, and L. Ji, "Image object recognition via deep feature-based adaptive joint sparse representation," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8258275, 9 pages, 2019.

[23] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-MobileNet models," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8879795, 11 pages
https://doi.org/10.1155/2020/8879795

Research Article

A Compressive Sensing Model for Speeding Up Text Classification

Kelin Shen,[1] **Peinan Hao**,[2,3] **and Ran Li**[2,3]

[1]*School of Foreign Languages, Xinyang Agriculture and Forestry University, Xinyang 46400, China*
[2]*School of Computer and Information Technology, Xinyang Normal University, Xinyang 46400, China*
[3]*Henan Key Lab of Analysis and Applications of Education Big Data, Xinyang 46400, China*

Correspondence should be addressed to Kelin Shen; shenkl@xynu.edu.cn

Received 25 June 2020; Revised 7 July 2020; Accepted 18 July 2020; Published 7 August 2020

Academic Editor: Nian Zhang

Text classification plays an important role in various applications of big data by automatically classifying massive text documents. However, high dimensionality and sparsity of text features have presented a challenge to efficient classification. In this paper, we propose a compressive sensing- (CS-) based model to speed up text classification. Using CS to reduce the size of feature space, our model has a low time and space complexity while training a text classifier, and the restricted isometry property (RIP) of CS ensures that pairwise distances between text features can be well preserved in the process of dimensionality reduction. In particular, by structural random matrices (SRMs), CS is free from computation and memory limitations in the construction of random projections. Experimental results demonstrate that CS effectively accelerates the text classification while hardly causing any accuracy loss.

1. Introduction

With the advancement of information technology over the last decade, digital resources have penetrated into all fields in our society, generating big data, which present a new challenge to data mining and information retrieval [1]. Texts are very common in daily life, and, with their large numbers, it remains an open question to organize and manage them [2]. As one of the fundamental techniques in natural language processing (NLP), text classification means assigning labels or categories to texts according to the content, and it is key to solving the problem of text overloads [3]. In its broad applications such as sentiment analysis, topic labeling, spam detection, and intent detection, text classification provides support for the efficient query and search of texts, attracting a lot of attention from both academia and industry [4, 5].

Word matching (WM), the simplest method in text classification, determines the category of a text by the categories of most words in the text [6]. But, due to the ambiguity of word meaning, WM fails to provide satisfying accuracy. By representing words as vectors, the vector space model (VSM) [7] improves the accuracy of text classification, thus replacing WM as the popular method, but the model requires many rules and great efforts from professionals in labeling texts, which would be a lot of cost. As machine learning (ML) [8] continues to develop, the accuracy of text classification has been further improved. By extracting features from a text to train a classifier, ML reforms VSM and avoids the rule-based inference. Recently, the rapidly developing deep learning (DL) [9], which is a branch of ML, has made text classification more efficient. However, high dimensionality and sparsity of text features pose a challenge to ML, restricting the practical use of ML-based text classification.

In ML, many classifiers can be used to classify texts, such as support vector machine (SVM) [10], decision tree [11], adaptive boosting (AdaBoost) [12], K-nearest neighbor (KNN) [13], and Naïve Bayes [14]. To train these classifiers, texts must be represented as feature vectors by some feature extraction models, among which the commonest is Bag of Words (BOW) [15]. BOW uses the term frequencies of n-grams in the vocabulary constructed by N-Gram [16] to encode every text. Because vocabulary may potentially run into millions, BOW faces the curse of dimensionality; that is, it produces a sparse representation with a huge dimensionality, resulting in the impracticality of training

classifiers. Therefore, dimensionality reduction (DR) is used to reduce the size of feature space. In DR, the most common techniques still introduce some time and memory complexity due to their nature of supervised learning, including principal component analysis (PCA) [17], independent component analysis (ICA) [18], and nonnegative matrix factorization (NMF) [19]. Many DL networks use autoencoder to compress the size of parameters. An autoencoder is a neural network that is trained to attempt to copy its input. Some popular architectures include sparse autoencoder [20], denoising autoencoder [21], and variational autoencoder [22]. Internally, they have a hidden layer that describes a code used to represent the input. By being embedded into the neural network, the autoencoder can end up learning a low-dimensional representation very similar to PCAs.

Compared with the above-mentioned DR techniques, random projection [23, 24] is a better choice, since it avoids the model training, but it is still a challenge to store random projections due to the huge dimensionality of text feature. Compressive sensing (CS) [25–27], which has recently been rapidly developing, can be regarded as a random projection technique specially for sparse vectors, and it proves that the perfect recovery of sparse vector can be realized by several random projections. CS retains the advantages of random projection in DR and further overcomes the problem of memory with the help of structural random matrices (SRMs) [28, 29], which makes CS a potential DR technique for text classification. In view of the merits of CS, we use it to speed up the training of text classifiers in this paper. For a low time and memory complexity, SRMs are selected as CS measurement matrices to reduce the size of sparse feature vector. Experimental results demonstrate that CS effectively accelerates the text classification while hardly causing any accuracy loss.

The rest of this paper is organized as follows. Section 2 briefly reviews text classification and CS theory. Section 3 describes the CS model for text classification in detail. Section 4 presents experimental results, and finally Section 5 concludes this paper.

2. Background

2.1. Text Classification. Given a text dataset $D = \{d_1, d_2, \ldots, d_L\}$ of L documents and a set $C = \{c_1, c_2, \ldots, c_J\}$ of J predefined categories, the goal of text classification is to learn a mapping f from inputs $d_i \in D$ to outputs $c_j \in C$. If $J = 2$, it is called binary classification; if $J > 2$, it is called multiclass classification. The mapping f is called the classifier, and it is trained by being fed with a labeled dataset, where each document in D has been assigned a category from C by professionals in advance. The trained classifier f is used to make predictions on new documents which are not included in D. Because of the subjectivity of text labeling, a test dataset is still needed to evaluate the prediction accuracy of f.

A typical flow of text classification is illustrated in Figure 1. In text preprocessing, we tokenize each document in D, erase punctuations, and remove unnecessary words such as stop words, misspelling, and slang. To reduce the size of vocabulary from D, some operations, e.g., capitalization, lemmatization, and stemming, can also be added. After text preprocessing,

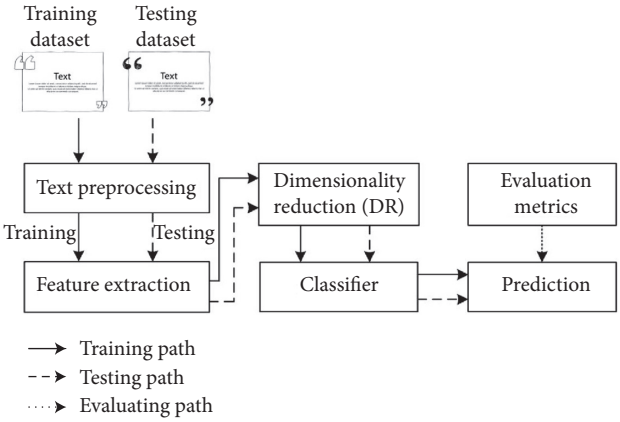


Figure 1: A typical flow of text classification.

feature extraction is performed to represent documents in D as feature vectors, which is a crucial step for the accuracy and complexity of text classification. By N-Gram, we collect n-grams from D as the vocabulary of BOW model. It is very common to use unigram and bigram, where unigram is a single word and bigram is a word pair. Each document in D is encoded as a feature vector based on the frequency distribution of its n-grams on the BOW vocabulary. The size of feature vector is the same as that of BOW vocabulary, resulting in the huge dimensionality of feature space. By using DR techniques, dimensionality can be significantly decreased, reducing the time complexity and memory consumption when training the classifier. The feature vector of a document is also highly sparse because the number of its n-grams is far smaller than the size of BOW vocabulary. The high sparsity makes it possible to realize DR by CS without the loss of classification accuracy. Compared with the traditional DR methods, CS not only avoids the computations invested in supervised learning but also reduces the memory burden for constructing random projections. In this paper, we use CS to reduce the feature dimensionality and try to prove its efficiency of speeding up text classification.

2.2. Compressive Sensing. CS is a novel sampling paradigm that goes against the traditional Nyquist/Shannon theorem, and it shows that a signal can be recovered precisely from only a small set of samples. The success of CS relies on two principles: sparsity and incoherence, where the former defines an S-sparse signal $\boldsymbol{s}$ in R^N with all but the S entries set to be zero, and the latter highlights the incoherent measure vectors $\{\phi_i \in R^N\}_{i=1}^M$ with $\boldsymbol{s}$. The following briefly describes the CS framework.

By ordering these measure vectors in column, a measurement matrix $\Phi \in R^{M \times N}$ is constructed as follows:

$$\boldsymbol{\Phi} = \begin{bmatrix} — & \boldsymbol{\varphi}_1 & — \\ & \vdots & \\ — & \boldsymbol{\varphi}_i & — \\ & \vdots & \\ — & \boldsymbol{\varphi}_M & — \end{bmatrix}. \tag{1}$$

By using Φ to linearly measure s, we obtain the sampled vector $\boldsymbol{y} \in R^M$ by

$$\mathbf{y} = \boldsymbol{\Phi} \cdot \mathbf{s}. \tag{2}$$

We define the ratio of M/N as the subrate R; that is, $R = M/N$, and DR is realized by setting R to be less than 1, but it also becomes an ill-posed problem to find s from y. Based on the sparsity property of $\boldsymbol{s}$, this problem can be solved by an optimizing model:

$$\begin{aligned} \hat{\mathbf{s}} &= \arg\min_{\mathbf{s}} \|\mathbf{s}\|_0 \\ \text{s.t.} \mathbf{y} &= \boldsymbol{\Phi} \cdot \mathbf{s}, \end{aligned} \tag{3}$$

where $||\cdot||_0$ represents l_0 norm to count the number of nonzero entries in s, and the solution $\hat{s}$ is an estimate of s. The incoherence between φ_i and $\boldsymbol{s}$ has an effect on the convergence of the solution $\hat{s}$ to the original s, which presents a challenge for CS, that is, how to construct incoherence measurement vectors. Fortunately, it is found that random vectors are largely incoherent with any fixed signal, so Φ can be produced by some random distributions, for example, Gaussian, Bernoulli, and uniform.

By performing incoherent measuring with random matrices, CS can be categorized as the random projection technology in DR. In particular, in order to enhance the robustness of recovery, CS requires Φ to further hold the restricted isometry property (RIP) for S-sparse signals. When RIP holds, Φ preserves the approximate Euclidean length of S-sparse signals, which implies that all pairwise distances between S-sparse signals can be well preserved in the measurement space. In text classification, the feature vectors of documents in text dataset are highly sparse, so RIP of CS can significantly reduce feature dimensionality while preserving pairwise distances between feature vectors. Superior to traditional DR methods, CS ensures less memory consumption and faster computing by SRMs. In view of the merits of CS, we explore CS features extracted by SRMs to speed up text classification.

3. Proposed CS-Based Text Classification

3.1. Framework Description. Figure 2 presents the framework of the proposed CS-based text classification. After text preprocessing, the text dataset is divided into training dataset P and testing dataset Q, where the former is used to train classifiers, and the latter is used to evaluate the classification accuracy. The core of our work is to extract CS features to represent documents in text dataset. In CS feature extraction, we represent each document p_i in the training dataset P as the highly sparse vector x_i by BOW and construct an SRM $\Phi \in R^{M \times N}$ to linearly measure x_i, producing the CS feature vector y_i of x_i. CS feature is a low-dimensional and dense vector, which can shorten the time of training classifier, especially for a large-scale text dataset. In the following parts, we describe, respectively, CS feature extraction, SRMs construction, and classifiers in detail.

3.2. CS Feature Extraction. We collect unigrams and bigrams from the training dataset P to create the vocabulary of BOW model. Unigrams are single words from P, and most of them occur very few times to impact classification, so we only add top N_1 words from these unigrams to the BOW vocabulary. Bigrams are word pairs from P, and they are a good way to model negation like "not good." The total amount of bigrams is very big, but most of them are noise at the end of frequency spectrum, so we use top N_2 word pairs from these bigrams, adding them to the BOW vocabulary. In the experiment part, we set suitable N_1 and N_2 for different classification tasks.

After collecting unigrams and bigrams, we convert each document p_i in P into the feature vector $\boldsymbol{x}_i$ in sparse representation. The BOW feature x_i is the frequency distribution of p_i on the BOW vocabulary, and its size is N, which is the sum of N_1 and N_2. All BOW features consist of a feature matrix X as follows:

$$\mathbf{X} = \begin{bmatrix} | & & | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_i & \cdots & \mathbf{x}_{L_1} \\ | & & | & & | \end{bmatrix}, \tag{4}$$

where L_1 is the amount of P. In the ordinary classification, X is input into the classifier to train it. Being a large size, X results in the curse of dimensionality; for example, when N and L_1 are set to be 25000 and 800000, respectively, the size of X is 25000×800000, and it needs a memory of 8×10^{10} bytes ($\approx$75 GB) assuming that 4 bytes encode each entry in X. That would lead to a heavy computational burden, so we reduce the size of X by CS measuring as follows:

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} | & & | & & | \\ \mathbf{y}_1 & \cdots & \mathbf{y}_i & \cdots & \mathbf{y}_{L_1} \\ | & & | & & | \end{bmatrix} \\ &= \begin{bmatrix} | & & | & & | \\ \boldsymbol{\Phi}\mathbf{x}_1 & \cdots & \boldsymbol{\Phi}\mathbf{x}_i & \cdots & \boldsymbol{\Phi}\mathbf{x}_{L_1} \\ | & & | & & | \end{bmatrix} \\ &= \boldsymbol{\Phi} \cdot \mathbf{X}, \end{aligned} \tag{5}$$

where $\Phi \in R^{M \times N}$ is a CS measurement matrix and $Y \in R^{M \times L1}$ is the CS feature matrix, of which the i-th column y_i is the CS feature vector of the i-th document p_i in the training dataset P.

To precisely recover signals, the CS measurement matrix is required to hold RIP. In practice, a random matrix, e.g., produced by Gaussian or Bernoulli distribution, obeys RIP for S-sparse signal provided that

$$M \geq 4 \cdot S, \tag{6}$$

is satisfied [30]. M can be set to be far smaller than N since BOW features are highly sparse, so the size of Y can be significantly reduced. Importantly, RIP can be enforced or degraded by widening or reducing the gap between M and S; that is, when M is far larger than $4 \cdot S$, the pairwise distances between S-sparse signals are well preserved in the CS feature

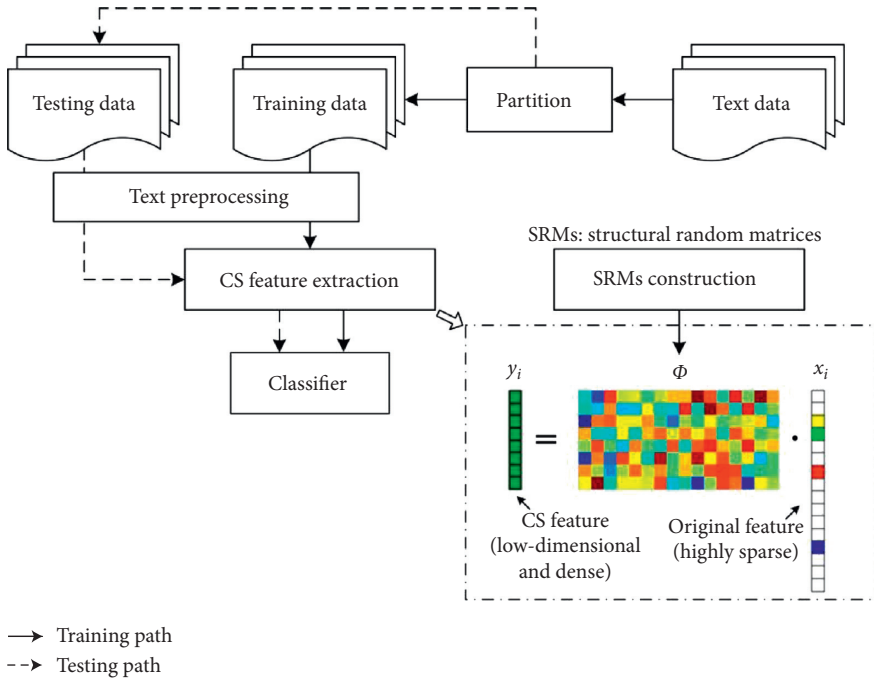


Figure 2: Framework of CS-based text classification.

space, and these pairwise distances can be destroyed when gradually reducing M, so the subrate R becomes a key factor impacting the accuracy of text classification. In the experiment part, we will evaluate the effects of different R values on pairwise distances between features and the accuracy of classification. In general, these random projections are dense, and a common computer does not have sufficient memory to store them, so CS-based DR is not applicable to a large-scale dataset if traditional method is used to produce the random projections. However, CS offers some measurement matrices for large-scale and real-time applications, among which the most famous is SRMs. The following describes how to construct SRMs, so as to make CS-based DR feasible for a large-scale dataset.

3.3. SRMs Construction. SRM, proposed by Do et al. [28], is a known sensing framework in the field of CS. With its fast and efficient implementation, it brings some benefits to CS-based DR, for example, low complexity, fast computation, block-based processing support, and optimal incoherence. By using SRMs, with less memory consumption, the length of BOW feature can be fast and greatly reduced while holding RIP.

SRM is defined as a product of three matrices; that is,

$$\mathbf{\Phi} = \sqrt{\frac{N}{M}}\mathbf{D} \cdot \mathbf{F} \cdot \mathbf{E}, \tag{7}$$

where $E \in R^{N\times N}$ is a random permutation matrix that uniformly permutes the locations of vector entries globally, $F \in R^{N\times N}$ is an orthonormal matrix constructed by popular fast computable transform, e.g., Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Walsh-Hadamard Transform (WHT), or their block diagonal versions, $D \in R^{M\times N}$ is a random subset of M rows of the identity matrix of $N \times N$ in size to subsample the input vector, and $\sqrt{N/M}$ is a scale to normalize the transform so that the energy of the subsampled vector is almost similar to that of the input vector. By plugging (7) into (5), the matrix product $\Phi{\cdot}X$ can be performed according to a sensing algorithm as shown in Algorithm 1. The SRM sensing algorithm can be computed fast; that is, the computational complexity is typically in the order of $O(N)$ to $O(N\log N)$. Suppose that F is FFT or DCT matrix; the implementation of SRM takes $O(N\log N)$ operations. SRM is used to measure L_1 BOW features one by one, which takes $O(L_1 N\log N)$ operations; that is, the total computational complexity of the proposed CS model is $O(L_1 N\log N)$. Compared with existing random projection techniques, SRMs not only cost less time and space complexity, but they also convert the sampled vector into a white noise-like one by scrambling the vector structure to achieve universal incoherence. Therefore, SRMs can make CS-based text classification more efficient.

3.4. Classifiers. Many popular classifiers can be used in our model, e.g., SVM, decision tree, AdaBoost, KNN, and Naïve Bayes. In the experiment part, these classifiers are applied and their classification accuracy is evaluated to verify the efficiency of our model. This section reviews these popular classifiers in text classification.

SVM [10] is a nonprobabilistic linear binary classifier. For a training set of points (y_i, l_i), where $\boldsymbol{y}_i$ is the CS feature vector and l_i is the category of the document d_i, we try to find the maximum-margin hyperplane that divides the points with $l_i = 1$ and $l_i = -1$. The equation of the hyperplane is as follows:

Task: Perform $\boldsymbol{\Phi}\cdot X$ in which $\boldsymbol{\Phi}$ is one of SRMs
Input: The BOW feature matrix $X = [x_1, \ldots, x_i, \ldots, x_{L1}]$, the measurement number M, and a fast transform operator $F(\cdot)$.
Main iteration: Iterate on i until $i > L_1$ is satisfied.
(1) Pre-randomization: randomize x_i by uniformly permuting its sample locations. This step corresponds to multiplying x_i with E.
(2) Transform: apply a fast transform $F(\cdot)$ to the randomized vector, e.g, FFT, DCT, etc.
(3) Subsampling: randomly pick up M samples out of N transform coefficients. This step corresponds to multiplying the transform coefficients with D.
Output: The CS feature matrix $Y = [y_1, \ldots, y_i, \ldots, y_{L1}]$.

Algorithm 1: Flow of SRM sensing algorithm.

$$\mathbf{w}^{\mathrm{T}}\mathbf{y} + b = 0. \tag{8}$$

We maximize the margin, denoted by γ, as

$$\begin{aligned} &\max_{\mathbf{w},\boldsymbol{\gamma}} \boldsymbol{\gamma} \\ &\text{s.t. } \forall i, \boldsymbol{\gamma} \le l_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{y}_i + b\right), \end{aligned} \tag{9}$$

to separate the points well. By error-correcting output codes (ECOC) model [31], SVM can also undertake multiclass classification tasks.

Decision tree [11] is a classifier model in which each node of the tree represents a test on the attribute of the data set, its children represent the outcomes, and the leaf nodes represent the final categories of the data points. The training dataset is used to form the decision tree, and the best decision has to be made for each node in the tree. The decision tree can be fast trained, but it is also extremely sensitive to small perturbations in the dataset and can be easily overfit. By cross validation and pruning, these effects can be suppressed.

AdaBoost [12] extracts a classifier from the set of weak classifiers at each iteration and assigns a weight to the classifier according to its relevance. The weight in AdaBoost for each sample is measured according to how difficult previous classifiers have found it to get it correct. At each iteration, a new classifier is trained on the training dataset, and the weights are modified based on how successfully the training sample has been classified before. Training terminates after several iterations or when all training samples are classified correctly.

KNN [13] is a nonparametric technique used for classification. Given the CS feature y_i, KNN finds the K-nearest neighbors of y_i among all CS features in the training dataset and gives the category candidate a score based on the labels of the K neighbors. The similarity between y_i and its neighbor can be the score of the category of the neighbor features. After sorting the score values, KNN decides which category the candidate falls into with the highest score from y_i. KNN is easy to implement and adapts to any kind of feature space. It can also handle multiclass cases. The performance of KNN depends on finding some meaningful distance functions, and it is limited by data storage when finding the nearest neighbors for large search problems.

Naïve Bayes [14] has been widely used for text classification, and it is a generative model based on Bayes theorem. This model assumes that the value of a particular feature is independent of the value of any other feature. The proposed CS model is on the assumption that any entry in a CS feature vector is independent of other entries. Given a to-be-tested CS feature $\boldsymbol{y}$, its category is predicted as follows:

$$\widehat{l} = \arg\max_{l} p(l \mid \mathbf{y}). \tag{10}$$

According to Bayes inference, we see that

$$p(l \mid \mathbf{y}) \propto p(l) \prod_{m=1}^{M} p\left(y_m \mid l\right), \tag{11}$$

where y_m is the m-th entry in the CS feature y. The probabilities $p(l)$ and $p(y_m|l)$ can be estimated by maximum likelihood on the training dataset.

4. Experimental Results

4.1. Dataset and Setting. We conduct experiments on two datasets, one for a binary classification task and the other for a multiclass classification task. For the binary classification task, we use the Twitter sentiment dataset, which was crawled and labeled positive or negative. For the multiclass classification task, we use the weather report dataset that contains a text description and category labels for each event including thunderstorm wind, hail, flash flood, high wind, and winter weather. The classes of two datasets are imbalanced, especially for weather report dataset. To avoid the effects of imbalance on classification accuracy, the two datasets are preprocessed to make their classes balanced; i.e., for Twitter sentiment dataset, we randomly remove some positive and negative observations and make each class having 10000 observations; for weather report dataset, we delete the classes with few observations, and 9 classes remain: thunderstorm wind, hail, flash flood, high wind, winter weather, Marine Thunderstorm Wind, Winter Storm, Heavy Rain, and Flood, among which one has 1000 observations. Figure 3 presents the statistics of Twitter sentiment dataset and weather report dataset after balancing. For any dataset, 20% of observations in each class are set aside at random for testing. In feature extraction, we first do some preprocessing on documents in two datasets including the following: (1) tokenize the documents; (2) lemmatize the words; (3) erase punctuation; (4) remove a list of stop words such as "and," "of,", and "the"; (5) remove words with 2 or fewer characters; (6) remove words with 15 or more characters. Then, for both datasets, we, respectively, collect the

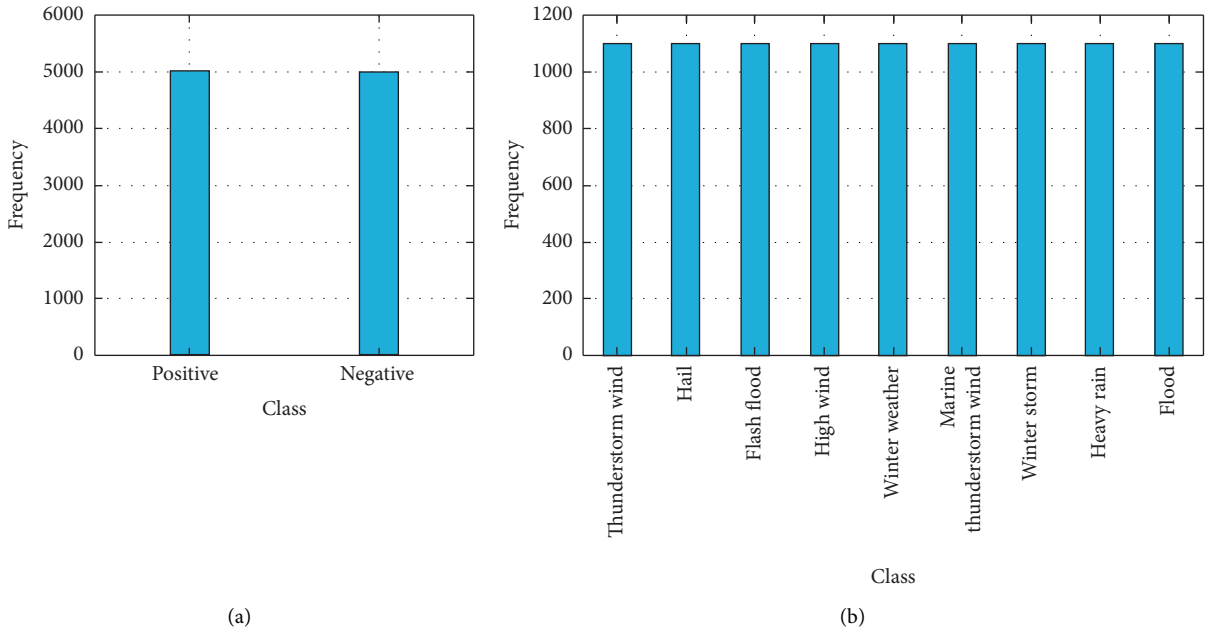


Figure 3: Statistics of the Twitter sentiment and weather report datasets after balancing. (a) Twitter sentiment dataset. (b) Weather report dataset.

top 8000 unigrams and 10000 bigrams from the training set to construct the BOW vocabulary, i.e., $N_1 = 8000$ and $N_2 = 10000$, and represent each training observation as the BOW feature vector with length of N being 18000. Finally, by setting different subrates, the SRMs are used to measure the BOW feature vectors, and the corresponding CS feature vectors are produced. We train different classifiers on the BOW-based and CS-based training sets, respectively, tune parameters by cross validation, and evaluate these classifiers on the test sets. Due to the random partition of dataset, the training and testing are repeated five times, and the mean testing accuracy is used as the evaluation metrics.

The experimental settings are as follows. To evaluate the effects of different SRMs on feature distance and classification accuracy, we construct five SRMs by using transform matrices F including DCT, FFT, Block DCT, Block WHT, and Block Gaussian, in which the latter three are block diagonal matrices, of which the diagonal elements are DCT and WHT and Gaussian matrices with the size of 32×32. We use six classifiers including SVM, decision tree, AdaBoost, KNN, and Naïve Bayes to evaluate the classification accuracy of our model and compare the proposed CS model with the three DR methods: PCA [17], ICA [18], and NMF [19]. The subrate R is set to be between 0.1 and 0.6, and it is preset parameter, which is used to decide the length of CS feature vector. All of the experiments are conducted under the following computer configuration: Intel(R) Core (TM) i7 @3.30 GHz CPU, 8 GB, RAM, Microsoft Windows 7 64 bits, and MATLAB Version 9.6.0. (R2019a). The datasets and experimental codes have been downloaded from SIGMULL Team Website: http://www.scholat.com/showTeamScholar.html?id=1234&changeTo=Ch&nav=4.

4.2. Effects of SRMs. Feature distance measures the similarity between any two documents, which has a significant impact on training accuracy. If the features output by DR can well preserve their pairwise distances in original space, DR suppresses the loss of training accuracy; therefore, we evaluate the effects of SRMs on pairwise distances between text features. In the training set P, the average distance between the i-th BOW or CS feature and others is computed as follows:

$$\text{dist}_i^{\text{BOW}} = \frac{1}{L_1} \sum_{j=1}^{L_1} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2, \tag{12}$$

$$\text{dist}_i^{\text{CS}} = \frac{1}{L_1} \sum_{j=1}^{L_1} \left\| \mathbf{y}_i - \mathbf{y}_j \right\|_2, \tag{13}$$

where x_i and y_i are, respectively, the i-th BOW and CS feature vector in P and L_1 is the amount of P. We select Block DCT as the core of SRM and use (12) and (13) to compute the average distance of each BOW and CS feature as shown in Figure 4. We can see that the tendencies of all distance curves are similar, and the curve of CS features trends closer to that of BOW features as the subrate increases, which indicates that the pairwise distances between BOW features correspond to those between CS features. To measure the distance differences between BOW and CS features, we compute the Mean Square Error (MSE) between the average distances of BOW and CS features as follows:

$$\text{MSE}_{\text{dist}} = \frac{1}{L_1} \sum_{i=1}^{L_1} \left(\text{dist}_i^{\text{BOW}} - \text{dist}_i^{\text{CS}} \right)^2. \tag{14}$$

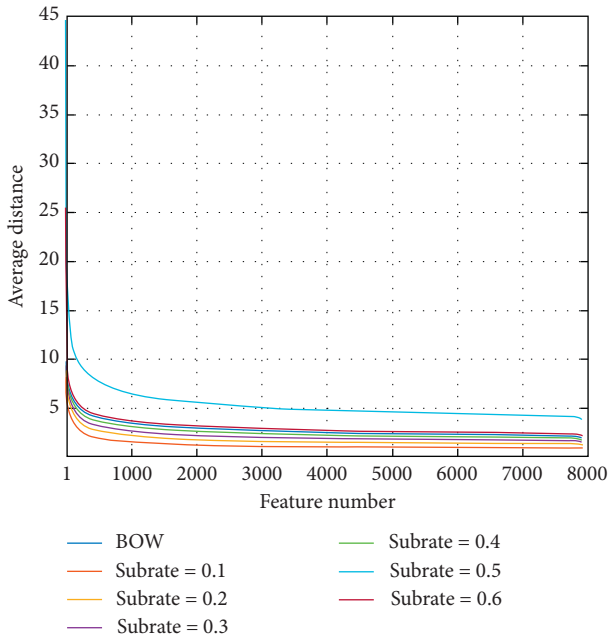


Figure 4: Average distances between any BOW or CS feature and others on multiclass classification dataset at different subrates when using Block DCT matrix. BOW denotes average distances between any BOW feature and others. These average distances are sorted in a descending order.

Table 1 presents the MSEs on multiclass classification dataset when using different subrates and SRMs. It can be seen from Table 1 that all SRMs provide similar MSEs at any subrate; e.g., the average MSE of each SRM at all subrates is about 11.00, and the MSEs of SRMs decrease as the subrate increases; e.g., the MSE of DCT is 18.78 at the subrate of 0.1, and it is reduced to 5.92 at the subrate of 0.6. These MSE results indicate that SRMs can preserve the approximate pairwise distances between BOW features in the CS feature space.

Then, we select SVM as the classifier in our model and evaluate the effects of SRMs on classification accuracy. With different SRMs, the accuracies of SVM classifier on binary and multiclass classification datasets are presented in Table 2. It can be seen that all SRMs provide similar accuracies in most cases at any subrate; e.g., with all subrates considered, the average accuracies of SRMs range from 0.7121 to 0.7203 on binary classification dataset, and similar results are obtained on multiclass classification dataset. We also see that the accuracy is gradually improved for any SRM as the subrate increases. The above results indicate that the selection of SRMs has little impact on classification accuracy, and the subrate is a key factor in controlling the accuracy. Therefore, any SRM can be used in our model, and we need to consider the balance between accuracy and subrate in practice.

4.3. Evaluation on Classifiers. To verify the validity of CS, we have compared CS features and BOW features in terms of the accuracies and training time of different classifiers driven by them. The Block DCT is selected as SRM, and the accuracy results are presented in Table 3. It can be seen that, for binary classification, the accuracies of classifiers driven by the CS features go up with the increase of subrate. Though lower than those with BOW feature when the subrate is small, they quickly catch up; e.g., for SVM, the CS feature overtakes the BOW feature when the subrate is 0.3 and outperforms it thereafter. All the classifiers considered, the average accuracy by the CS features is also comparable with that by BOW feature. The same result can be obtained for multiclass classification. As for the training time in Figure 5, whether it is binary or multiclass classification, the CS feature costs far less than the BOW feature, especially when the subrate is small. Table 4 presents average accuracy, precision, recall, and F_1 on all classifiers for binary classification dataset. It can be seen that the precision, recall, and F_1 by CS features at any subrate are similar to those by BOW features, which indicates that the classification accuracy is reliable for CS features. From the above results, it can be

Table 1: MSEs between average distances of CS and BOW features on multiclass classification dataset when using different subrates and SRMs.

Subrate R	SRMs				
	DCT	FFT	Block DCT	Block WHT	Block Gaussian
0.1	18.78	18.81	18.38	18.48	18.62
0.2	14.37	14.40	14.38	14.08	14.51
0.3	11.39	11.39	11.17	11.11	11.78
0.4	9.14	9.12	9.018	9.01	9.33
0.5	7.36	7.34	7.19	7.21	7.33
0.6	5.92	5.91	5.96	5.90	6.03
Avg.	11.16	11.16	11.02	10.96	11.27

Table 2: Accuracies of SVM classifier associated with different SRMs on binary and multiclass classification datasets at different subrates.

Subrate R	SRMs				
	DCT	FFT	Block DCT	Block WHT	Block Gaussian
Binary classification					
0.1	0.6955	0.7220	0.6975	0.6880	0.6930
0.2	0.7185	0.7135	0.7135	0.7200	0.7055
0.3	0.7195	0.7140	0.7285	0.7215	0.7125
0.4	0.7285	0.7190	0.7265	0.7170	0.7185
0.5	0.7235	0.7195	0.7290	0.7270	0.7145
0.6	0.7255	0.7290	0.7265	0.7280	0.7285
Avg.	0.7185	0.7195	0.7203	0.7169	0.7121
Multiclass classification					
0.1	0.8590	0.8575	0.8358	0.8444	0.8227
0.2	0.8616	0.8606	0.8651	0.8636	0.8585
0.3	0.8651	0.8737	0.8666	0.8737	0.8606
0.4	0.8686	0.8702	0.8712	0.8747	0.8712
0.5	0.8712	0.8732	0.8767	0.8691	0.8757
0.6	0.8747	0.8782	0.8803	0.8732	0.8762
Avg.	0.8668	0.8689	0.8660	0.8665	0.8609

Table 3: Accuracies of different classifiers driven by BOW and CS features on binary and multiclass classification datasets when SRM is Block DCT.

Classifier	BOW feature	Subrate R for CS feature					
		0.1	0.2	0.3	0.4	0.5	0.6
Binary classification							
SVM	0.7220	0.6975	0.7135	0.7285	0.7265	0.7290	0.7265
Decision tree	0.6235	0.6365	0.6395	0.6460	0.6355	0.6465	0.6485
AdaBoost	0.7060	0.7020	0.6975	0.7075	0.7035	0.7020	0.7110
KNN	0.6040	0.5955	0.6120	0.6200	0.6140	0.6145	0.6125
Naïve Bayes	0.7275	0.7035	0.7130	0.7125	0.7170	0.7200	0.7150
Avg.	0.6766	0.6670	0.6751	0.6829	0.6793	0.6824	0.6827
Multiclass classification							
SVM	0.8732	0.8358	0.8651	0.8666	0.8712	0.8767	0.8803
Decision tree	0.8560	0.8454	0.8434	0.8510	0.8520	0.8525	0.8530
AdaBoost	0.7777	0.7535	0.7737	0.7732	0.7813	0.7808	0.7818
KNN	0.8252	0.8080	0.8146	0.8207	0.8242	0.8257	0.8252
Naïve Bayes	0.7737	0.7373	0.7404	0.7464	0.7429	0.7424	0.7454
Avg.	0.8212	0.7960	0.8074	0.8116	0.8143	0.8156	0.8171

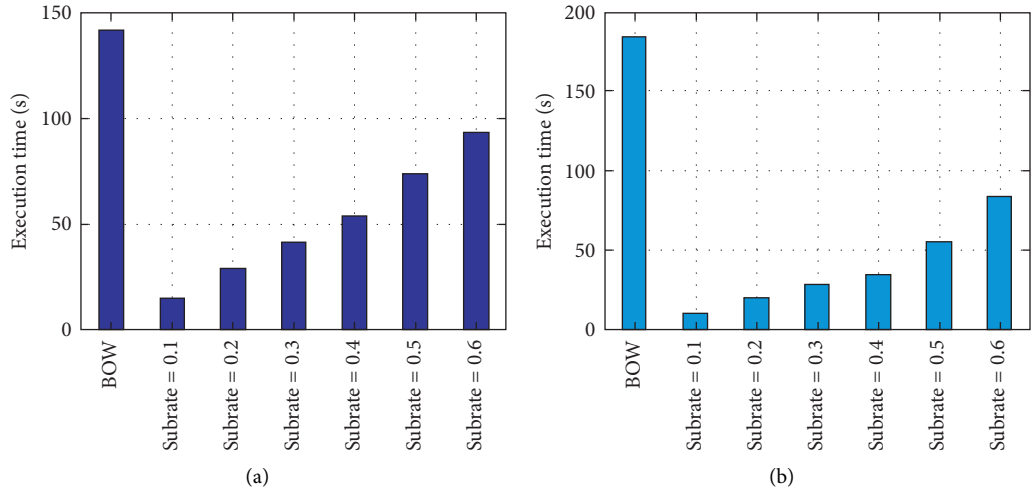


Figure 5: Average training time (s) on all classifiers driven by BOW and CS features for binary and multiclass classification tasks when SRM is Block DCT. (a) Binary classification. (b) Multiclass classification.

Table 4: Average accuracy, precision, recall, and F_1 on all classifiers for binary classification dataset when SRM is Block DCT.

Metrics	BOW feature	Subrate R for CS feature					
		0.1	0.2	0.3	0.4	0.5	0.6
Accuracy	0.6766	0.6670	0.6751	0.6829	0.6793	0.6824	0.6827
Precision	0.6564	0.6658	0.6674	0.6722	0.6694	0.6670	0.6694
Recall	0.6817	0.6671	0.6775	0.6866	0.6824	0.6871	0.6864
F_1	0.6679	0.6664	0.6723	0.6790	0.6756	0.6766	0.6774

concluded that CS speeds up the training of classifiers while providing the accuracies that can match the BOW feature.

4.4. Comparisons on DR Methods. We compare the performance of the proposed CS model with that of some popular DR methods including PCA, ICA, and NMF. PCA learns all principal components from the training set, and, according to the preset subrate, selects part of principal components to construct the transform matrix. ICA and NMF learn their transform matrices at different subrates by numerical iterative algorithms, and their maximum numbers of iterations are both set to be 20 in order to keep the execution time at a moderate level. We use each of the above transform matrices to project all training and testing observations onto a low-dimension space. The proposed CS model uses Block DCT to reduce the dimensionalities of observations at different subrates. Table 5 presents the

Table 5: Average accuracies of all classifiers for binary and multiclass classification datasets when using different DR methods.

DR method	Subrate R					
	0.1	0.2	0.3	0.4	0.5	0.6
Binary classification						
PCA	0.6221	0.6236	0.6206	0.6154	0.6222	0.6091
ICA	0.5754	0.5830	0.5862	0.5974	0.5903	0.6009
NMF	0.5926	0.6127	0.6193	0.6067	0.6157	0.6000
CS	0.6670	0.6751	0.6829	0.6793	0.6824	0.6827
Multiclass classification						
PCA	0.7253	0.7213	0.7019	0.6845	0.6822	0.6726
ICA	0.4938	0.5170	0.5305	0.5448	0.5455	0.5479
NMF	0.7112	0.7080	0.7123	0.7123	0.7096	0.7063
CS	0.7960	0.8074	0.8116	0.8143	0.8156	0.8171

Note that SRM in CS is Block DCT.

Table 6: Execution time (s) of different DR methods on binary and multiclass classification datasets when using different subrates.

DR method	Subrate R					
	0.1	0.2	0.3	0.4	0.5	0.6
Binary classification						
PCA	384.75	384.75	384.75	384.75	384.75	384.75
ICA	369.72	3094.00	17259.27	34511.16	35281.73	50355.25
NMF	187.33	456.67	1169.65	1873.32	2481.44	2201.12
CS	3.32	3.64	3.92	4.19	4.58	4.63
Multiclass classification						
PCA	275.49	275.49	275.49	275.49	275.49	275.49
ICA	188.77	382.82	990.19	6592.11	10829.64	20559.64
NMF	159.21	327.14	652.83	1239.35	1529.07	2358.88
CS	3.10	3.77	3.94	4.03	4.25	4.41

Note that SRM in CS is Block DCT.

average accuracies of all classifiers for binary and multiclass classification datasets when using different DR methods. We can see that the proposed CS model obtains higher accuracies than PCA, ICA, and NMF at any subrate for both binary and multiclass classification tasks. The proposed CS model is more stable, and its accuracy increases gradually as the subrate increases, but the accuracies of PCA, ICA, and NMF float up and down as the subrate increases; for example, for binary classification, the accuracy of PCA is 0.6221 at the subrate of 0.1. However, when the subrate is raised to 0.6, the accuracy drops to 0.6091. Table 6 presents the execution time of different DR methods on binary and multiclass classification datasets when using different subrates. PCA learns all principal components, so its execution time does not vary as the subrate increases, and it costs 387.75 s and 275.49 s for binary classification and multiclass classification, respectively. At the preset subrate, ICA and NMF determine the final dimensionalities of observations and learn the corresponding transform matrices, so their execution time increases as the subrate increases; e.g., for binary classification, NMF costs 187.33 s at the subrate of 0.1 and costs 2201.12 at the subrate of 0.6. The accuracies of ICA and NMF can be improved by increasing iteration times, but their execution time can also increase dramatically. Compared with PCA, ICA, and NMF, the proposed CS model costs less execution time; e.g., for binary classification, CS costs only 3.32 s and 4.63 s at the subrates of 0.1 and 0.6, respectively. From the above results, it can be concluded that the proposed CS model obtains higher accuracy with less execution time when compared with PCA, ICA, and NMF. Therefore, the proposed CS model is a reliable DR method.

5. Conclusion

In this paper, we develop a CS-based model for text classification tasks. Traditionally, the BOW features are extracted from the text dataset, and they are the highly sparse representations with a huge dimensionality. It costs a lot to train classifiers by using BOW features. By using the incoherent measuring of CS, we greatly reduce the dimensionality of BOW features, and, at the same time, the RIP of CS ensures that the pairwise distances between BOW features are well preserved in a low-dimensional CS feature space. CS also provides the SRMs that are fast computable with low memory consumption. In the proposed model, different SRMs are constructed to linearly measure BOW features at a preset subrate, generating the CS features that are used to train the classifiers. Experimental results show that the proposed CS model provides a comparable classification accuracy with the traditional BOW model and

significantly reduces the space and time complexity required by a large-scale dataset training.

Data Availability

The datasets and experimental codes have been downloaded from SIGMULL Team Website: http://www.scholat.com/showTeamScholarEn.html?id=1234&changeTo=En&nav=4.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61572417 and 31872704, in part by Innovation Team Support Plan of Henan University of Science and Technology (no. 19IRTSTHN014), and in part by Nanhu Scholars Program for Young Scholars of Xinyang Normal University.

References

[1] W. Zhu, P. Cui, Z. Wang, and G. Hua, "Multimedia big data computing," *IEEE Multimedia*, vol. 22, no. 3, p. 96, 2015.

[2] G. Song, Y. M. Ye, X. L. Du, X. H. Huang, and S. F. Bie, "Short text classification: a survey," *Journal of Multimedia*, vol. 9, pp. 635–643, 2014.

[3] K. Kowsari, K. J. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: a survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[4] X. L. Deng, Y. Q. Li, J. Weng, and J. L. Zhang, "Feature selection for text classification: a review," *Multimedia Tools and Applications*, vol. 78, pp. 3797–3816, 2019.

[5] L. Qing, W. Linhong, and D. Xuehai, "A novel neural network-based method for medical text classification," *Future Internet*, vol. 11, no. 12, p. 255, 2019.

[6] C. C. Aggarwal and C. X. Zhai, *Mining Text Data*, Springer, Berlin, Germany, 2012.

[7] G. Salton, A. Wong, and C. S. Yang, "A vector-space model for information retrieval," *Communications of the Acm*, vol. 18, pp. 13–620, 1975.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, New York, NY, USA, 2006.

[9] G. Ian, B. Yoshua, and C. Aaron, *Deep Learning*, The MIT Press, New York, NY, USA, 2016.

[10] W. Zhang, T. Yoshida, and X. Tang, "Text classification based on multi-word with support vector machine," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 879–886, 2008.

[11] D. Coppersmith, S. J. Hong, and J. R. M. Hosking, "Partitioning nominal attributes in decision trees," *Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 197–217, 1999.

[12] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[13] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient knn classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2018.

[14] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2/3, pp. 103–130, 1997.

[15] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1–4, pp. 43–52, 2010.

[16] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic dependency-based n-grams as classification features," in *Mexican International Conference on Artificial Intelligence*, pp. 1–11, Springer, Berlin, Germany, 2012.

[17] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[18] V. L. Quoc, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature jearning," *Advances in Neural Information Processing Systems*, vol. 24, pp. 1017–1025, 2011.

[19] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp. 452–456, Lake Buena Vista, FL, USA, April 2004.

[20] C. Huang, L. Zhong, Y. Huang, G. Zhang, and X. Zhong, "A novel method for text recognition in natural scene based on sparse stacked autoencoder," *Journal of Computational Information Systems*, vol. 11, pp. 1399–1406, 2015.

[21] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Proceedings of the 2015 IEEE International Conference on Acoustics Speech & Signal Processing*, pp. 1996–2000, Brisbane, QLD, Australia, April 2015.

[22] W. Xu and Y. Tan, "Semisupervised text classification by variational autoencoder," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 1, pp. 295–308, 2020.

[23] S. Chakrabarti, S. Roy, and M. V. Soundalgekar, "Fast and accurate text classification via multiple linear discriminant projections," *The VLDB Journal The International Journal on Very Large Data Bases*, vol. 12, no. 2, pp. 170–185, 2003.

[24] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: replacing minimization with randomization in learning," *Neural Information Processing Systems*, vol. 21, pp. 1313–1320, 2009.

[25] E. J. Candè and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, 2008.

[26] R. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[27] R. Li, X. Duan, X. Li, W. He, and Y. Li, "An energy-efficient compressive image coding for green internet of things (IoT)," *Sensors*, vol. 18, no. 4, p. 1231, 2018.

[28] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 139–154, 2012.

[29] R. Li, X. Duan, and Y. Li, "Measurement structures of image compressive sensing for green internet of things (IoT)," *Sensors*, vol. 19, p. 102, 2019.

[30] B. Richard, D. Mark, and Devore, "A simple proof of the restricted Isometry property for random matrices," *Constructive Approximation*, vol. 45, pp. 113–127, 2008.

[31] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 120–134, 2010.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8817849, 10 pages
https://doi.org/10.1155/2020/8817849

Research Article

A New Image Classification Approach via Improved MobileNet Models with Local Receptive Field Expansion in Shallow Layers

Wei Wang,[1] **Yiyang Hu**,[1] **Ting Zou**,[2] **Hongmei Liu**,[3] **Jin Wang**,[1,4] **and Xin Wang**[1]

[1]*College of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2]*Yiyang Branch, China Telecom Co., Ltd., Yiyang 413000, China*
[3]*Hunan Railway Professional Technology College, Zhuzhou 410116, China*
[4]*School of Information Science and Engineering, Fujian University of Technology, Fujian 350118, China*

Correspondence should be addressed to Hongmei Liu; lhm4133@126.com and Xin Wang; wangxin@csust.edu.cn

Received 17 June 2020; Revised 4 July 2020; Accepted 10 July 2020; Published 1 August 2020

Academic Editor: Nian Zhang

Because deep neural networks (DNNs) are both memory-intensive and computation-intensive, they are difficult to apply to embedded systems with limited hardware resources. Therefore, DNN models need to be compressed and accelerated. By applying depthwise separable convolutions, MobileNet can decrease the number of parameters and computational complexity with less loss of classification precision. Based on MobileNet, 3 improved MobileNet models with local receptive field expansion in shallow layers, also called Dilated-MobileNet (Dilated Convolution MobileNet) models, are proposed, in which dilated convolutions are introduced into a specific convolutional layer of the MobileNet model. Without increasing the number of parameters, dilated convolutions are used to increase the receptive field of the convolution filters to obtain better classification accuracy. The experiments were performed on the Caltech-101, Caltech-256, and Tubingen animals with attribute datasets, respectively. The results show that Dilated-MobileNets can obtain up to 2% higher classification accuracy than MobileNet.

1. Introduction

Computer image classification is one of the research hotspots in the field of computer vision. It can replace human visual interpretation to some extent by analyzing the image and classifying it into one of several categories. Image classification research mainly focuses on image feature extraction and classification algorithm. The features are very critical to the image classification, but traditional image features such as SIFT [1], HOG [2], and NSCT [3] are usually manually designed. So, the traditional methods are difficult to meet the requirements of the designer. On the contrary, convolutional neural network (CNN) can automatically extract features by using the prior knowledge of known image samples. It can avoid the complex feature extraction process in traditional image classification methods, and the extracted features have strong expression ability and high classification efficiency.

Deep learning technologies [4, 5] have been increasingly applied in image classification [6], target tracking [7], object detection [8], image segmentation [9, 10], and so on, all of which have achieved good results. Russakovsky et al. [11] used AlexNet of approximately 60 million parameters with 5 convolutional layers and 3 fully connected layers to win the 2012 champion of ImageNet Large-scale Visual Recognition Challenge. Then, in order to achieve higher classification accuracy, the deep neural network (DNN) structures have become deeper and more complex. For example, VGG [12] deepened the network to 19 layers, GoogleNet [13] used inception as the basic structure (the network reaches 22 layers), and ResNet [14] introduced residual network structure to solve the gradient vanishing problem. However, the complex DNNs have a large number of parameters and a large amount of computation, which requires a lot of memory access and CPU/GPU resources. Some real-time

applications and low-memory portable devices still cannot fully meet the resource requirements of the DNN models.

To solve the above problems, more and more researches have focused on lightweight networks, which have fewer parameters and less computation while maintaining high accuracy. When analyzing the number of network parameters, Denil et al. [15] found that the parameters in the deep network have a lot of redundancy. In the process of processing, these parameters were useless to improve the classification accuracy but affected the processing efficiency. Hinton et al. [16] significantly improved the compressed model by distilling the models' ensemble knowledge. The classification accuracy of this simple network was almost as same as that of complex network. In terms of network compression, Iandola et al. [17] proposed a small CNN structure called SqueezeNet in 2016, which greatly reduced the number of network parameters. By using depthwise Separable Filters, Howard et al. [18] designed a streamlined architecture called MobileNet, based on depthwise convolution filters and pointwise convolution filters. MobileNet used two global hyperparameters to keep a balance between efficiency and accuracy. As an extremely computation-efficient CNN architecture, ShuffleNet [19] adopted two new operations, pointwise group convolution and channel shuffle. This network can be applied to mobile devices with very limited computing power.

Although the parameters or computation of lightweight network is reduced, the accuracy of classification also decreases correspondingly. Therefore, by introducing the dilated convolution filter into MobileNet, a Dilated-MobileNet approach is proposed based on local receptive field expansion. Without increasing the parameters, the dilated convolution filter can make the network obtain larger local receptive field and improve the classification accuracy.

2. Fundamental Frameworks

2.1. CNN Structure. Convolutional neural network usually consists of convolutional layer, pooling layer, and full connection layer [20], as shown in Figure 1. First, the features are extracted by one or more convolution layers and pooling layers. Then, all the feature maps from the last convolution layer are transformed into one-dimensional vectors for full connection. Finally, the output layer classifies the input images. The network adjusts the weight parameters by back propagation and minimizing the square difference between the classification results and the expected outputs. The neurons in each layer are arranged in three dimensions: width, height, and depth, in which width and height are the size of neurons, and depth refers to the channels number of the input picture or the number of input feature maps.

The convolutional layer, which contains several convolution filters, extracts different features from the image by convolution operation. The convolution filters of the current layer convolute the input feature maps to extract local features and get the output feature maps. Then, the nonlinear feature maps can be obtained by using activation function.

The pooling layer, also known as the subsampling layer, is behind the convolutional layer. It performs downsampling operation, using a specific value as output in a certain subregion. By removing the unimportant sample points from the feature map, the size of input feature map of the subsequent layer is reduced, and the computational complexity is also diminished. At the same time, the adaptability of the network to the changes of image translation and rotation is increased. The most common pooling operations are maximum pooling and average pooling.

The structure based on convolutional layer and pooling layer can improve the robustness of the network model. The convolutional neural network can get deeper through multilayer convolutions. With the number of layers increasing, the features achieved through learning become more global. The global feature map learned at last is transformed into a vector to connect the full connection layer. Most of the parameters in the network model are at the full connection layer.

2.2. MobileNet Structure. MobileNet, as shown in Figure 2, has smaller structure, less computation, and higher precision, which can be used for mobile terminals and embedded devices. Based on depthwise separable convolutions, MobileNets use two global hyperparameters to keep a balance between efficiency and accuracy.

The core idea of MobileNet is the decomposition of convolution kernels. By using depthwise separable convolution, the standard convolution can be decomposed into a depthwise convolution and a pointwise convolution with 1×1 convolution kernel, as shown in Figure 3. The depthwise convolution filters perform convolution to each channel, and the 1×1 convolution is used to combine the outputs of the depthwise convolution layers. In this way, N standard convolution kernels (Figure 3(a)) can be replaced by M depthwise convolution kernels (Figure 3(b)) and N pointwise convolution kernels (Figure 3(c)). A standard convolutional filter combines the inputs into a new set of outputs, while the depthwise separable convolution divides the inputs into two layers, one for filtering and the other for merging.

3. Dilated-MobileNet (Dilated Convolution MobileNet) Structure

MobileNet (Figure 2) mostly uses 3×3 convolution filters. Although this network can reduce the computation cost, the local receptive fields of small convolution filter are too small to capture better features in the case of higher resolution of the feature maps. However, using large convolution filters will increase the number of parameters and the computation load. Therefore, in some first shallow convolutional layers, we use the dilated convolution with the expansion rate of 2 instead of the standard convolution. We call this network Dilated Convolution MobileNet (Dilated-MobileNet).

3.1. Dilated Convolution. Dilated convolution filter [22], which was first applied in image segmentation, is a kind of convolution filter which inserts 0 values between the adjacent

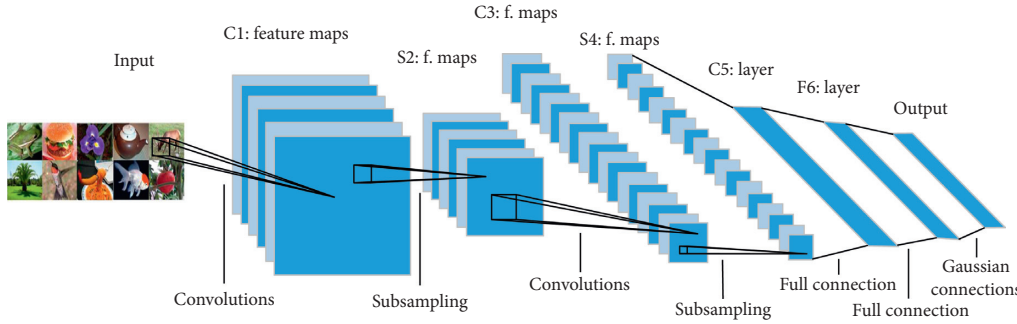


Figure 1: The basic structure of convolution neural network [21].

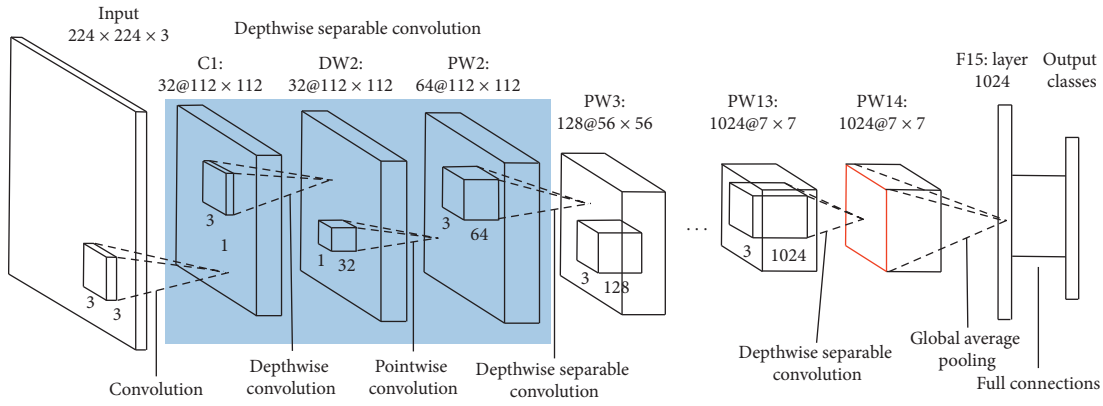


Figure 2: Architecture of MobileNet.

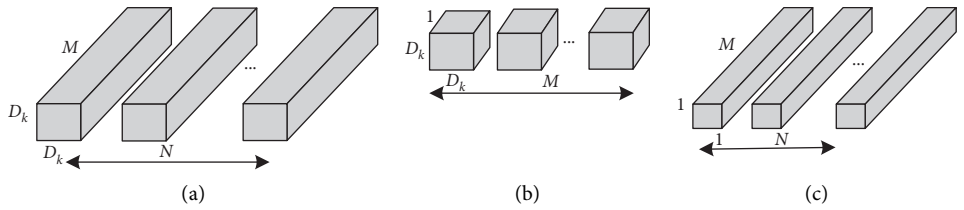


Figure 3: (a) Standard convolution filters, (b) depthwise convolution filters, and (c) pointwise convolution filters.

nonzero values in feature maps. Image segmentation needs the same size image as the original input image, but the pooling layer in traditional DNN will reduce the spatial resolution of the feature map. In order to generate an effective dense feature map and obtain the same size of receptive field, Chen et al. [10] removed the maximum pooling layer in last layers of the full CNN and added dilated convolution. This method not only avoids the reduction of the spatial resolution of the feature map in the pooling layer but also increases receptive field as same as the pooling layer does.

The dilated convolution filter expands the receptive field by inserting 0 values between the nonzero values, as shown in Figure 4. Figure 4(a) represents the receptive field of a 3×3 convolution filter. Figure 4(b) indicates the receptive field, while the 3×3 convolution kernel changed to 5×5 when the expansion rate is 2. Figure 4(c) shows the receptive field, while the 3×3 convolution kernel changed to 7×7 when the expansion rate was 3. Therefore, the dilated convolution can expand the receptive field of convolution filter without increasing the parameters of convolution filter.

3.2. Dilated-MobileNet. Receptive field refers to the size of each element in the feature map of every layer's output mapped on the input image, so the layer will have larger receptive field when closer to the bottom of the network, and its receptive field is approximately equal to the global receptive field. In our research, expanding local receptive field is to improve the classification accuracy of MobileNet, so the layers which need increasing receptive field are near the input of the MobileNet. According to the location of the dilated convolution filter, we propose 3 new network models named D1-MobileNet, D2-MobileNet, and D3-MobileNet.

3.2.1. Dilated1-MobileNet. D1-MobileNet sets convolutional stride as 1 in the first layer and replaces the standard convolution filters with dilated convolution filters with an expansion rate of 2. At the same time, in order to restrain the increase of calculation cost, the stride of the 2nd depthwise separable convolution is set as 2, and the other layers remain unchanged. Compared with MobileNet, the first

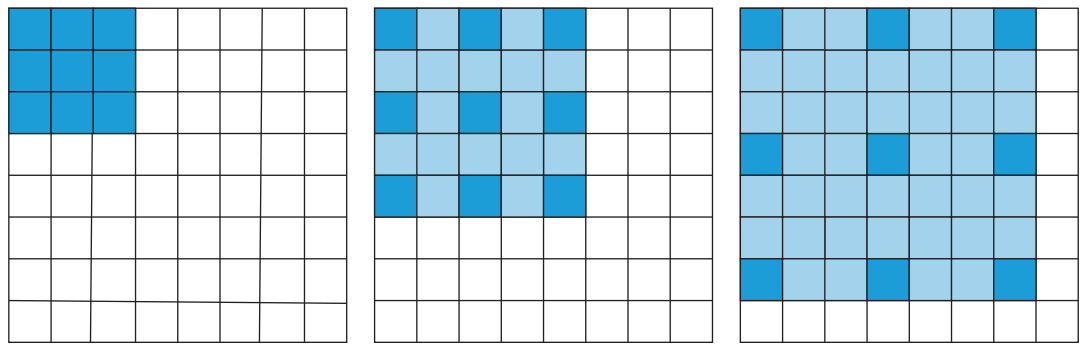


Figure 4: Schematic diagram of dilated convolution kernel.

convolutional layer with stride 1, the size of the output feature map of the first convolutional layer changes from 112×112 to 224×224, as shown in Figure 5.

3.2.2. Dilated2-MobileNet. In DWD2 (depthwise separable) layer, the depthwise convolution filters is expanded by dilated convolution filters with an expansion rate of 2, while the other layers remain unchanged. This approach does not increase the amount of computation and parameters nor does it change the size of the output feature map of any layer, as shown in Figure 6.

3.2.3. Dilated3-MobileNet. D3-MobileNet sets the convolutional stride in first convolutional layer as 1 and replaces the standard convolution filters with dilated convolution filters by using an expansion rate of 2. After the convolution operation in the first convolution layer, it is normalized through batch normalization layer [23]. Then, a maximum pooling layer with a stride of 2 is behind the batch normalization layer, and the other layers are unchanged, as shown in Figure 7.

In terms of receptive field expansion, there are also different ways of expansion. For example, Sun W combined dilated convolution and depthwise separable convolution to form standard blocks for network construction [21]. Their approach is to add a dilated convolution layer before each depthwise separable convolution. Unlike their approach, in the Dilated1-MobileNet, we use dilated convolution instead of the standard convolution in the first layer of MobileNet, without adding dilated convolution in front of all subsequent depthwise separable convolution blocks because that would increase the number of parameters. The difference in Dilated2-MobileNet is greater because we extend the receptive field in depthwise convolution layer rather than adding a dilated convolution layer in front of the depthwise separable convolution layer. Similarly, Dilated3-MobileNet replaces standard convolution with a dilated convolution at the first level and add a pooling layer after it, rather than adding a dilated convolution in front of all depthwise separable convolution blocks.

3.3. Computation Analysis. In the standard convolutional layer, assuming the height, width, and input channel number of the input feature maps I are h, w, and m, the convolution filter K is $s \times s$, the output channel number is n, and the output feature maps $O = K \times I$ can be obtained by the convolution of I and K with no padding zeros and stride 1, as shown in the following formula:

$$O(y, x, j) = \sum_{i=1}^{m} \sum_{u,v=1}^{s} K(u, v, i, j) I(y + u - 1, x + v - 1, i), \tag{1}$$

where $O(y, x, j)$ represents the value of point (y, x) in jth output feature map, $K(u, v, i, j)$ represents the value of point (u, v) on channel i in jth convolution filter, and $I(y, x, i)$ represents the value of point (y, x) on ith input feature map. From Formula (1), it is known that an output value needs $s \times s \times m$ times multiplication, so the total amount of calculations is $s \times s \times m \times (h - s + 1) \times (w - s + 1) \times n$ and the number of parameters is $s \times s \times m \times n$.

When Dilated-MobileNet introduces the dilated convolution in the standard convolution layer, with feature map I, the dilated convolution is performed with no padding zeros by using convolution kernel K of the same size and expansion rate of 2. So, we can get the output feature map O_d by the following formula:

$$O_d(y, x, j) = \sum_{i=1}^{m} \sum_{u,v=1}^{s} K(u, v, i, j) I(y + u + (u - 1)(r - 1) - 1, x + v + (v - 1)(r - 1) - 1, i). \tag{2}$$

So, the total computational amount of the dilated convolution layer is $(s \times s \times m) \times (h - s - (s - 1)(r - 1) + 1) \times (w - s - (s - 1)(r - 1) + 1) \times n$, and the number of parameters is $s \times s \times m \times n$. With no padding zeros, the computation of dilated convolution with expansion rate $r > 1$ is less than that of standard convolution, and the number of parameters is the same, but the receptive field of dilated convolution is larger than that of standard convolution. Under the convolution operation with padding zeros, the map size of the dilated convolution is the same as that of the

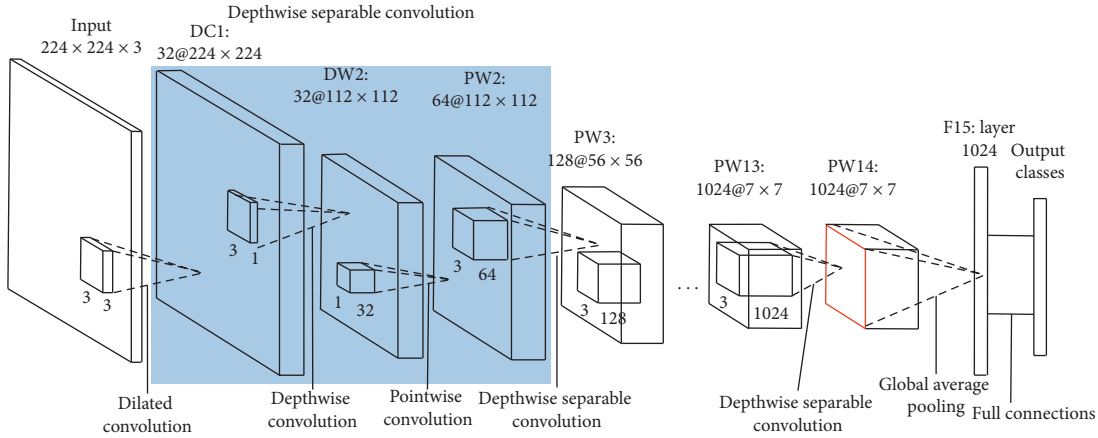


Figure 5: Architecture of Dilated1-MobileNet.

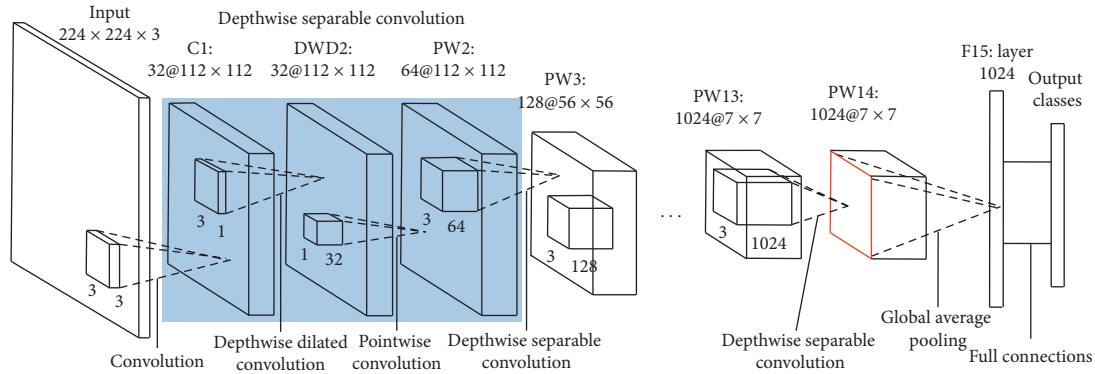


Figure 6: Architecture of Dilated2-MobileNet.

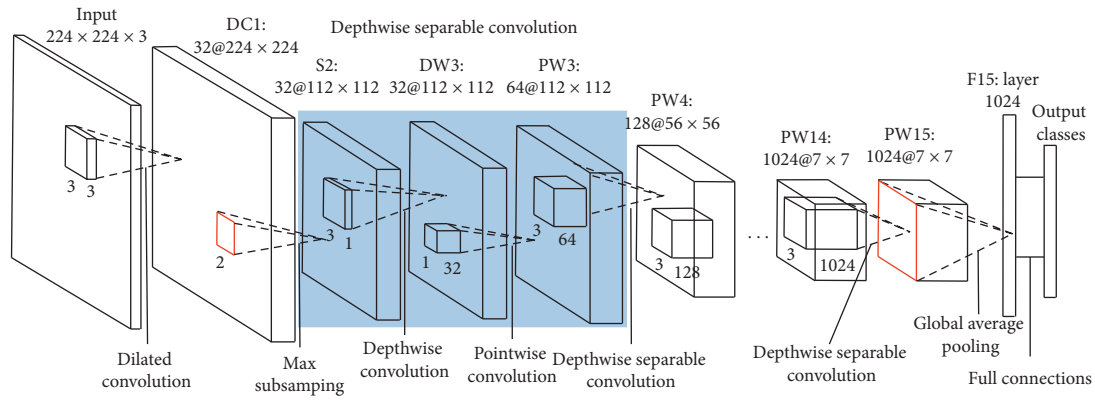


Figure 7: Architecture of Dilated3-MobileNet.

standard convolution, both of which are $h \times w \times n$, and the computation and the number of parameters are the same too.

When introducing dilated convolution filters to the depthwise convolution, the above feature maps I is firstly convoluted with the depthwise convolution filter K, and the output feature graph O_{dc} is obtained through the following formula:

$$O_{dc}(y, x, j) = \sum_{u,v=1}^{s} K(u, v, j) I(y + u + (u-1)(r-1) - 1, x + v + (v-1)(r-1) - 1, j), \tag{3}$$

where $O_{dc}(y, x, j)$ represents the value of point (y, x) in jth feature map. Since the depthwise convolution filter has only one channel, $K(u, v, j)$ represents the value of point (u, v) on

jth convolution filter and $I(y, x, j)$ represents the value of point (y, x) on jth input channel.

The total computation of the depthwise separable convolution is $(s \times s \times n) \times (h - s - (s-1)(r-1) + 1) \times (w - s - (s-1)(r-1) + 1) \times m$, and the total number of parameters is $s \times s \times m + m \times n$. It can be seen that the parameter of the depthwise separable convolution are reduced compared with the standard convolution:

$$\frac{s \times s \times m + m \times n}{s \times s \times m \times n} = \frac{1}{n} + \frac{1}{s^2}. \tag{4}$$

The ratio of computation is

$$\frac{(s \times s + n) \times (h - s - (s-1)(r-1) + 1) \times (w - s - (s-1)(r-1) + 1) \times m}{s \times s \times m \times n \times (h - s + 1) \times (w - s + 1)} = \frac{1}{n} + \frac{1}{s^2}. \tag{5}$$

Similarly, when carrying out the depthwise convolution with padding zeros, the reduction ratio of parameters is

$$\frac{(s \times s + n) \times m \times h \times w}{s \times s \times m \times n \times h \times w} = \frac{1}{n} + \frac{1}{s^2}. \tag{6}$$

From the above analysis, it can be seen that the receptive field of the deep convolution kernel with expansion rate r and convolution kernel size $s \times s$ is equivalent to that of the convolution kernel $(r \times s - r + 1) \times (w \times s - r + 1)$, thus can expand the receptive field without increasing the number of parameters and calculation amount.

3.4. Receptive Field. In many tasks, especially intensive prediction tasks such as semantic image segmentation and optical flow estimation, it is necessary to predict each pixel's value of the input image, and each output pixel's value needs a large receptive field to retain important information. Local receptive field refers to the size of the region in the input feature map of the upper layer, and the region is mapped by the pixel in the output feature map. In this paper, dilated convolution is used to enlarge the local receptive field of a certain layer to capture better features and further influence the receptive field size of the convoluted layer behind. The size of receptive field of each layer is shown in in the following formula:

$$r_k = \begin{cases} f_k, & k = 1, \\ r_{k-1} + \left((f_k - 1) \times \prod_{i=1}^{k-1} s_i \right), & k > 1, \end{cases} \tag{7}$$

where r_k denotes the receptive field size of the kth layer, f_k denotes the size of filter, and s_i denotes the stride of the ith layer. The receptive field of the first layer equals to the size of the filter. By using Formula (7), we can get the receptive field size of each layer of MobileNet and Dilated-MobileNet, as shown in Table 1.

The "ds" in Table 1 shows the depthwise separable convolution, and the pointwise convolution has the same receptive field as the depthwise convolution in depthwise separable convolution, so the receptive field is given uniformly. The receptive field sizes of the first convolution layers in D1-MobileNet and Dilated3-MobileNet show that the receptive field of the 3×3 convolution kernel changed to 5×5 when the expansion rate is 2. In summary, dilated convolution is able to enlarge the size of local receptive field. Moreover, Dilated1-MobileNet and Dilated2-MobileNet also slightly increase the receptive field size of the underlying layers. It can be seen from Table 1 that, for Dilated-MobileNet networks, although the expansion ratio of the receptive fields of the latter convolution layers becomes smaller, their receptive fields of the first few layers are larger than those of MobileNet. In this way, it is easier to extract more detailed information, which is conducive to the improvement of classification accuracy.

4. Experiments and Result Analysis

In the experiments, we compare the classification results of 6 networks: SqueezeNet [17], MobileNet [18], Dense1-MobileNet [24], Dense2-MobileNet [24], D1-MobileNet, D2-MobileNet, and D3-MobileNet on Caltech-101 [25] and Catech-256 [26] datasets and Tubingen Animals with Attributes [27].

The Caltech-101 dataset is an image object recognition dataset, which consists of a total of 9146 images, split between 101 different object classes and an additional background/clutter class. Each object class contains between 40 and 800 images on average. After labeling the pictures in the dataset, 1500 pictures are randomly selected as the test pictures and the rest as the training pictures. Some samples are shown in Figure 8.

The Caltech-256 dataset is based on the Caltech-101 dataset, adding image classes and the number of images in each class. The dataset contains 30607 images in 257 classes, including 256 object classes and one background class. Each class has at least 80 pictures and a maximum of 827 in background class. Figure 9 shows the image examples in the Caltech-256 dataset. Each picture in the dataset is labeled and shuffled. 3060 pictures are randomly selected as test images, and the remaining pictures are used as training images.

We also verify our method on the Animals with Attributes (AwA) dataset, as shown in Figure 10. There are a total of 50 animal classes in the database with a total of 30475 pictures. In experiments, we select 21 animal categories, which are the largest classes and have almost the same number of pictures, as the experimental dataset. There are 22742 pictures in these 21 animal classes, and the number of pictures in each class is between 850 and 1600. After labeling the pictures in the dataset, 2000 pictures are randomly selected as the test pictures and the rest as the training pictures.

Table 1: The receptive field size of each layer.

	MobileNet	Dilated1-MobileNet	Dilated2-MobileNet	Dilated3-MobileNet
Conv1	3	5	3	5
Pool	—	6	—	—
Conv2 ds	7	10	11	7
Conv3 ds	11	14	15	11
Conv4 ds	19	22	23	19
Conv5 ds	27	30	31	27
Conv6 ds	43	46	47	43
Conv7 ds	59	62	63	59
Conv8 ds	91	94	95	91
Conv9 ds	123	126	127	123
Conv10 ds	155	158	159	155
Conv11 ds	187	190	191	187
Conv12 ds	219	222	223	219
Conv13 ds	251	254	255	251
Conv14 ds	315	318	319	315

Figure 8: Picture instances in the Caltech-101 dataset.

Figure 9: Picture instances in the Caltech-256 dataset.

The experiments are under TensorFlow framework and the programming language is Python. The experimental server is equipped with an NVIDIA TITAN GPU. RMSprop optimization algorithm is used in the experiments. RMSprop is an adaptive learning rate method, which can adjust the learning rate. In the experiments, the initial learning rate is 0.1. Since the Xavier initialization method can determine the random initialization distribution range of parameters according to the number of inputs and outputs of each layer, we use it to initialize the weight coefficients. ReLU is used as the activation function in the experiments, and a total of 50,000 batches are trained, with 64 samples per batch.

In the following experiments, all the results are the averages of 10 times experiments, and the best classification accuracy rates are in bold in the tables. Table 2 shows the classification accuracies of 7 network models on the Caltech-101 dataset.

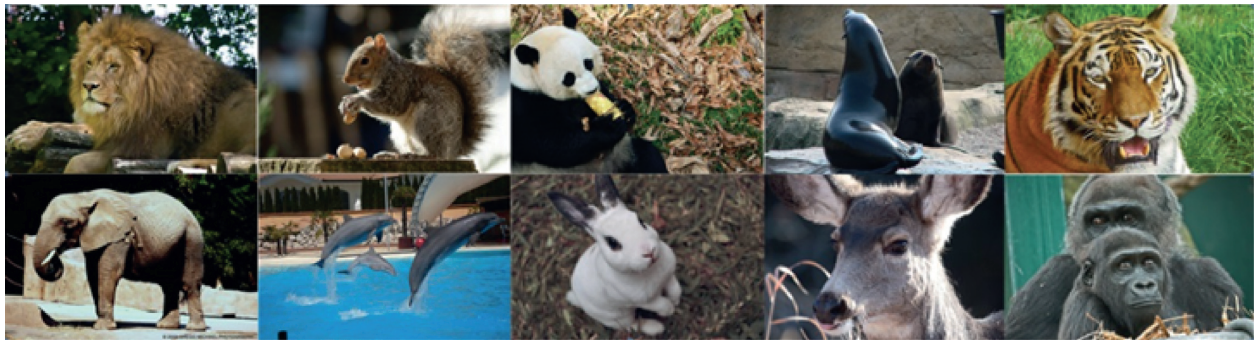


Figure 10: Picture instances in Tuebingen Animals (21) dataset.

Table 2: Classification accuracy rates (%) on Caltech-101 dataset.

Number of iterations	30000	35000	40000	45000	50000
SqueezeNet	53.60	53.60	53.47	53.40	53.47
MobileNets	76.73	76.60	76.60	76.80	76.60
Dense1-MobileNet	76.60	76.53	76.47	76.40	76.47
Dense2-MobileNet	77.60	77.67	77.87	77.80	77.80
Dilated1-MobileNet	77.40	77.47	77.53	77.40	77.47
Dilated2-MobileNet	77.67	77.80	77.73	77.67	77.73
Dilated3-MobileNet	78.60	78.60	78.53	78.53	78.73

Table 3: Classification accuracy rates (%) on Caltech-256 dataset.

Number of iterations	30000	35000	40000	45000	50000
SqueezeNet	41.48	43.06	43.39	43.58	44.03
MobileNets	64.48	64.58	64.55	64.67	64.52
Dense1-MobileNet	64.61	64.53	64.45	64.44	64.47
Dense2-MobileNet	65.62	65.67	65.84	65.78	65.79
Dilated1-MobileNet	65.77	65.74	65.87	65.90	65.87
Dilated2-MobileNet	66.10	66.06	65.94	65.84	65.94
Dilated3-MobileNet	64.97	64.9	64.87	65.19	65.16

We also validate our method on the Animals with Attributes (AwA) dataset [28]. The classification accuracy rates are shown in Table 4.

As seen from Table 2, the accuracy rates of the 7 network models have reached a balance after 30000 iterations, and the accuracy rates of our 3 improved Dilated-MobileNets models are about 0.8%~2% higher than those of the MobileNet model. Among of them, the classification accuracy rate of Dilated1-MobileNet model is improved by 0.87% and that of Dilated2-MobileNet model is improved by 1.13%. The Dilated3-MobileNet model has the best effect, the accuracy rate is increased by 2.13%, and the final classification accuracy rate is 78.73%.

Table 3 is a comparison of the classification accuracy rates of the 7 network models on the Caltech-256 dataset. As shown in Table 3, the accuracy rates of the 7 network models also have reached a balance after 30000 iterations, and the accuracy rates of our 3 improved models are improved by 0.5%~1.5% than that of MobileNet model. Among of them, the accuracy rate of Dilated1-MobileNet model is improved by 1.35%, the accuracy rate of Dilated3-MobileNet model is improved by 0.64% and that of Dilated2-MobileNet model is the highest, which is improved by 1.42% and final reaches to 65.94%.

It can be seen from Table 4 that the accuracy rates of MobileNets and Dilated-MobileNet models have reached a balance after 30000 iterations, but the accuracy rate of SqueezeNet still increases and finally reaches a balance at the accuracy rate of 73.85% after 50000 iterations. As in the previous 2 experiments, the accuracy rates of MobileNet, Dense-MobileNets, and our 3 improved models are much higher than those of SqueezeNet. The accuracy rates of the 3 improved Dilated-MobileNet models are about 0.5%~1.2% higher than those of MobileNet. Among them, the classification accuracy rate of Dilated1-MobileNet model is finally improved by 0.8%, the classification accuracy rate of Dilated2-MobileNet is finally improved by 0.4%, and the

Table 4: Classification accuracy rates (%) on AwA (21) dataset.

Number of iterations	30000	35000	40000	45000	50000
SqueezeNet	72.65	72.10	73.30	73.40	73.85
MobileNets	91.60	91.60	91.60	91.55	91.60
Dense1-MobileNet	90.65	90.60	90.60	90.60	90.65
Dense2-MobileNet	92.10	92.05	92.10	92.05	92.05
Dilated1-MobileNet	92.45	92.45	92.50	92.35	92.40
Dilated2-MobileNet	92.00	92.05	92.05	92.00	92.00
Dilated3-MobileNet	92.85	92.75	92.80	92.70	92.80

classification accuracy rate of Dilated3-MobileNet is the highest, reaching 92.8%.

In the above 3 kinds of experiments, the Dense1-MobileNet and Dense1-MobileNet based on dense connection also achieved good classification effect. The results of the experiments on caltech-256 dataset are slightly better than those of Dilated3-MobileNet and a little worse than those of Dilated1-MobileNet and Dilated2-MobileNet. The design idea of Dense-MobileNets is different from that of the Dilated-MobileNets, and the network structures are also different, so the two approaches can be used together in the practical application.3

5. Conclusions

The memory-intensive and highly computation-intensive properties of deep learning approaches restrict their applications in portable devices. At the same time, the compression and acceleration of network models will reduce the classification accuracy. So, this paper uses the dilated convolution in the lightweight neural network (MobileNet) to improve the classification accuracy without increasing the network parameters and proposes three Dilated-MobileNet models. The experimental results show that Dilated-MobileNets have better classification accuracies on Caltech-101, Catech-256, and AWA datasets.

In recent years, new lightweight networks, such as mobilenetv2 [29] and mobilenetv3 [28], have emerged. How to reduce the parameters and improve the classification effect is still one of the research hotspots. Meanwhile, some deep learning methods combined with traditional methods have achieved good results in target recognition and classification [30]. On the other hand, designing specific deep learning networks based on the characteristics of classification targets is a very effective classification approach [31, 32]. Therefore, how to give full use of the advantages of different methods is also worth further studying.

Data Availability

All datasets in this article are public datasets and can be found on public websites.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

W.W. and H.L. were involved in the conceptualization; T.Z. and Y.H. were responsible for the methodology; T.Z. and X.W. were responsible for the software; J.W. performed the formal analysis; H.L. investigated the study; W.W. and X.W. wrote and prepared the original draft.

Acknowledgments

We would like to thank the National Defense Pre-Research Foundation of China (7301506); the National Natural Science Foundation of China(61070040); the Education Department of Hunan Province (17C0043); and the Hunan Provincial Natural Science Fund (2019JJ80105) for their support.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[3] W. Xin, T. Can, W. Wei, and L. Ji, "Change detection of water resources via remote sensing: an L-V-NSCT approach," *Applied Sciences*, vol. 9, no. 6, p. 1223, 2019.

[4] W. Wang, Y. Yang, and X. Wang, "Development of convolutional neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, p. 1, Article ID 040901, 2019.

[5] W. Wang, Y. Hu, Y. Luo, and Y. Zhang, "Brief survey of single image super-resolution reconstruction based on deep learning approaches," *Sensing and Imaging*, vol. 21, no. 1, 2020.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, Nevada, December 2012.

[7] N. Wang and D. Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, pp. 809–817, Lake Tahoe, Nevada, December 2013.

[8] J. Wan, D. Wang, S. C. H. Hoi et al., "Deep learning for content-based image retrieval: a comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, Orlando, FL, USA, pp. 157–166, November 2014.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, IEEE, Boston, MA, USA, June 2015.

[10] L. C. Chen, G. Papandreou, and I. Kokkinos, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[11] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

[13] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, IEEE, Boston, MA, USA, June 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, IEEE, Las Vegas, NV, USA, June 2016.

[15] M. Denil, B. Shakibi, and L. Dinh, "Predicting parameters in deep learning," in *Advances in Neural Information Processing Systems*, pp. 2148–2156, Lake Tahoe, Nevada, December 2013.
[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, https://arxiv.org/abs/1503.02531.
[17] F. N. Iandola, S. Han, and M. W. Moskewicz, "SqueezeNet: AlexNet-level accuracy with 50 x fewer parameters and< 0.5 MB model size," 2016, https://arxiv.org/abs/1602.07360.
[18] A. G. Howard, M. Zhu, and B. Chen, "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.
[19] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, IEEE, Salt Lake City, UT, USA, June 2018.
[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[21] W. Sun, X. Zhou, X. Zhang, and X. He, "A lightweight neural network combining dilated convolution and depthwise separable convolution," in *Cloud Computing, Smart Grid and Innovative Frontiers in Telecommunications. CloudComp 2019, SmartGift 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, X. Zhang, G. Liu, M. Qiu, W. Xiang, and T. Huang, Eds., Vol. vol 322, Springer, Cham, Switzerland, 2020.
[22] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, https://arxiv.org/abs/1511.07122.
[23] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.
[24] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-MobileNet models," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.
[25] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.
[26] G. Griffin, A. Holub, and P. Perona, "The Caltech-256 Object Category Dataset," Tech. Rep. CNS-TR-2007-001, Caltech, Pasadena, Calif, USA, 2007.
[27] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, IEEE, Miami, FL, USA, June 2009.
[28] A. Howard, M. Sandler, and G. Chu, "Searching for mobilenetV3," 2019, https://arxiv.org/abs/1905.02244.
[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, pp. 4510–4520, June 2018.
[30] W. Wang, C. Tang, X. Wang, L. Yanhong, H. Yongle, and L. Ji, "Image object recognition via deep feature-based adaptive joint sparse representation," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8258275, 9 pages, 2019.
[31] W. Wang, C. Zhang, J. Tian, J. Qu, and J. Li, "A SAR image targets recognition approach via novel SSF-net models," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8859172, 9 pages, 2020.
[32] W. Wang, C. Zhang, and J. Tian, "High resolution radar targets recognition via inception-based VGG (IVGG) networks," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8893419, 11 pages, 2020.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8891778, 15 pages
https://doi.org/10.1155/2020/8891778

Research Article

Density Peaks Clustering by Zero-Pointed Samples of Regional Group Borders

Lin Ding,[1] **Weihong Xu**,[1,2] **and Yuantao Chen**[1]

[1]*School of Computer and Communication Engineering and Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, Hunan 410114, China*
[2]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China*

Correspondence should be addressed to Yuantao Chen; chenyt@csust.edu.cn

Received 17 May 2020; Revised 1 June 2020; Accepted 6 June 2020; Published 18 July 2020

Academic Editor: Nian Zhang

Density peaks clustering algorithm (DPC) has attracted the attention of many scholars because of its multiple advantages, including efficiently determining cluster centers, a lower number of parameters, no iterations, and no border noise. However, DPC does not provide a reliable and specific selection method of threshold (cutoff distance) and an automatic selection strategy of cluster centers. In this paper, we propose density peaks clustering by zero-pointed samples (DPC-ZPSs) of regional group borders. DPC-ZPS finds the subclusters and the cluster borders by zero-pointed samples (ZPSs). And then, subclusters are merged into individuals by comparing the density of edge samples. By iteration of the merger, the suitable dc and cluster centers are ensured. Finally, we compared state-of-the-art methods with our proposal in public datasets. Experiments show that our algorithm automatically determines cutoff distance and centers accurately.

1. Introduction

Clustering algorithm [1], as the unsupervised learning method, divides the objectives that also are called elements, samples, and items, into several groups according to the similarity of objectives. Compared with supervised learning [2–16], it can carry out the grouping task even though the category labels are pending. Hence, it is widely used in image segmentation [17], bioinformatics [18], pattern recognition [19], data mining [20], and other fields [21, 22]. Representative clustering algorithms cover K-means [23, 24] and fuzzy c-means [25, 26] based on partitioning; AGNES [27], BIRCH [28, 29], and CURE [30, 31] based on hierarchy; DBSCAN [32] and OPTICS [33] based on density; STING [34] based on grids; and statistical clustering CMM [35] and spectral clustering [36] based on graph theory [37]. K-means is extremely sensitive to noise and the selection of the initial clustering centers, and the number of clusters needs to be set a priori. Similarly, fuzzy c-means suffers from initial partition dependence, noise, and outliers. The hierarchical clustering requires to determine the number of clusters a priori, and its effect depends on the choice of distance measurement of groups. Density-based DBSCAN, OPTICS, and grid-based clustering algorithms determine the number of clusters without artificial intervention. Still, all require preset parameters epsilon and minpts, and a mass of argument adjustments were taken to obtain optimal clustering results. These two types of algorithms generate noises around the cluster boundaries. Statistics-based CMM needs to select one or more suitable probability models to fit a dataset.

Clustering by fast search and find of density peaks [38] was published in Science, by the preset threshold (cutoff distance, dc), manually selecting the cluster centers from the decision graph proposed by DPC. Compared with traditional clustering algorithms, it has many advantages, such as higher efficiency in finding cluster centers, fewer parameters, no iteration, no noise around the cluster border, and others. However, the algorithm still has the following defects:

(1) The original DPC does not provide a reliable and specific selection method of dc. Hence, the cutoff distance is computed in different ways depending on the size of datasets, in which the inappropriate dc leads to performance degradation [39]. Moreover, the dc is generally challenging to determine since the range of each attribute is unknown in most cases [40].

(2) It is hard to manually select the cluster centers from a dataset with a large number of clusters. And the artificial option for cluster centers cannot meet the system with high timeliness.

To overcome the above defects, many scholars proposed improvements in the original DPC algorithm. Xie et al. proposed a local density metric based on fuzzy weighted k-nearest neighbors to solve the problem of difficult to determine dc in the DPC algorithm [39]. Liu et al. proposed shared-nearest-neighbor-based clustering by fast search and find of density peaks clustering (SNN-DPC), which converts cutoff distance to the number of nearest neighbors [40]. Mehmood presented a nonparametric method for DPC via heat diffusion for estimating the probability distribution of a given dataset [41]. Guo et al. used linear regression to fit the decision values with a given dc and selected the elements above the fitting function as the central elements [42]. Ding et al. proposed an algorithm based on the generalized extreme value distribution (GEV) to fit the decision values in descending order [43]. In order to reduce the time complexity, an alternative method based on density peaks detection using Chebyshev inequality (DPC-CI) was also given. Ni et al. presented the concepts of density path and density gap, as well as a new threshold called dc percentage in [44]. The density gaps are used to draw the summary graph of density gaps calculated by several dc percentages. Instead of the decision graph, the appropriate threshold value is determined by manually observing the summary graph. The algorithm is able to reduce the negative impact of inappropriate dc on the clustering result.

However, in [39–41, 44–47], it is necessary to select the centers or observe the summary graph of density gaps, with the human operation. Gu et al. [42] and Ding et al. [43] proposed the strategies of automatic center selection for the original DPC, but they depend on the given appropriate dc. However, Xie et al. [39] and Liu et al. [40] showed that it was challenging to select the proper dc.

In this paper, we propose the density peaks clustering by zero-pointed samples (DPC-ZPSs) of regional group borders. Our method not only determines the suitable range of dc and the center of each cluster but also reduces the negative impact caused by manual participation in the clustering process. The main innovations and contributions in our algorithm are as follows:

(1) To merge the local clusters into individuals, we present a cluster merging strategy based on comparing density among elements of two cluster borders.

(2) In order to find the border of each cluster, we propose two conceptions: neighboring cluster border (NCB) and pure cluster border (PCB).

(3) For the determination of the correct number of clusters, we provide an iterative procedure, which can converge dc to a suitable value.

The remainder of this paper comprises four sections: Section 2 describes the details of the original DPC and our proposal; Section 3 presents the clustering results on our method and related works and discusses the impact and value range of the parameter of DPC-ZPS; in the final section, we have a summary of the contributions and features of this paper and put forward to future work.

2. Materials and Methods

2.1. The Original DPC Algorithm. For a given dataset $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$, $i = 1, 2, \ldots, n$.

DPC is based on an assumption where each cluster center has a higher local density than other elements and is far from each other. Centers are manually selected using a decision graph with the local density as the abscissa and δ_i as the ordinate. DPC algorithm provides two methods for calculating the local density for each element of the given dataset and is expressed in equations (1) and (2). δ_i is calculated by equation (3):

$$\rho_i = \sum_j \aleph\left(d_{ij} - \mathrm{dc}\right), \quad \aleph(\cdot) = \begin{cases} 1, & \cdot < 0, \\ 0, & \cdot \geq 0, \end{cases} \tag{1}$$

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{\mathrm{dc}}\right)^2\right), \tag{2}$$

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j}\left(d_{ij}\right), & if\ \exists j\ s.t.\ \rho_i < \rho_j, \\ \max\limits_{j}\left(d_{ij}\right), & \text{otherwise}, \end{cases} \tag{3}$$

where d_{ij} is the Euclidean distance between elements i and j and dc is the cutoff distance. As shown in equation (3), δ_i is the minimum distance between elements i and j whose density is higher than i. Moreover, for i with the highest density, its δ_i is the maximum distance between i and j.

Meanwhile, to simplify the selection of centers, DPC provides the decision value γ_i as follows:

$$\gamma_i = \rho_i \times \delta_i. \tag{4}$$

After the cluster centers are determined, each of the remaining samples is assigned to the nearest denser one. And the assignment is recorded in the process of calculating δ_i.

2.2. Our Method. The main process of DPC-ZPS is to select multiple distances as dc at equal intervals and calculate the corresponding decision values. Then, among the decision values of each group, the elements greater than the sum of the mean and standard deviation of the decision values are

selected as the potential centers. In the range of multiple groups of dc, the iterative merging process makes the number of clusters close to the real value gradually.

2.2.1. Related Concepts

Definition 1 (zero-pointed sample). in the assignment, each sample is assigned to the nearest denser one. And the zero-pointed sample (ZPS) is the one without any subordinates.

When dc is fixed, we use an array that consists of n zero units to store the assignment process. And the indexes of the array represent the sequence number of objectives. Let $\text{array}(i) = j$, in which sample j is the nearest and has density more significant than sample i. And cluster centers and potential cluster centers are not assigned. Subsequently, the array is broken at the zero units; then, $|C|$ trees can be obtained, and each tree is a cluster.

Definition 2 (initial border). in a cluster tree, the initial border (IB) consists of all leaf nodes and their father nodes.

As shown in Figure 1, elements 1, 7, and 8 are zero-pointed and leaf nodes because they are less dense than neighboring elements. Elements 3 and 32 are inner, but they are still the zero-pointed elements since they have no adjacent samples. And there are assignment paths of items $10 \longrightarrow 11 \longrightarrow 13$ and $12 \longrightarrow 11 \longrightarrow 13$.

Definition 3 (neighboring cluster border). clusters in a dataset X are denoted as $C = \{C_v \mid v = 1, 2, \ldots, |C|\}$, where $|C|$ is the number of clusters in C and $C_v = \{c_{vl} \mid l = 1, 2, \ldots, |C_v|\}\ \forall c_{vl}, c_{v'l'}$, where $v \neq v'$, $l\prime = \{1, 2, \ldots, |C_{v'}|\}$, satisfies the following equation, and then $c_{vl}, c_{v'l'} \in \text{NCB}(C_v, C_{v'})$:

$$d(c_{vl}, c_{v'l'}) < \overline{\overline{\text{dc}}}\left[\text{floor}\left(\text{DF} \cdot \frac{n_{vv'} \cdot (n_{vv'} - 1)}{2}\right)\right], \tag{5}$$

$$n_{vv'} = |C_v| + |C_{v'}|, \tag{6}$$

where $d(c_{vl}, c_{v'l'})$ is the distance between c_{vl} and $c_{v'l'}$, $\overline{\overline{\text{dc}}}$ is an array storing all $\overline{\overline{d}}(c_{vl}, c_{v'l'})$ of cluster pair C_v and $C_{v'}$ in descending order, $\overline{\overline{\text{dc}}}[a]$ represents the a^{th} distance, DF is the depth factor of the neighboring cluster-border, its range is $(0, 1]$, and $\text{floor}(b)$ is the integer part of b.

Neighboring cluster border (NCB) consists of all $\text{NCB}(C_v, C_{v'})$, and it is expressed as follows, where $v < v'$ is to delete the symmetrical cluster pairs:

$$\text{NCB} = \% \cup_{v<v'} \text{NCB}(C_v, C_{v'}). \tag{7}$$

It is necessary that two clusters are far from each other with an enormous DF to attain a nonblank NCB. And the bigger the required DF value of the nonblank NCB is, the further distance the two clusters are. While for neighboring subclusters, DF is relatively minute. In the fourth chapter, the DF will be compared with parameters of DPC and is discussed to show the impact on the clustering result.

As shown in Figure 1, there are two clusters A and B in a dataset, and cluster B is misclassified into B1, B2, and B3. The elements I, 7 and 8, and II, 16, 17, 18, 19, 20, and 21, are marked with red wireframes. They belong to NCB.

Definition 4 (pure cluster border). in a cluster, the pure cluster border (PCB) is defined by the following equation:

$$\begin{aligned} \text{PCB} &= \text{initial border} - (\text{initial border} \cap \text{NCB}), \\ \text{PCB}_v &= \text{PCB} \cap C_v. \end{aligned} \tag{8}$$

Correspondingly, elements 1, 2, 4, 5, 6, 9, 10, 11, 12, 22, 23, 24, 29, 30, and 31 belong to pure cluster border (PCB) of respective clusters. However, as shown in Figure 2, elements 3 and 32 are zero-pointed since they are relatively isolated, but their density is much larger than other ZPS.

To filter out interior and isolated ZPS, we use the three-point method in fuzzy math to measure the three memberships of the elements in the PCB_v, including "low density," "medium density," and "high density." In order to prevent the extreme value of elements density from affecting the membership value, we select the normal distribution function as the membership function, and three functions are expressed as follows:

$$D1(x) = \exp\left(-\left(\frac{x - \min_{f \in \text{PCB}_v}(\rho_f)}{\sigma}\right)^2\right), \tag{9}$$

$$D3(x) = \exp\left(-\left(\frac{x - \max_{f \in \text{PCB}_v}(\rho_f)}{\sigma}\right)^2\right), \tag{10}$$

$$D2(x) = 1 - D1(x) - D3(x), \tag{11}$$

where σ is the standard deviation of the density values of all elements in PCB_v.

In Figure 3, when $\rho \in (0, M)$, the membership of the element is smaller acute-angle border element than a higher density. For example, element 1 is an acute-angular border element, and elements 2, 12, and 23 belong to obtuse-angular border elements. When $\rho = L$, the degrees of two memberships are equal. When $\rho \in (M, \max_{f \in \text{PCB}_v}(\rho_f)]$, the higher the element density is, the smaller the membership degree of the element is, which is an obtuse-border element, and the higher the membership degree of the independent objective within the cluster. When $\rho = R$, the two memberships are equal.

2.2.2. Merger Strategy. If a real cluster is mistakenly divided into several subclusters, there are some zero-pointed elements in the NCB since the NCB is not only the inner part of the actual group but also the border of subclusters. Due to the aggregation of zero-pointed objectives in the NCB, the density of NCB elements is smaller than other inner parts, which corresponds to $\rho \in (M, R)$ in Figure 3. Meanwhile, the density of PCB is in $\rho \in (0, M)$. We propose a merging

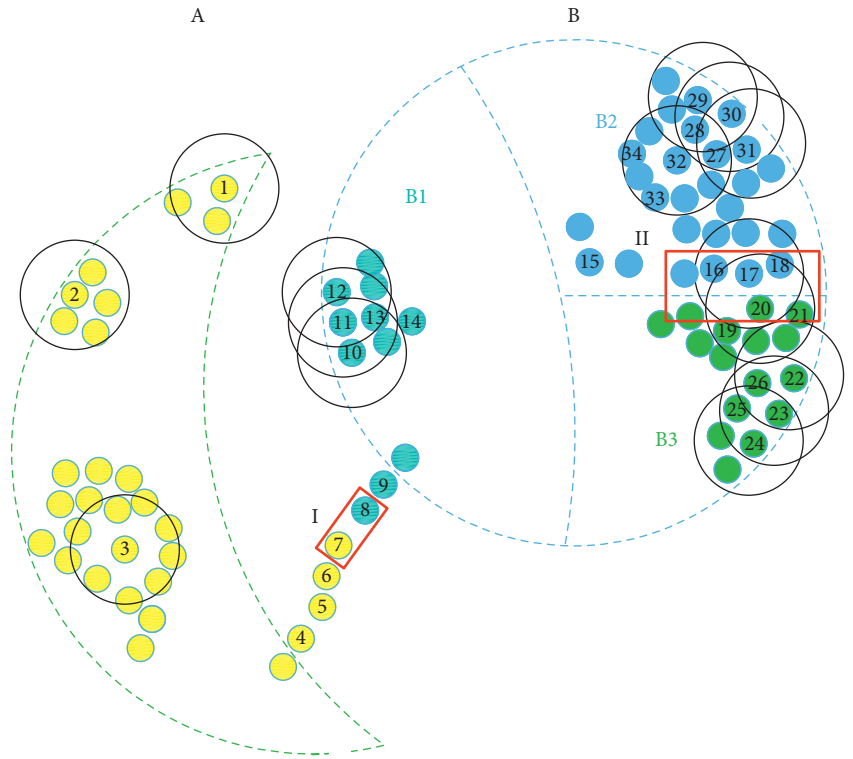


Figure 1: A schematic diagram of the distribution of the dataset, which only shows the distribution of a part of elements. The dashed lines represent two cluster borders, and the diameter of the solid black circle is dc.

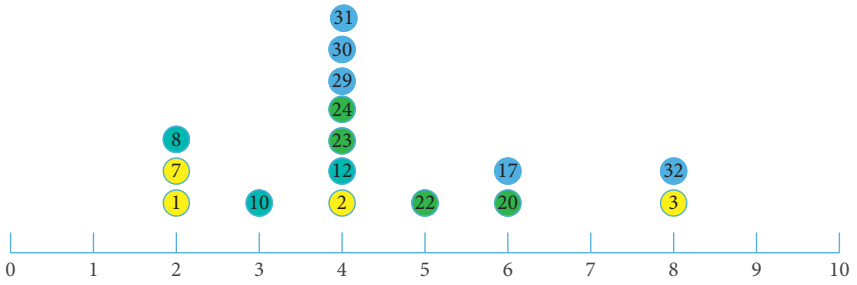


Figure 2: Density of parts of elements in initial borders calculated by equation (1).

strategy based on the comparison of element density values of NCB and PCB.

If $\exists\ c_{v_i}, c_{v_j} \in \mathrm{NCB}\ (C_v, C_{v'})$ satisfies $\rho_{c_{v_i}} > \max_{c_{v_l} \in \mathrm{PCB}_v} \rho_{c_{v_l}}$ and $\rho_{c_{v_j}} > \max_{c_{v'_l} \in \mathrm{PCB}_{v'}} \rho_{c_{v'_l}}$, where $\max_{c_{v_l} \in \mathrm{PCB}_v} \rho_{c_{v_l}}$ and $\max_{c_{v'_l} \in \mathrm{PCB}_{v'}} \rho_{c_{v'_l}}$ are equal to respective M, then C_v and $C_{v'}$ are merged; namely, if the density of the elements of the NCB is not more prominent than R but more significant than M, they must be the inner elements of the real cluster.

2.2.3. The Iteration Strategy. The δ value of each center depends on the minimum distance between the central objectives and the more significant density objectives. But when the dc is small and far from its suitable range, the algorithm does not measure the density of each sample accurately and precisely. The inexact measurement shows that, in some clusters, local center elements with more prominent local density and far from the suitable center of each group are selected, and their δ values are much larger than noncenter items. With the increase in dc, the density measurement capability gradually strengthens. The DPC-ZPS algorithm sequentially filters out fake centers with the weakest central attributes until dc $\in$ suitable range. When dc is bigger than the most significant value of the suitable range, the clusters with smaller distribution areas will be filtered out; namely, there is not the center selected by the threshold. When dc continues to increase, in the groups with a larger distribution area, the fake centers will appear again. Essentially, the process of dc increase is a gradual transition of

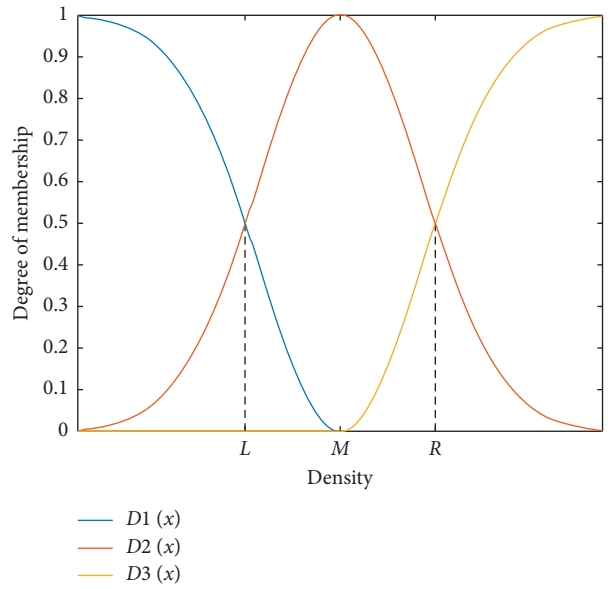


Figure 3: A schematic diagram of $D1(x)$, $D2(x)$, and $D3(x)$.

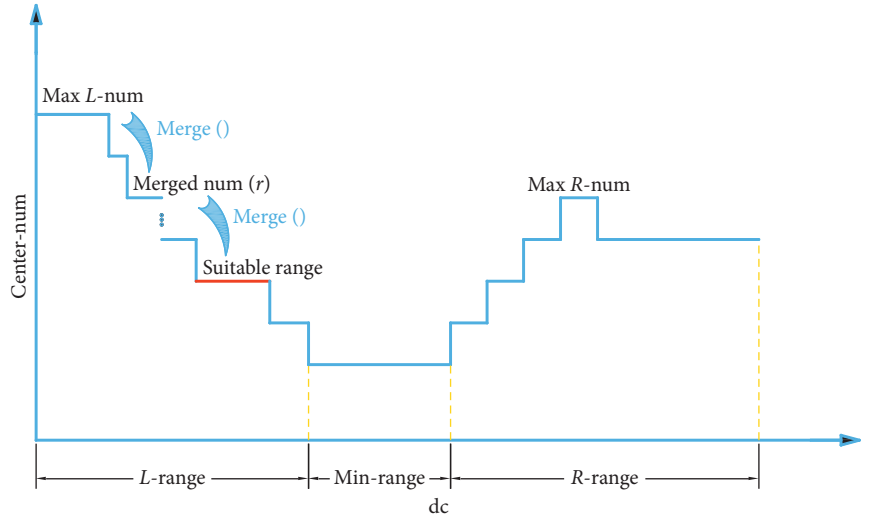


Figure 4: The ordinate is the number of cluster centers, and the abscissa is the dc; the min-range corresponds to the minimum number of centers. The left side of min-range is the left subrange (L-range), and the right side of min-range is the right subrange (R-range); max L-num and max R-num are the maximum numbers of cluster centers in the left and right subranges, respectively; the red line is the suitable range of dc.

the density metric to measure the universal density of elements from their local density. This change process is generally shown in Figure 4.

Based on the above analysis, we propose an automatic iteration strategy as follows:

Step 1: as shown in Figure 4, after counting cluster center combination and centers quantity of each dc, the algorithm determines the min-range and divides the rest into L-range and R-range. If the min-range is not only one, the DPC-ZPS chooses the biggest one to separate the dc range.

Step 2: let the algorithm find the max L-num and record its center combination as well as the sequence number of its dc.

Step 3: according to the center combination and dc, the noncenter elements are assigned to the closest element among the denser elements.

Step 4: execute merge() with clusters of clustering result from step 3.

Step 5: if the number of clusters after merge() does not change, the clustering result and the number of clusters are stores; if the number of groups reduces to merged num(r+1) from merged num(r), the third to fifth steps are repeated with the center combination corresponding to the merged num(r+1).

Step 6: the second to fifth steps are performed in the R-range after finding the max R-num.

Step 7: the final result is the maximum value of the final number of clusters in two subranges and its clustering results stored by step 5.

2.2.4. Time Complexity Analysis. Suppose that the number of samples in a dataset is n, the max center-num is N, the number of pairwise points in SNB is n_s, the max center-num in dc domain is N^t, and the number of zero-pointed samples is n_0. Just like DPC, our method needs time complexity $O(n^2)$ to calculate the distance matrix D. We search the nearest denser neighbor for each sample via a K-D tree. And the complexity of building the K-D tree is $O(n \log n)$. Searching nearest neighbor queries has an average running time of $O(\log n)$, and hence, for n groups of dc, the complexity of searching nearest neighbor of each sample queries is $O(n^2 \log n)$. For the determination of NCB, we need a matrix M, and the rows and columns represent the samples of two clusters. In the matrix M, each cell stores the distance from matrix D, and then, all distances in the M are sort in ascending order to find the NCB by equation (5). Therefore, the time complexity of NCB depends on the assignment to M, the times of assignment of the matrix M are $0.5(N^t)(N^t - 1)$, the average cost is $O(2n/N^t)$, and the total time complexity is $O((N^t - 1)n)$. How many times the operation for PCB is to be done depends on the number of zero-pointed samples, so the time complexity is less than $O(n)$. In the merger process, the density of each pairwise points is compared, and hence, the complexity of the merger depends on the number of pairwise points in SNB and is $O(0.5n_s(n_s - 1))$, where $n_s \in [0, 0.5n(n-1)]$, and only when $\mathrm{DF} = 1$, $n_s = 0.5n(n-1)$. However, the reasonable range of DF is (0, 0.05], which will be discussed in Section 3.3. Therefore, the time complexity of the merger is far less than $O(0.5n(n-1))$. And iteration is based on the max center-num, and $n \gg N$. We can conclude that the time complexity of the entire algorithm is $O(n^2 \log n)$.

3. Results and Discussion

We tested our algorithm and several related works, including PPC [44], DPC [38], DBSCAN [32], OPTICS [33], and AP [54], on several datasets. These datasets have different numbers of samples and stimulate different element distributions. The detailed information is shown in Table 1. Like DPC, AP (affinity propagation) is another advanced clustering algorithm published in *Science*. The basic idea of the AP algorithm is to treat all data points as potential cluster centers (called exemplar), then connect the data points in pairs to form a network (similarity matrix), and finally transmit the information (responsibility and availability) of each edge in the network to calculate the cluster center of each sample.

3.1. Evaluation Criteria, Parameters of Each Algorithm, and Code Sources and Preprocessing

3.1.1. Evaluation Criteria. For intuitive comparison, we chose the adjusted Rand index (ARI) [55] and adjusted mutual information (AMI) [55] to evaluate the clustering results.

Table 1: Detailed information on tested datasets.

Dataset	No. of records	No. of attributes	No. of clusters	Source
Aggregation	788	2	7	[48]
Flame	240	2	2	[49]
Spiral	312	2	3	[50]
D31	3100	2	31	[51]
R15	600	2	15	[51]
DIM512	1024	512	16	[52]
Olivetti faces	400	92×112	40	[53]

The ARI formula is shown as follows:

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\mathrm{MAX}\{\mathrm{RI}\} - E[\mathrm{RI}]}, \tag{12}$$

where E [RI] represents the expectations of RI. RI is calculated as follows:

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{C_n^2}, \tag{13}$$

where TP indicates the true positive, TN indicates the real negative, and C_n^2 is the total number of sample pairs in a dataset containing n samples.

The AMI formula is shown as follows:

$$\mathrm{AMI} = \frac{\mathrm{MI}(U,V) - E[\mathrm{MI}(U,V)]}{\mathrm{MAX}\{H(U), H(V)\} - E[\mathrm{MI}(U,V)]}, \tag{14}$$

where $H(U) = \sum_{i=1}^{|U|} P(i)\log_2 P(i)$, $H(V) = \sum_{i=1}^{|V|} P'(i)\log_2 P'(i)$, and $E[\mathrm{MI}(U,V)]$ represents the expectations of $\mathrm{MI}(U,V)$; $\mathrm{MI}(U,V)$ is expressed as follows:

$$\mathrm{MI}(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} P(i,j)\log_2 \frac{P(i,j)}{P(i)P'(j)}, \tag{15}$$

where $P(i) = |U_i|/n$, $P'(j) = |V_j|/n$, $P(i,j) = |U_i \cap V_j|/n$, $U = \{U_i \mid i = 1, 2, \ldots, |U|\}$, and $V = \{V_j \mid j = 1, 2, \ldots, |V|\}$. U and V represent two allocation methods for a dataset containing n elements, and U_i and V_j are clusters. In experimental verification, let U and V be the original labels and the clustering results of an algorithm, respectively. The value ranges of the two evaluation criteria are [−1, 1], and "1" denotes the best experimental result.

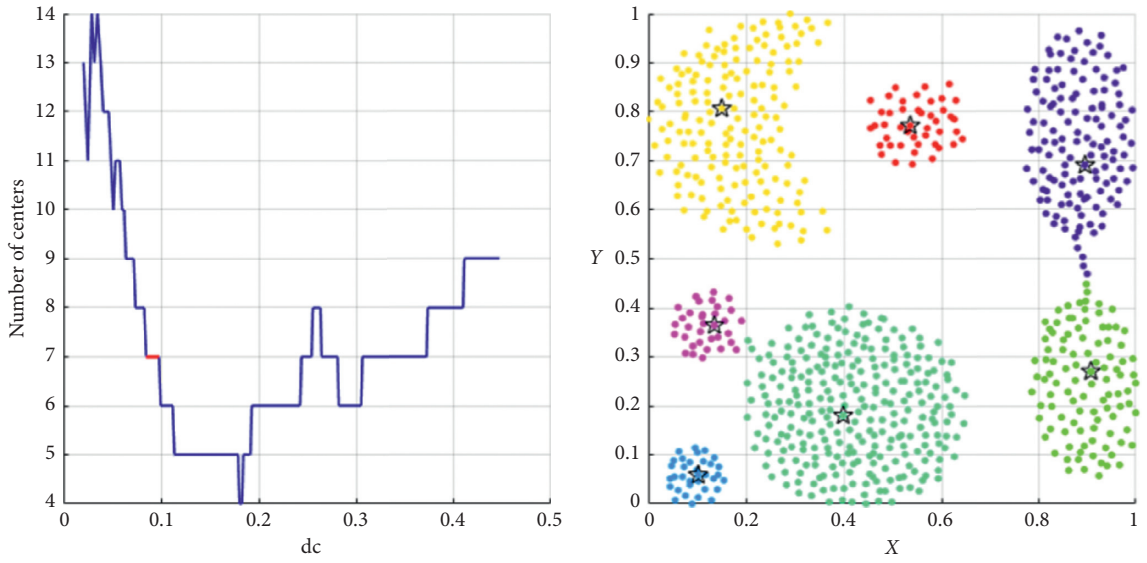
3.1.2. Parameters of Each Algorithm. DF, the parameter of our proposal, was set from 0.01 to 0.05, in which 0.005 is the interval. And by an equal interval, we choose n dc from all d_{ij} in ascending order, where n is the number of samples of a given dataset. When performing DBSACN and OPTICS experiments, we took "$(\min(d_{ij}) - \max(d_{ij}))/100$" as the step and $\min(d_{ij})$ as the initial value to attain 100 epsilons, let the minpts be from 1 to 50, and choose the best result among five thousand clustering results. During the AP experiment, we set the initial value of the unique parameter "performance" of the AP algorithm to 1.5 times the maximum value of the similarity matrix, and each cycle is reduced by 0.03%; the optimal result is selected. The specific situation is shown in Table 2, where the DPC algorithm

Table 2: Parameters setting.

Dataset	DPC-ZPS	PPC	DPC	DBSCAN	OPTICS	AP
Aggregation	0.02	0.012	0.034	0.0643/14	0.06/10	−0.96
Flame	0.03	0.027	0.028	0.1177/14	0.10/8	−2.19
Spiral	0.02	0.01	0.018	0.0418/1	0.04/1	−1.73
R15	0.02	0.015	0.006	0.0508/30	0.004/11	−0.17
D31	0.02	0.006	0.006	0.0377/37	0.03/23	−0.08
DIM512	0.02	0.039	0.006	0.36/2	0.19/1	−1
Olivetti face	0.02	0.001	0.004	0.0294/2	0.59/2	−0.247

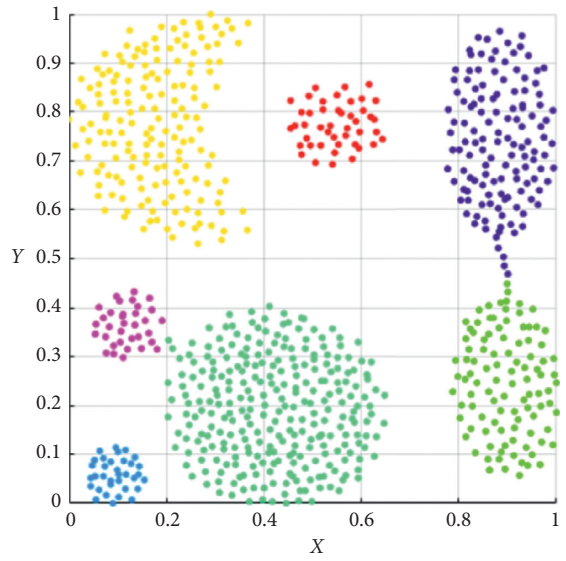
Table 3: Clustering results.

Dataset	Evaluation criteria	DPC-ZPS	PPC	DPC	DBSCAN	OPTICS	AP
Aggregation	AMI	**1.0000**	0.9922	**1.0000**	0.9785	0.9368	0.7352
	ARI	**1.0000**	0.9956	**1.0000**	0.9888	0.9747	0.6427
Flame	AMI	**1.0000**	**1.0000**	**1.0000**	0.8844	0.7385	0.3239
	ARI	**1.0000**	**1.0000**	**1.0000**	0.9550	0.8965	0.3950
Spiral	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	−0.0014
	ARI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	−0.0016
*D*31	AMI	**0.9556**	0.9554	0.9554	0.9087	0.7901	0.8563
	ARI	**0.9367**	0.9365	0.9365	0.8450	0.5814	0.7991
*R*15	AMI	**0.9938**	**0.9938**	**0.9938**	0.9916	0.9734	0.9907
	ARI	**0.9928**	**0.9928**	**0.9928**	0.9893	0.9785	0.9891
DIM512	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9029	**1.0000**
	ARI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9432	**1.0000**
Olivetti face	AMI	0.8086	**0.8447**	0.8259	0.7106	0.4286	0.7297
	ARI	**0.7385**	0.7155	0.6863	0.4668	0.5036	0.6260



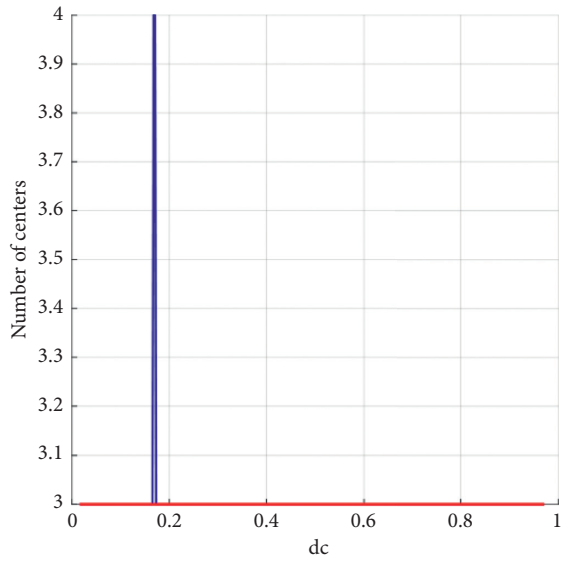
(a) (b)

Figure 5: Continued.

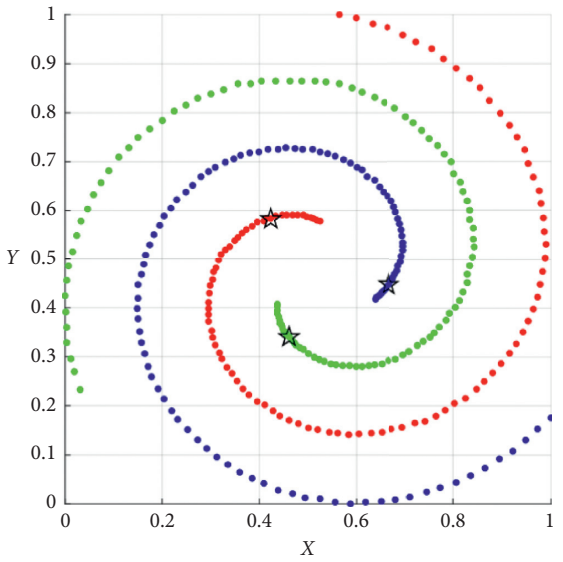


(c)

Figure 5: The analysis result of DPC-ZPS on the aggregation dataset: (a) relationship between dc and the number of centers; (b) DPC-ZPS on aggregation; (c) aggregation-ground truth.

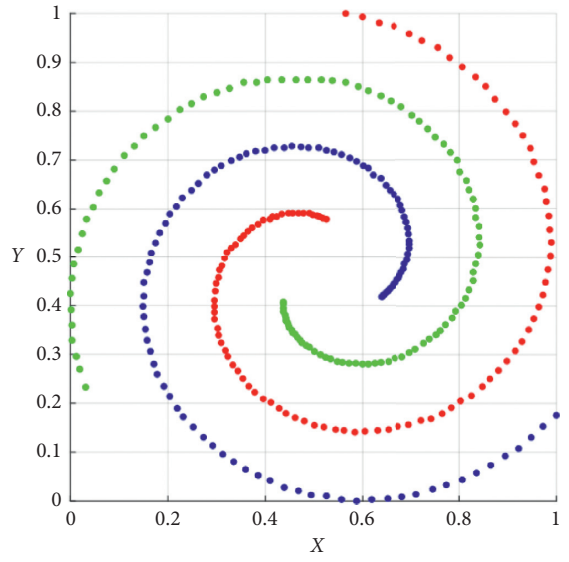


(a)



(b)

Figure 6: Continued.



(c)

Figure 6: The analysis result of DPC-ZPS on the spiral dataset: (a) relationship between dc and the number of centers; (b) DPC-ZPS on spiral; (c) spiral-ground truth.

parameter is a suitable dc, and the PPC algorithm parameter is dc_percent. The results and arguments of DPC and PPC are obtained from [44].

3.1.3. Code Sources and Preprocessing. To ensure that the experimental comparison is valid, we processed each dataset according to the method described in [25] and normalized the low-dimensional dataset and the DIM512 dataset. For preparing the Olivetti faces dataset, we first scaled each image (originally 92×112) to a smaller size of 15×15 and then performed principal component analysis (PCA) to filter out attributes of cumulative contribution rates greater than 90%. The normalization formula is as follows:

$$x'_{ij} = \frac{x_{ij} - \min\left(x_j\right)}{\max\left(x_j\right) - \min\left(x_j\right)}, \tag{16}$$

where x_{ij} represents the j^{th} value of the i^{th} data in the dataset X and $\max(x_j)$ and $\min(x_j)$ represent the maximum and minimum values of the j^{th} feature in the dataset X, respectively.

The DBSCAN codes are all built-in functions of Matlab 2019a. The OPTICS code is from the pyclustering library, the AP code is from the sklearn library, and we provide the DPC-ZPS codes. We executed all methods on a personal computer with Windows 10, Intel(R) Core (TM) i7-8750H, 16 GB memory, and Matlab 2019a or Python 3.0.

3.2. Experimental Results and Analyses. As shown in Table 3, the performance of DPC-ZPS is better than other control groups. Next, we will analyze the specific iterative process of our proposal from Figures 5–9. And each of the Figures 5–8 consist of three subgraphs. The left subgraphs represent the cutoff distance and the number of cluster centers determined by the DPC-ZPS algorithm, and the red line marks the suitable range of dc. The middle subgraph represents the clustering results of DPC-ZPS, and the right subgraph represents the category labels. Figure 9 shows the clustering results of our method and the original DPC on the Olivetti face dataset.

As shown in Figure 5, our algorithm selects seven appropriate centers and successfully converges dc to the appropriate value interval through iteration. In the iterative processes, the change of center-num in the L-range is "14-8-7-7." The number of centers remains unchanged, which means the seven clusters are relatively dependent. The final center-num of the R-range is "4," so the clustering result of the L-range is selected as the final result.

In Figure 10(a), there is a min-range, and center-num is one. And in the L-range, the process of iteration is "6-2-2," and that of the R-range is "2-1-1." Therefore, the final clustering result lies in the L-range.

In the spiral dataset, three spiral clusters are far from each other. So in Figure 6(a), in most of the dc range, there are three suitable cluster centers. There is no R-range. And our method successfully merges all subclusters to three correct groups, which is consonant with Figure 6(c).

In the L-range of $R15$, the biggest center-num is 15, and the merge does not happen, while the last center-num of the R-range is 14. Hence, the actual clustering result is determined and is shown in Figure 7(b). The change process of $D31$ L-range is from 33 to 31. The ultima center number of the R-range is approximate to

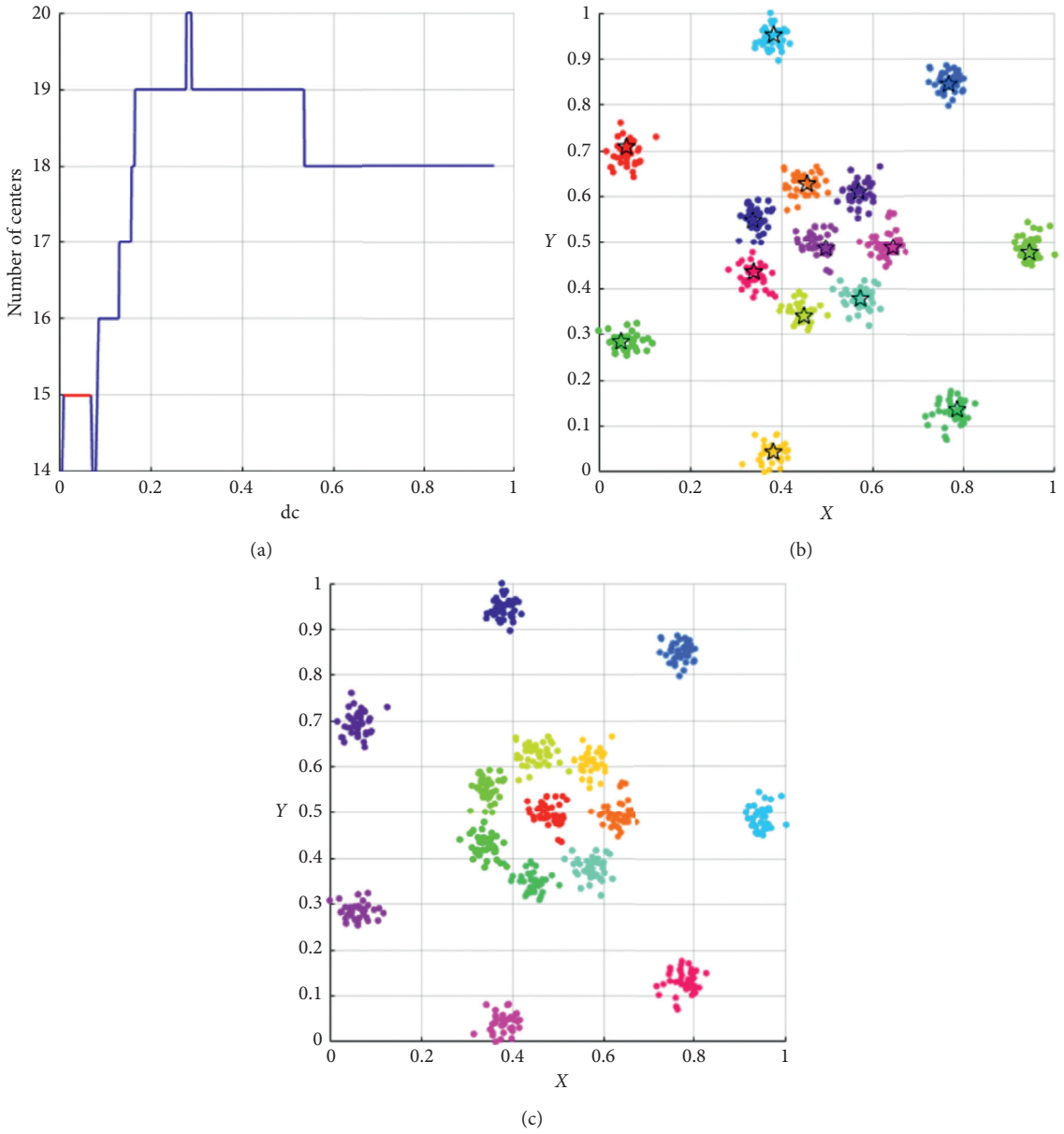


Figure 7: The analysis result of DPC-ZPS on the *D*31 dataset: (a) relationship between dc and the number of centers; (b) DPC-ZPS on *R*15; (c) *R*15 ground truth.

the minimum in Figure 8(a). Hence, the final cluster number is thirty-one.

The Olivetti faces dataset contains 40 (person) × 10 (photo) photos and is widely used in machine learning to test various algorithms. As shown in Table 3, the evaluation results of the DPC-ZPS on ARI are better than other algorithms. Figure 9 shows the clustering results of the DPC-ZPS and DPC. The image marked with a white dot in the upper right corner is the cluster center, and the gray photos indicate that there are less than three elements in the cluster.

In Figure 9(b), there are no centers in the 4th, 6th, 8th, 10th, 11st 18th, and 35th group photos, which suggest that the traditional DPC algorithm may also incorrectly merge multiple clusters into one cluster. However, as shown in Figure 9, there are only the 16th and 18th group photos without centers. It demonstrates that DPC-ZPS is less likely to merge clusters incorrectly.

3.3. Discussion. Xie et al. [39, 40, 44] manifest that the selection rule of dc provided in [38] cannot meet various

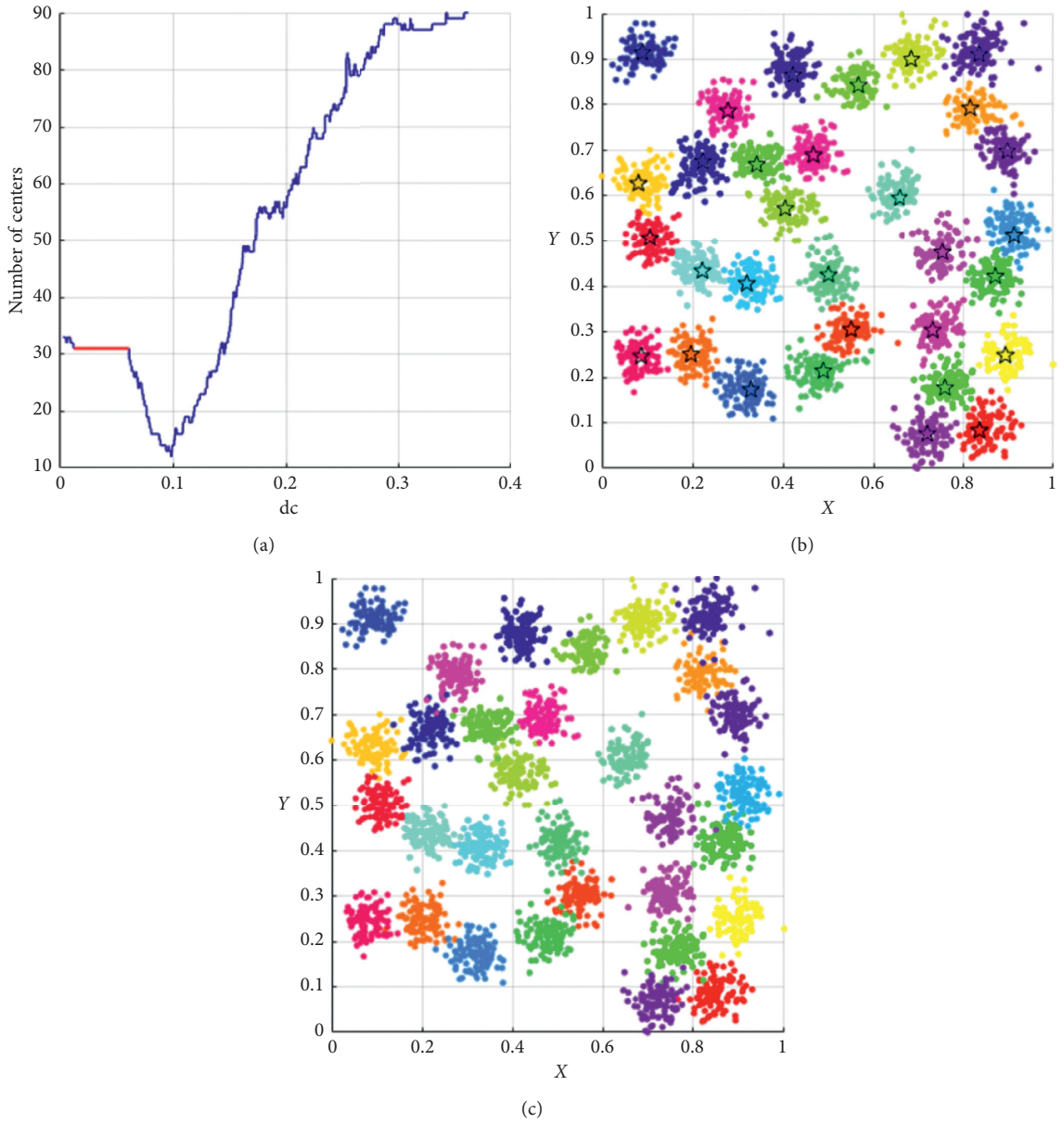
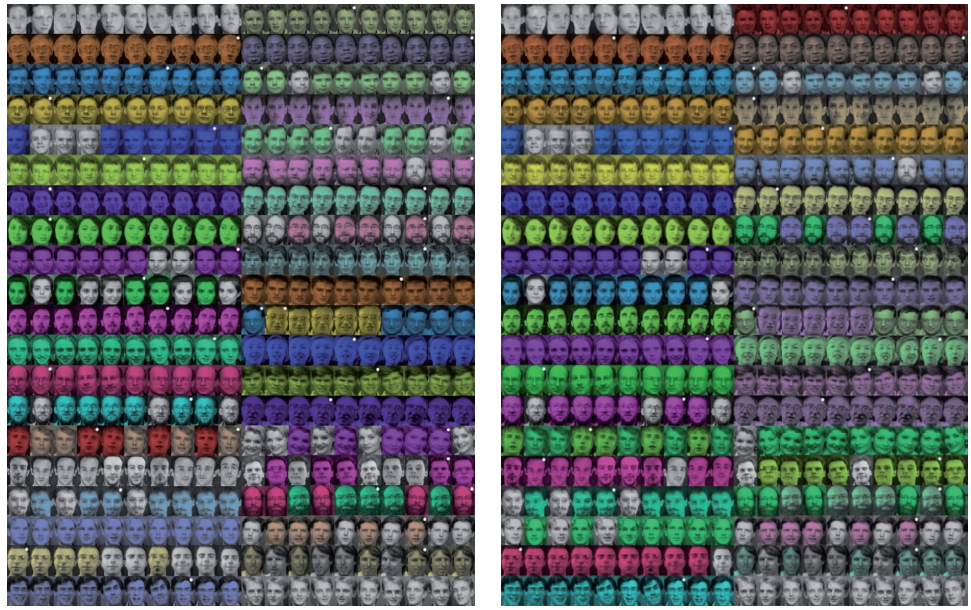


Figure 8: The analysis result of DPC-ZPS on the *D*31 dataset: (a) relationship between dc and the number of centers; (b) DPC-ZPS on *D*31; (c) *D*31 ground truth.

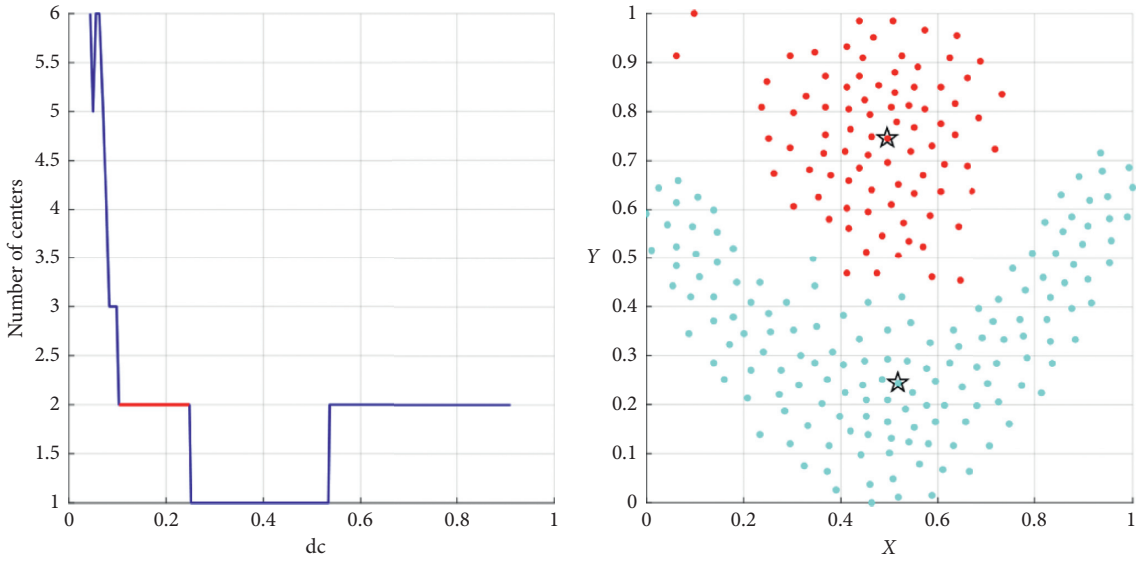
datasets. Table 2 shows that the values of dc and dc_percentage are diverse in diverse datasets, which increases the tuning cost and magnitude of difficulty, while in the six of the seven tested datasets, our argument is equal to 0.02.

The depth factor, the only parameter of the DPC-ZPS algorithm, is used in equation (6) to control the depth of the border between two adjacent clusters. When DF = 1, the neighboring cluster borders will contain all the elements in the two clusters. However, the edge should be composed of the elements with a shallow depth, so there are minimal parameter values in different datasets. Therefore, [0.005, 0.05] is a reasonable range for all of the tested datasets. As shown in Figure 11, most datasets severely fluctuate before DF = 0.015, which is just a small part of the whole; after that, our algorithm is not sensitive to the parameter changes. In addition, compared with the DPC and PPC algorithms, the DPC-ZPS algorithm does not require human intervention in the entire clustering process, which can overcome many defects caused by manual operation.



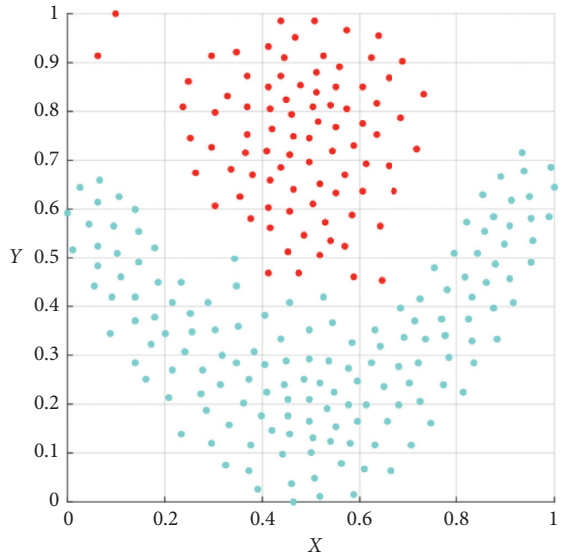
(a) (b)

Figure 9: The clustering results on Olivetti faces by (a) DPC-ZPS and (b) DPC.



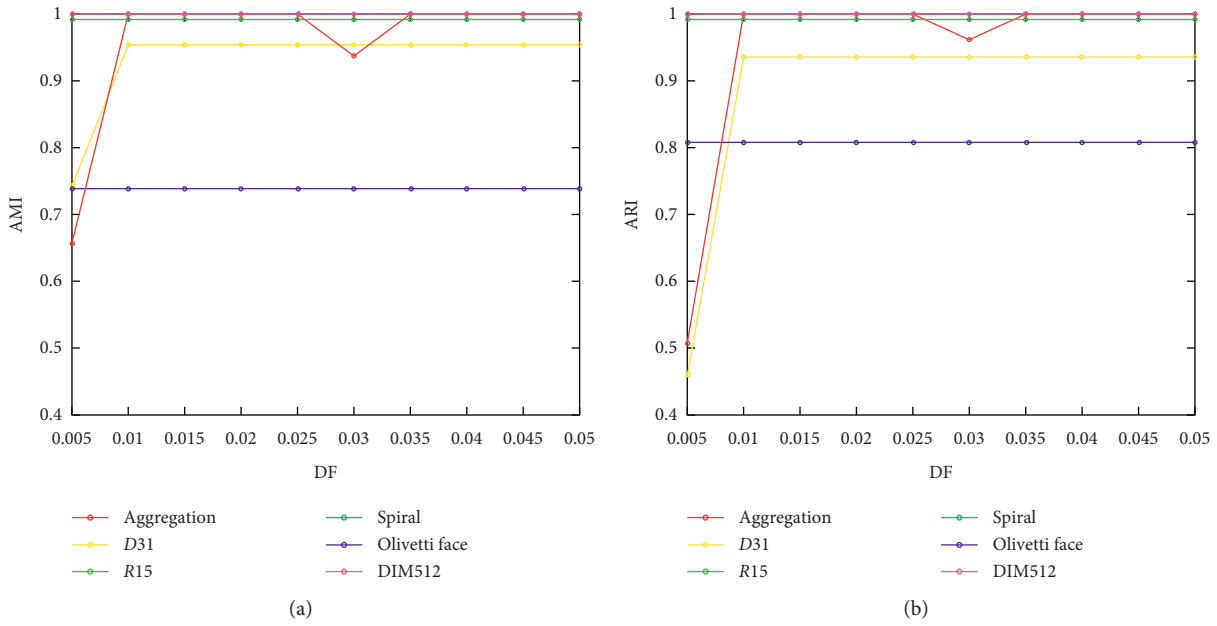
(a) (b)

Figure 10: Continued.



(c)

FIGURE 10: The analysis result of DPC-ZPS on the flame dataset: (a) relationship between dc and the number of centers; (b) DPC-ZPS on flame; (c) flame-ground truth.



(a) (b)

FIGURE 11: Results on different datasets with different depth factors.

4. Conclusions

In this paper, to overcome the defects of human operation and the difficulty in determination of the suitable dc, we proposed the density peaks clustering by zero-pointed samples (DPC-ZPSs) of regional group borders. DPC-ZPS is based on the in-depth analyses of not only the changing rule between the dc and centers but also the relationship between the density of NCB and PCB. Our proposal covers two main parts: the merger strategy of subclusters based on the cluster borders and the iteration strategy. The merger strategy adaptively determines the threshold of merge for each pairwise local cluster. And the iterative process is to find a suitable range of dc automatically. And experimental results indicate our method is more accurate without artificial operation and has a more reasonable and less sensitive threshold value range. Additionally, we will use the natural nearest neighbors to optimize the local density measurement and assignment process.

Data Availability

All datasets in this paper are from UCI. All readers are able to access datasets from it.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61972056, 61772454, 61402053, and 61981340416), the Natural Science Foundation of Hunan Province of China (2020JJ4623), the Scientific Research Fund of Hunan Provincial Education Department (17A007, 19C0028, and 19B005), the Changsha Science and Technology Planning (KQ1703018, KQ1706064, KQ1703018-01, and KQ1703018-04), the Junior Faculty Development Program Project of Changsha University of Science and Technology (2019QJCZ011), the "Double First-class" International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology (2019IC34), the Practical Innovation and Entrepreneurship Ability Improvement Plan for Professional Degree Postgraduate of Changsha University of Science and Technology (SJCX202072), the Postgraduate Training Innovation Base Construction Project of Hunan Province (2019-248-51).and the Beidou Micro Project of Hunan Provincial Education Department (XJT[2020] No.149).

References

[1] A. Saxena, M. Prasad, A. Gupta et al., "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[2] Y. Chen, J. Wang, S. Liu et al., "Multiscale fast correlation filtering tracking algorithm based on a feature fusion model," *Concurrency and Computation: Practice and Experience*, p. e5533, 2019.

[3] Z. Liao, R. Zhang, S. He, D. Zeng, J. Wang, and H.-J. Kim, "Deep learning-based data storage for low latency in data center networks," *IEEE Access*, vol. 7, pp. 26411–26417, 2019.

[4] Y. Chen, J. Tao, Q. Zhang et al., "Saliency detection via the improved hierarchical principal component analysis method," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8822777, 12 pages, 2020.

[5] F. Yu, L. Liu, H. Shen et al., "Dynamic analysis, circuit design and Synchronization of a novel 6D memristive four-wing hyperchaotic system with multiple coexisting attractors," *Complexity*, vol. 2020, Article ID 5904607, 17 pages, 2020.

[6] Y. Chen, J. Wang, X. Chen et al., "Single-image super-resolution algorithm based on structural self-similarity and deformation block features," *IEEE Access*, vol. 7, pp. 58791–58801, 2019.

[7] F. Yu, L. Liu, S. Qian et al., "Chaos-based application of a novel multistable 5D memristive hyperchaotic system with coexisting multiple attractors," *Complexity*, vol. 2020, Article ID 8034196, 19 pages, 2020.

[8] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier," *Cluster Computing*, vol. 22, no. S3, pp. 7665–7675, 2019.

[9] F. Yu, H. Shen, L. Liu et al., "CCII and FPGA realization: a multistable modified four-order autonomous Chua's chaotic system with coexisting multiple attractors," *Complexity*, vol. 2020, Article ID 5212601, 17 pages, 2020.

[10] Y. Chen, J. Xiong, W. Xu, and J. Zuo, "A novel online incremental and decremental learning algorithm based on variable support vector machine," *Cluster Computing*, vol. 22, no. S3, pp. 7435–7445, 2019.

[11] J. Zhang, Y. Wu, W. Feng, and J. Wang, "Spatially attentive visual tracking using multi-model adaptive response fusion," *IEEE Access*, vol. 7, pp. 83873–83887, 2019.

[12] W. Li, H. Xu, H. Li et al., "Complexity and algorithms for superposed data uploading problem in networks with smart devices," *IEEE Internet of Things Journal*, 2019.

[13] K. Gu, N. Wu, B. Yin, and W. Jia, "Secure data query framework for cloud and fog computing," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 332–345, 2020.

[14] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[15] Y. Chen, J. Tao, L. Liu et al., "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, p. 1, 2020.

[16] Y. Chen, J. Wang, R. Xia, Q. Zhang, Z. Cao, and K. Yang, "The visual object tracking algorithm research based on adaptive combination kernel," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855–4867, 2019.

[17] O. O. Olugbara, E. Adetiba, S. A. Oyewole, and S. A. Oyewole, "Pixel intensity clustering algorithm for multilevel image segmentation," *Mathematical Problems in Engineering*, vol. 2015, Article ID 649802, 19 pages, 2015.

[18] Z. Hong, H. He, J. Xu, Q. Fang, and W. Wang, "Medical image segmentation using fruit fly optimization and density peaks clustering," *Computational and Mathematical Methods in Medicine*, vol. 2018, Article ID 3052852, 11 pages, 2018.

[19] T. Vo-Van, A. Nguyen-Hai, M. V. Tat-Hong, and T. Nguyen-Trang, "A new clustering algorithm and its application in assessing the quality of underground water," *Scientific Programming*, vol. 2020, Article ID 6458576, 12 pages, 2020.

[20] C. Ju and C. Xu, "A new collaborative recommendation approach based on users clustering using artificial bee colony algorithm," *The Scientific World Journal*, vol. 2013, Article ID 869658, 9 pages, 2013.

[21] H. Qu, L. Lei, X. Tang, and P. Wang, "A lightweight intrusion detection method based on fuzzy clustering algorithm for wireless sensor networks," *Advances in Fuzzy Systems*, vol. 2018, Article ID 4071851, 12 pages, 2018.

[22] A. Amineh, H. Saboohi, T.-Y. Wah, and T. Herawan, "A fast density-based clustering algorithm for real-time internet of things stream," *The Scientific World Journal*, vol. 2014, Article ID 926020, 11 pages, 2014.

[23] D. Lam and D. C. Wunsch, "Clustering," *Academic Press Library in Signal Processing*, vol. 1, pp. 1115–1149, 2014.

[24] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, Oakland, CA, USA, 1967.

[25] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[26] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[28] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch," *ACM Sigmod Record*, vol. 25, no. 2, pp. 103–114, 1996.

[29] J. Zhong, P. W. Tse, and Y. Wei, "An intelligent and improved density and distance-based clustering approach for industrial survey data classification," *Expert Systems with Applications*, vol. 68, pp. 21–28, 2017.

[30] S. Guha, R. Rastogi, and K. Shim, "Cure," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data ACM*, pp. 73–84, Seattle, WA, USA, 1998.

[31] S. Guha, R. Rastogi, and K. Shim, "Rock: a robust clustering algorithm for categorical attributes," in *Proceedings of the IEEE Conference on Data Engineering*, pp. 512–521, Sydney, Australia, March 1999.

[32] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, Portland, OR, USA, 1996.

[33] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *Proceedings of the ACM Sigmod Record*, pp. 49–60, Philadelphia, PA, USA, 1999.

[34] W. Wang, J. Yang, and R. Muntz, "Sting: a statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 186–195, Athens, Greece, August 1997.

[35] G. McLachlan and D. Peel, "Finite mixture models," in *Encyclopedia of Autism Spectrum Disorders*, F. R. Volkmar, Ed., p. 1296, 1st edition, Springer, New York, NY, USA, 2013.

[36] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[37] I. Anderson and R. Diestel, "Graph-theory," *The Mathematical Gazette*, vol. 85, no. 502, p. 176, 2001.

[38] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[39] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

[40] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Information Sciences*, vol. 450, pp. 200–226, 2018.

[41] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, 2016.

[42] P. Guo, X. Wang, Y. Wang, Y. Chen, and Y. Zhang, "Research on automatic determining clustering centers algorithm based on linear regression analysis," in *Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 1016–1023, Chengdu, China, June 2017.

[43] J. Ding, X. He, J. Yuan, and B. Jiang, "Automatic clustering based on density peak detection using generalized extreme value distribution," *Soft Computing*, vol. 22, no. 9, pp. 2777–2796, 2018.

[44] L. Ni, W. Luo, W. Zhu, and W. Liu, "Clustering by finding prominent peaks in density space," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 727–739, 2019.

[45] Y. Luo, J. Qin, X. Xiang, Y. Tan, and Q. Liu, "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2020.

[46] Y. Tan, J. Qin, X. Xiang, W. Ma, W. Pan, and N. N. Xiong, "A robust watermarking scheme in YCbCr color space based on channel coding," *IEEE Access*, vol. 7, no. 1, pp. 25026–25036, 2019.

[47] B. Yin, X. Wei, J. Wang, N. Xiong, and K. Ge, "An industrial dynamic skyline based similarity joins for multi-dimensional big data applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2520–2532, 2020.

[48] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, p. 4, 2007.

[49] L. Fu and E. Medico, "Flame, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinformatics*, vol. 8, no. 1, p. 3, 2007.

[50] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[51] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1273–1280, 2002.

[52] P. Franti, O. Virmajoki, and V. Hautamaki, "Fast agglomerative clustering using a k-nearest neighbor graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1875–1881, 2006.

[53] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of the 1994 IEEE Workshop on Applications of Computer Vision*, pp. 138–142, Sarasota, FL, USA, December 1994.

[54] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[55] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8893419, 11 pages
https://doi.org/10.1155/2020/8893419

Research Article

High-Resolution Radar Target Recognition via Inception-Based VGG (IVGG) Networks

Wei Wang,[1] Chengwen Zhang,[1] Jinge Tian,[1] Xin Wang,[1] Jianping Ou,[2] Jun Zhang,[2] and Ji Li[1]

[1]*School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2]*ATR Key Laboratory, National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Jun Zhang; zhj64068@sina.com and Ji Li; hangliji@163.com

Received 13 June 2020; Revised 21 June 2020; Accepted 1 July 2020; Published 18 July 2020

Academic Editor: Nian Zhang

Aiming at high-resolution radar target recognition, new convolutional neural networks, namely, Inception-based VGG (IVGG) networks, are proposed to classify and recognize different targets in high range resolution profile (HRRP) and synthetic aperture radar (SAR) signals. The IVGG networks have been improved in two aspects. One is to adjust the connection mode of the full connection layer. The other is to introduce the Inception module into the visual geometry group (VGG) network to make the network structure more suik / for radar target recognition. After the Inception module, we also add a point convolutional layer to strengthen the nonlinearity of the network. Compared with the VGG network, IVGG networks are simpler and have fewer parameters. The experiments are compared with GoogLeNet, ResNet18, DenseNet121, and VGG on 4 datasets. The experimental results show that the IVGG networks have better accuracies than the existing convolutional neural networks.

1. Introduction

Radar automatic target recognition (RATR) technology can provide inherent characteristics of the target, such as the attributes, categories, and models, and these characteristics can provide richer information for battlefield command decisions. The high-resolution radar echo signal obtained from the wide bandwidth signal transmitted by the broadband radar provides more detailed features of the target, which makes it possible to identify the target type. Therefore, more and more research studies focus on RATR technology.

Traditional methods of radar target automatic recognition include k-nearest neighbor classifier (KNN) and support vector machine learning (SVM) and so on. Zhao and Principe [1] applied support vector machine to automatic target recognition of SAR image. Obozinski et al. [2] proposed the Trace-norm Regularized multitask learning method (TRACE) to solve the problem of recovering a set of common covariates related to several classification problems at the same time. It assumes that all models share a common low-dimensional subspace, but the method cannot be extended to the nonlinear field well. Regularized multitask learning (RMTL) proposed by Evgeniou and Pontil [3] extends the existing kernel-based learning methods of single-task learning, such as SVM. Zhou et al. [4] proposed the clustered multitask learning (CMTL) method to replace multitask learning (MTL). It assumes that multiple tasks follow the cluster structure and achieves high recognition accuracy of SAR image. Zhang and Yeung [5] proposed the multitask relationship learning (MTRL) method, which can learn the correlation between positive and negative tasks autonomously, and the recognition accuracy is higher than that of CMTL. Cong et al. [6] proposed a new classification method by improving MTRL, which can autonomously learn multitask relationship and cluster information of different tasks and be easily expanded to the nonlinear domain. He et al. [7] used the principal component analysis (PCA) method to realize the fast target recognition of SAR image.

With the development of artificial intelligence, more and more applications based on neural networks are used for target recognition [8]. In the field of image target recognition, convolutional neural network (CNN) has achieved great success, which is widely used in object detection and

localization, semantic segmentation, target recognition, and so on [9]. Visual geometry group networks (VGGNets) [10] proposed by Simonyan and Zisserman have significantly improved image recognition accuracy by deepening the network depth to 19 layers. In the same year, GoogLeNet [11] proposed by Christian Szegedy used the Inception module to have several parallel convolution routes for extracting input features, which widened the network structure horizontally and deepened the network depth to a certain extent while the network parameters are reduced. Studies have shown that deeper networks have better performance, but deepening the network is faced with the problem of gradient disappearance, and the complex networks also have the risk of overfitting. Residual networks (ResNets) [12] and dense convolutional network (DenseNet) [13, 14] solve the above problems by using skip connections and significantly increase the depth of the network. Recently proposed highway networks, ResNets, and DenseNet have deepened the network structure to more than 100 layers and demonstrated outstanding performance in the field of image recognition.

Different from image data, radar data are sparse and have a little amount. Therefore, the network should be able to extract multidimensional features, and the depth could not be too deep. So, we considered using the Inception module and VGG network for training. VGG networks have limited depth and been proven to have excellent feature extraction capabilities. The Inception module has multipath convolution, which can extract radar multidimensional information for learning, and its internal large-scale convolution kernels are also more effective to extract the information with sparse characteristics. Therefore, we proposed a method to fuse the Inception module with the VGG network.

This paper focuses on target recognition based on 1D HRRP and SAR images and proposes the IVGG convolutional neural network structure which is most suitable for high-resolution radar target recognition. The parameters of IVGG can also be greatly reduced.

2. Target Recognition Model: IVGG Networks

2.1. VGGNets. VGGNets [10] adopted the convolution filters with a small local receptive field and proposed 6 different network configurations. In VGGNets, the convolution filters are set to 3×3 and the max-pooling is 2×2, with stride 2.

The contribution of the VGGNet is the application of the 3×3 small convolution filters. By stacking small convolution filters, the depth of the network is increased, and the nonlinearity of the convolutional layers is strengthened too [15]. Therefore, the nonlinear function can be better fitted (but the overfitting phenomenon needs to be prevented) and the parameters of the network are reduced.

Before the VGG network was proposed, An et al. also used small convolution filters, but the network was not as deep as VGGNet [16]. The VGGNet has better performance than other convolutional networks in extracting target features.

In the structure of VGGNet, the convolutional layers and pooling layers alternately appear. After two to four convolutional layers, a max-pooling layer is followed. In order to keep the computational complexity of the constituent structures at each feature layer roughly consistent, the number of convolution kernels at the next layer is doubled when the size of the feature map is reduced by half through the max-pooling layer. VGGNet ends with three fully connected layers, which are also the classifier for the system.

2.2. The Improved Model: IVGG Network. Because SAR images and HRRP data are sparse, it is difficult to fully represent all the feature information of the targets by using all 3×3 convolution filters. GoogLeNet, proposed by Christian Szegedy [11], uses the Inception modules with larger convolution filters, which can extract radar multidimensional information for learning. As shown in Figure 1, there are several parallel convolutional lines in the Inception module, and the large convolution filters in parallel lines increase the width and the depth of the network structure. So, the Inception module is used to modify the VGG module. The new network is specially designed for radar data analysis and has a high recognition rate of radar target models. The principle of improvement will be introduced in the next section.

In this paper, the "Conv" module includes convolution, batch standardization, and activation functions, as shown in Figure 2.

Based on the above structures, we propose 4 new IVGG networks. In this structure, a certain number of Inception modules are used to replace "Conv3" module in the original VGGNets. Note that we add a very deep point convolutional layer after the Inception module, and it is important. Many traditional algorithms show poor performance for radar target recognition is because they cannot effectively fit the nonlinear structure in the radar signal [6]. Drawing on this point of view, we have strengthened the nonlinear capabilities of IVGG by adding a point convolutional layer. Immediately following the Inception module, the layer contains activation function, which increases the nonlinearity of the network. Further, we set the input number of channels is same with output. In other words, the point convolutional layer does not compress the output feature maps. It also strengthens the nonlinearity of the network. Table 1 shows the specific configuration of the IVGG networks, where the Inception module and Conv1 module which are used to replace Conv3 modules in the original network are identified in italics.

The fully connected layers of VGGNets are shown in Table 2. Since there have 3 layers, we use "3FC" to refer to the structure in Table 2.

The classifier of the VGG networks is fully connected layers, containing most of the parameters of the whole network. In order to reduce the parameters, we improved the FC layers, reducing the 3-layer FC to a single-layer "FC-4/10", which is represented by "1FC".

In the experiment, the network we proposed relates to the above two classifiers, which can be represented by

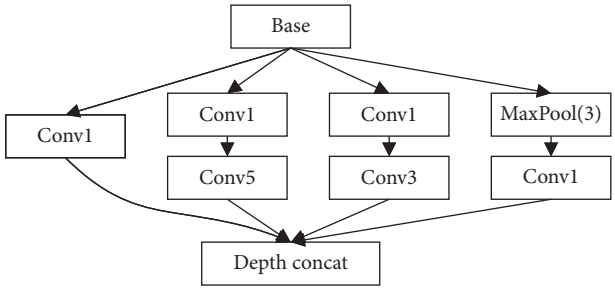


Figure 1: The Inception module, where Conv1 means the convolutional filter is 1×1, Conv3 means the convolutional filter is 3×3, and Conv5 means the convolutional filter is 5×5.

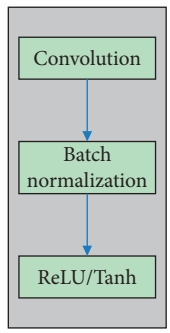


Figure 2: "Conv" module.

Table 1: IVGG network configuration.

IVGG11	IVGG13	IVGG16	IVGG19
11 weight layers	13 weight layers	16 weight layers	19 weight layers
Input (HRRP OR SAR)			
conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
MaxPool			
conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
MaxPool *Inception-256* *conv1-256* conv3-256	conv3-256 *Inception-256* *conv1-256*	conv3-256 *Inception-256* *conv1-256* conv3-256	conv3-256 *Inception-256* *conv1-256* conv3-256 conv3-256
MaxPool *Inception-512* *conv1-512* *Inception-512* *conv1-512*	conv3-512 conv3-512	conv3-512 *Inception-512* *conv1-512* *Inception-512* *conv1-512*	*Inception-512* *conv1-512* conv3-512 conv3-512 *Inception-512* *conv1-512*
MaxPool conv3-512 conv3-512	*Inception-512* *conv1-512* *Inception-512* *conv1-512*	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
MaxPool			
Fully connected layers			
Soft-max			

Table 2: Three fully connected layers (3FC).

FC-4096
FC-4096
FC-4/10

"*IVGGx*-1FC" and "*IVGGx*-3FC," respectively, where $\boldsymbol{x}$ is the network depth.

The IVGG11 network is shown in Figure 3, the structure shows how conv3 modules are replaced, and the other networks with different depths (IVGG13/16/19) in Table 1 also follow this rule.

3. Characteristic Analysis of IVGG Networks

3.1. Relationship between Data Sparsity and Network Structure. In this section, we perform theoretical analysis to demonstrate the sparse characteristics of 3×3 filters and 5×5 filters. It can further explain that the IVGG network can overcome the target recognition difficulties caused by sparse radar data to some extent.

Assume that in the convolution layers, the weight tensor is s $\mathbf{W} \in \mathbb{R}^{C_{\text{in}} \times C_{\text{out}} \times (k_1 k_2)}$, where C_{in} is the number of input channels, C_{out} is the number of output channels, and k_1 and k_2 are the convolutional kernel size. Considering the calculation process of convolution filters and feature map in each channel, the weight matrix of the filter is $\mathbf{W}_{\text{filter}} \in \mathbb{R}^{k_1 \times k_2}$. We unfold the weight matrix into a vector $\mathbf{w} \in \mathbb{R}^{k_1 k_2}$. Each local receptive field in the input (considering a certain channel) is expanded into a vector $\mathbf{x}$, and then $\mathbf{w}^T\mathbf{X}$ represents the output, where the matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, and the number of elements in the output feature map is represented by N.

If the kernel size of a convolution layer is (k_1, k_2), weight tensor $\mathbf{w}^T = (w_1, w_2, \ldots, w_{k_1 \times k_2})^T$, the output feature map can be represented as follows:

$$\begin{aligned}
\mathbf{w}^T\mathbf{X} &= \left(\mathbf{w}^{\mathbf{T}}\mathbf{x}_1, \mathbf{w}^{\mathbf{T}}\mathbf{x}_2, \ldots, \mathbf{w}^{\mathbf{T}}\mathbf{x}_N\right) = \left(\sum_{i=1}^{k_1 \times k_2} w_i x_{i,1}, \sum_{i=1}^{k_1 \times k_2} w_i x_{i,2}, \ldots, \sum_{i=1}^{k_1 \times k_2} w_i x_{i,N}\right), \\
\mathbf{w}^T\mathbf{X}_0 &= \#\left(n = (1, 2, \ldots, N) \mid \sum_{i=1}^{k_1 \times k_2} w_i x_{i,n} \neq 0\right), \\
\mathbf{w}^T\mathbf{X}_1 &= \sum_{n=1}^{N} \left| \sum_{i=1}^{k_1 \times k_2} w_i x_{i,n} \right|.
\end{aligned} \tag{1}$$

Assume the elements in matrix $\mathbf{X}$ are set to zero in probability P_1 ($P_1 < 1$), the weight vector $\mathbf{w}$ element values w_i are set to zero in probability P_2, that is, $P\{w_i\} = P_2$. When $P_1 \longrightarrow 1$, $\mathbf{X}_0 \longrightarrow 0, \forall n = (1, 2, \ldots, N)$, the probability when the neuron is activated is as follows:

$$\begin{aligned}
\lim_{P_1 \longrightarrow 1} P\left\{\sum_{i=1}^{5\times5} w_i x_{i,n} \neq 0\right\} &= \lim_{P_1 \longrightarrow 1}\left(1 - \left(1 - \left(1 - P_1\right)\left(1 - P_2\right)\right)^{25}\right) \\
&= \lim_{P_1 \longrightarrow 1}\left(1 - \left(1 - \left(1 - P_1 - P_2 + P_1P_2\right)\right)^{25}\right) \\
&= 1 - \left(1 - \left(1 - P_1 - P_2 + P_2\right)\right)^{25} \\
&= 1 - \left(1 - \left(1 - P_1\right)\right)^{25} = 1 - P_1^{25}.
\end{aligned} \tag{2}$$

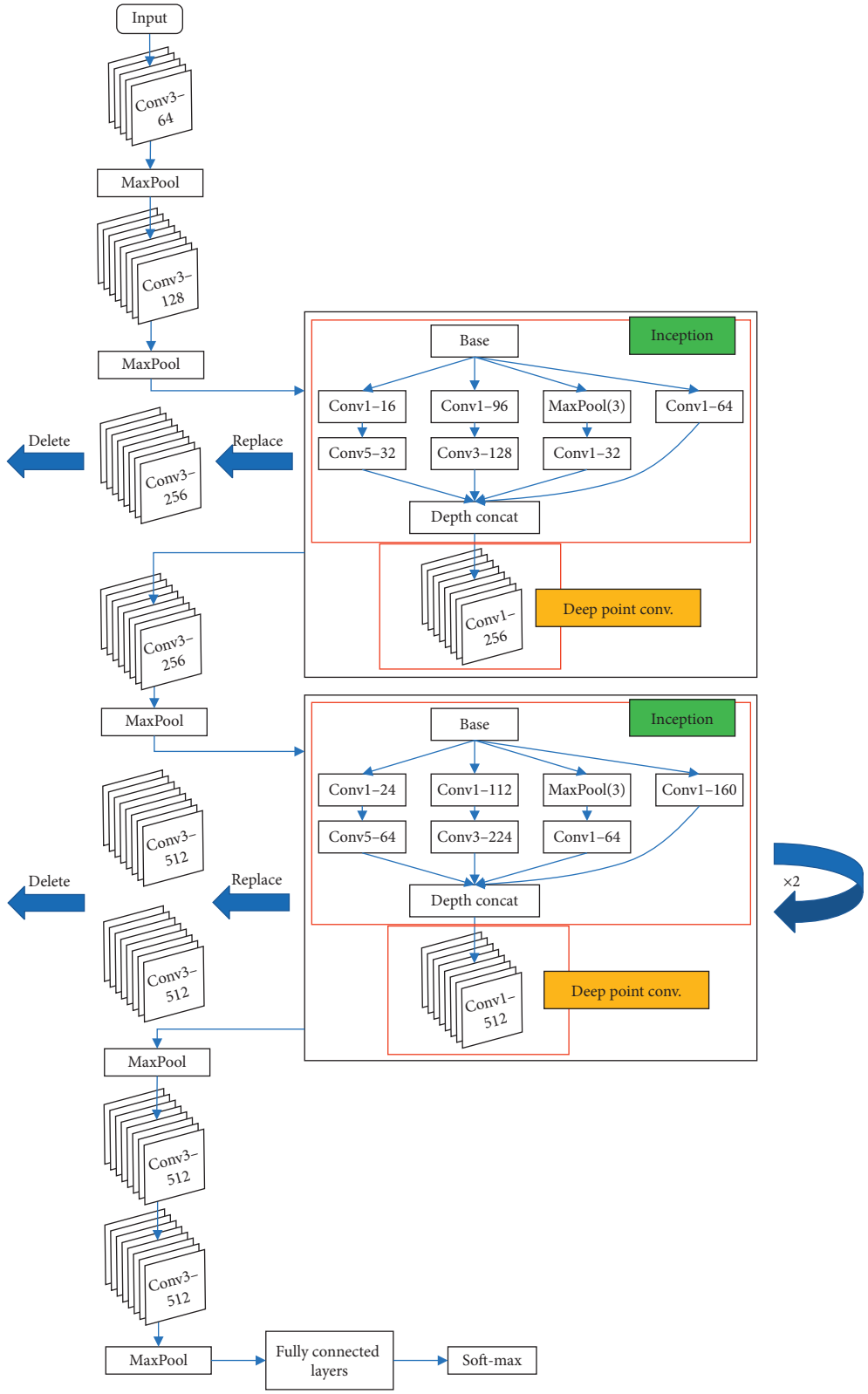


Figure 3: IVGG11 network architecture.

Similarly, we can get the following expression:

$$\lim_{P_1 \longrightarrow 1} P\left\{\sum_{i=1}^{3\times 3} w_i x_{i,n} \neq 0\right\} = 1 - P_1^9. \tag{3}$$

So, we can get the following inequality:

$$\therefore \lim_{P_1 \longrightarrow 1} P\left\{\sum_{i=1}^{5\times 5} w_i x_{i,n} \neq 0\right\} > \lim_{P_1 \longrightarrow 1} P\left\{\sum_{i=1}^{3\times 3} w_i x_{i,n} \neq 0\right\}. \tag{4}$$

When $k_1 = 3, k_2 = 3$, we use a_0 and a_1 to denote $\mathbf{w}^T\mathbf{X}_0$ and $\mathbf{w}^T\mathbf{X}_1$. When $k_1 = 5, k_2 = 5$, we use b_0 and b_1 to denote $\mathbf{w}^T\mathbf{X}_0$ and $\mathbf{w}^T\mathbf{X}_1$. Then,

$$\begin{aligned} a_0 &= N\left(1 - P_1^9\right), \\ b_0 &= N\left(1 - P_1^{25}\right), \\ a_0 &< b_0. \end{aligned} \tag{5}$$

For the convenience of calculations, we assume that input feature vector/tensor does zero padding. Because $P_1 \longrightarrow 1$, this does not affect the calculation result. Then, we have

$$\begin{aligned} a_1 &= \mathbf{w}^T\mathbf{X}_1 = \sum_{n=1}^{N}\left|\sum_{i=1}^{3\times 3} w_i x_{i,n}\right|, \\ b_1 &= \mathbf{w}^T\mathbf{X}_1 = \sum_{n=1}^{N}\left|\sum_{i=1}^{5\times 5} w_i x_{i,n}\right|. \end{aligned} \tag{6}$$

It is easy to prove $a_1 < b_1$. Therefore, the large-scale convolution kernel can effectively extract the target features if the input data are too sparse.

The sparsity of the convolutional layer can bring many benefits, such as better robustness and higher feature extraction efficiency. However, if the input data are excessive sparse, feature extraction will become more difficult. Therefore, after repeated experiments, we finally chose the Inception module instead of the larger convolution kernel. We just added an appropriate number of Inception module to the network, and they are not all composed of Inception modules like GoogLeNet. In order to improve the network's ability to fit nonlinear structures in radar data (such as SAR images), we add a very deep point convolutional layer behind the Inception module. It should be noted that the point convolutional layer introduces an activation function, and the channels of input and output channels are the same, which improves the nonlinearity of the new network.

3.2. The Parameter Number of the Networks. As shown in Figure 4, our method has about 3 million parameters less than the VGG network at the same depth. The number of parameters of networks connected to the above two classifiers is shown in Table 3. By improving the classifier, our network can further reduce the parameter amount by 86%–92%.

The comparisons of floating points of operations (FLOPs) are shown in Figure 5. According to Figure 5, the computation cost is most affected by the network depth. IVGG16 and IVGG19 are very computation-intensive. It can be seen from Figure 4 that at the same number of network layers, the FLOPs of IVGG are significantly less than those of the VGG networks. For example, IVGG16-3FC saves 23.61% FLOPs compared to VGG19. So, our methods not only save parameter storage space, but also reduce computation cost.

4. Experiment and Results Analysis

4.1. Dataset. The SAR image dataset used in this paper is a public dataset released by MSTAR. There are many research studies on radar automatic target recognition based on the MATAR SAR dataset, such as references [1–4, 17–20]. The experimental results in this paper are compared with the above methods. The MSTAR dataset and the HRRP dataset are used for experiments. Published by MSTAR [6, 21], the SAR dataset includes ground-based military targets. The acquisition conditions of the MSTAR dataset are classified into standard operating condition (SOC) and extended operating condition (EOC). There are 10 kinds of targets under SOC conditions, each of which contains omnidirectional SAR image data at 15° and 17° pitch angles. In the experiments, observation data at 17° were used for training, and the observation data at 15° pitch angle were used for testing. The optical image of the targets in the MSTAR SAR dataset collected under SOC conditions is shown in Figure 6. In the EOC-1 dataset, there are 4 kinds of ground targets, in which the targets with a side view angle of 17° are used for the training set and the targets with a side view angle of 30° are used for the test set.

The test set and training set are the same model targets in different pitch angles. In fact, this is one of the differences between high-resolution radar target recognition and image recognition. The purpose of this paper is to accurately recognize the target model through high-resolution radar data. In academia, there is only a difference in pitch angle between the test set and the training set, which is feasible and in line with reality [6, 21–23].

Because SAR images are extremely sensitive to changes in pitch angle, it is more difficult to identify the targets under EOC-1 conditions. The pitch angle difference between the SOC training set and the test set is 2°, while the difference under the EOC-1 is increased to 13°. This may lead to a big deviation of the same target in SAR images under the same posture, which increases the difficulty of recognition. Therefore, the experimental conclusions based on the SAR-EOC dataset are more valuable.

As shown in Table 4, the two vectors are two samples in the dataset HRRP-1, which reflects the scattering characteristics of the armored transport vehicle and the heavy transport vehicle, respectively.

The HRRP-1 dataset [22] is target electromagnetic scattering data obtained by high-frequency electromagnetic calculation software. HRRP provides the distribution of target scattering points along the distance and is an important structural feature of the target. HRRP has the characteristics of stable resolution, easy acquisition and realization, and short imaging period. The simulation database contains 4 kinds of ground vehicle targets: armored

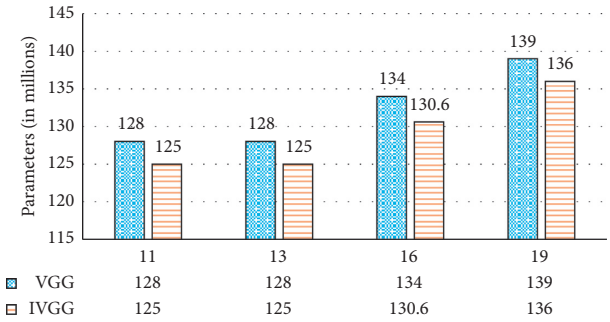


Figure 4: The number of parameters (in millions) of VGG networks and our methods.

Table 3: The number of parameters (in millions) of our networks with different classifiers.

Network	1FC	3FC
IVGG11	7.19	125
IVGG13	5.96	125
IVGG16	11.27	130.6
IVGG19	17.67	136

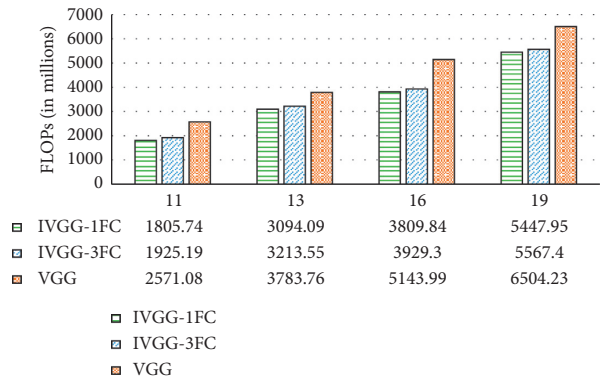


Figure 5: Comparison of floating points of operations (FLOPs).

transport vehicles, heavy transport vehicles, heavy trucks, and vans. Acting on the stepped frequency echo signal at the same observation angle of the target, Inverse fast Fourier transform (IFFT) is used to synthesize the HRRP. Since the electromagnetic simulation data are turntable-like data, it is not necessary to translate and align. In the experiment, the target electromagnetic scattering echo under the HH polarization mode is selected as the basic dataset. The targets with a pitch angle of 27° are used for training, and the targets with a pitch angle of 30° are used for test. Both the training set and the test set have 14400 samples, each of which is a 128×1 array with complex data type. The training set is the same as the test set except for the pitch angle. In addition, the HRRP data generated by inversion of the MSTAR SAR dataset are used as the second HRRP dataset (HRRP-2).

4.2. Preprocessing and Experimental Setup. For the MSTAR SAR images, each sample is resized to 128×128, and then, the center cut and random horizontal rotation are performed. After this preprocessing, the number of SAR images has been expanded by 3 times, which compensates for the shortage of SAR images and alleviates the overfitting problem of the network to some extent.

The phase profile of the complex high-resolution echo of the target can be divided into two parts: the initial phase that is sensitive to the distance and the remaining phase reflecting the scattering characteristics of the target. Therefore, like the amplitude profiles (real HRRP), phase profiles in the complex HRRP also represent a certain information of the scattering point distribution of the target, and it should be valuable in recognition. The complex HRRP contains all the phase information of the target scatter point subecho, including the initial phase and the remaining phase of the scatter point subecho. Therefore, although the complex HRRP has a sensitivity to the initial phase, which is not conducive to HRRP target recognition, it retains other phases information that is helpful for recognition [24]. The traditional RATR uses the amplitude image of HRRP and loses the phase information. Phase information is especially useful for target recognition, but most convolution network models cannot deal with complex data types. At present, the main processing method of complex HRRP is modulus operation, which can keep the amplitude information of range profile and get relatively high recognition accuracy.

Unlike images that can use superresolution method to improve recognition accuracy [25], HRRP is made up of one-dimensional data points, so we propose a new way to preprocess HRRP data. The real part and the imaginary part of each data are extracted and arranged in an orderly way, so that the length of each sample is expanded from 128 to 256. In this way, the differential phase information between the distance units in each HRRP sample can be preserved, and the amount of data in each sample can be expanded.

To compare the test results of different models, the experiments are carried out on the same platform and environment, as shown in Table 5.

Considering that the radar data are sparse, the activation function Rectified Linear Unit (ReLU) [26] will undoubtedly increase this sparseness and reduce the useful information of the target, which is unfavorable for recognition. So, we introduce another activation function, Hyperbolic Tangent function (Tanh). The resulting impact will be further analyzed in the experiments.

The learning rate attenuation method is also introduced in the training processing. As the number of iterations increases, the learning rate gradually decreases. This can ensure that the model does not fluctuate greatly in the later period of training and closer to the optimal solution.

We adjust the parameters according to the results of many experiments and get the final parameters. We use VGGNet pretrained by ImageNet in PyTorch to initialize the parameters of IVGG networks. In the training stage, the batch size of the training set is set to 16 and that of the test set is set to 32. For MSTAR SAR dataset recognition, the initial learning rate is set as 0.01, and 200 epochs are used for training. The learning rate decreases by 2 times since the first 50 epochs and then decreases by 2 times every 20 epochs. The average recognition accuracy of the last 100 epochs was

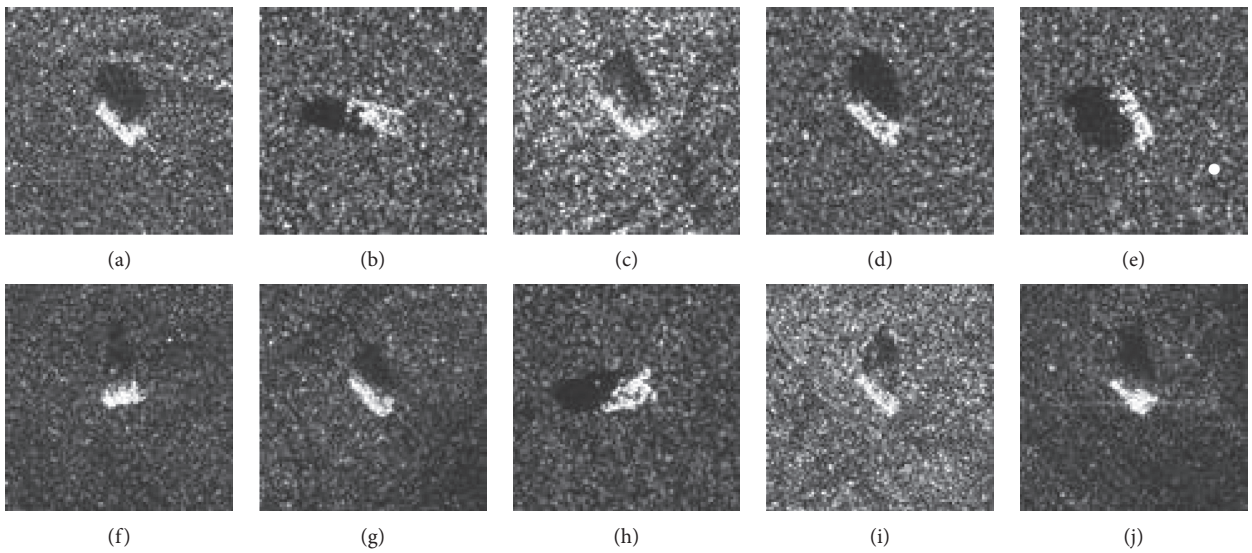


Figure 6: Images of the MSTAR SAR dataset under SOC.

Table 4: The samples of complex HRRP vector.

Sample 1 of HRRP	Sample 2 of HRRP
$5.947548139439314e-04-7.029982346588466e-04i$	$-0.001741710511154+0.005854695561424i$
$5.973508449729275e-04-7.301167648045039e-04i$	$-0.001602329272711+0.005996485005943i$
$5.998884995750467e-04-7.586149497061626e-04i$	$-0.001459788439038+0.006143776077643i$
$6.023640017197894e-04-7.885879483632503e-04i$	$-0.001313674253423+0.006297298858010i$
$6.047727981516010e-04-8.201413412111810e-04i$	$-0.001163535049426+0.006457875798999i$
...	...

Table 5: Experimental platform configuration.

Attribute	Configuration information
OS	Ubuntu 14.04.5 LTS
CPU	Intel (R) Xeon (R) CPU E5-2670 v3 @ 2.30 GHz
GPU	GeForce GTX TITAN X
CUDNN	CUDNN 6.0.21
CUDA	CUDA 8.0.61
Framework	PyTorch

calculated as the final results. For HRRP dataset recognition, the initial learning rate is set as 0.1 and 100 epochs are used for training. The learning rate decreases by 2 times since the first 50 epochs and then decreases by 2 times every 10 epochs. The average recognition accuracy of the last 10 epochs was calculated as the final results.

4.3. Recognition Results of the MSTAR SAR Dataset. The recognition accuracy on the MSTAR SAR dataset is shown in Table 6. On SAR-SOC, the results of IVGG networks and VGG networks are better than those of GoogLeNet, ResNet18, and DenseNet121. It can be seen from Table 6 that on the SAR-SOC, IVGG networks with both 1FC and 3FC have good recognition performance. It shows that our methods have better robustness. The recognition rates of IVGG networks are similar to those of VGGNets, but each of them reduces about 3 million parameters compared with the latter.

GoogLeNet achieves high recognition accuracies on SAR-SOC, but its recognition accuracies on SAR-EOC-1 are poor, which are only 90.62% and 90.19%. This shows that its generalization ability is not so ideal. Based on the horizontal comparison of the recognition accuracies of the activation functions, Tanh and ReLU in Table 6, we can see the performance of Tanh on SAR-EOC-1 is generally stronger, indicating that Tanh has a better effect on sparse data processing.

On SAR-SOC, IVGG16-3FC with "Tanh" achieves a maximum accuracy of 99.51%. On SAR-EOC-1, IVGG19-3FC achieves the highest accuracy of 99.27%, and IVGG13-3/1FC also achieves the accuracy of 99.22%. The classification on SAR-EOC is more difficult, and it requires that CNNs have higher performance. So, we especially focus on analyzing the experimental results on SAR-EOC.

The accuracy rate of IVGG13 on SAR-EOC is significantly higher than those of GoogLeNet, ResNet18, DenseNet121, VGG11, and VGG13. It is still 0.12% higher than VGG16 and VGG19. But the parameter number of IVGG13 is only 4.45% of VGG16 and 4.29% of VGG19, and the FLOPs are significantly lower than those of VGG16 and VGG19. Specifically, IVGG13-1FC saves 39.85% FLOPs than VGG16 and 52.43% than VGG19. The accuracy rate of IVGG13-1/3FC is only 0.05% lower than that of IVGG19-

Table 6: Accuracy rates (%) on the MSTAR SAR dataset.

Method	SAR-SOC		SAR-EOC-1	
	Tanh	ReLU	Tanh	ReLU
GoogLeNet	98.87	98.65	90.62	90.19
ResNet18	97.20	97.90	78.45	82.25
DenseNet121 ($k = 32$)	98.66	98.93	96.41	98.66
VGG11	99.31	99.32	98.61	97.60
VGG13	99.22	99.48	98.22	97.54
VGG16	99.14	99.50	99.10	96.75
VGG19	99.26	99.21	99.10	97.91
IVGG11-3FC	99.21	98.98	97.97	98.05
IVGG11-1FC	99.23	99.13	97.02	97.73
IVGG13-3FC	99.04	99.31	**99.22**	98.04
IVGG13-1FC	99.34	99.14	**99.22**	98.24
IVGG16-3FC	**99.51**	99.34	98.84	98.70
IVGG16-1FC	**99.42**	99.19	97.62	97.68
IVGG19-3FC	99.42	99.23	**99.27**	97.71
IVGG19-1FC	99.23	**99.37**	97.15	98.47

3FC, but the parameter of IVGG13-1FC is only 4.77% of that of IVGG19-3FC and the FLOPs of IVGG13-1/3FC are only about 56% of those of IVGG19-3FC.

The experiments show that the IVGG networks can work well on the SAR image public dataset and have good robustness and recognition performance. The important point is that IVGG uses a significantly shallower network to achieve better accuracy than other CNNs. It greatly improves the computational efficiency and can save great parameter space. In fact, IVGG13-1FC relies on relatively less parameters and FLPOs to achieve quite good results. In contrast, although IVGG16 and IVGG19 networks can slightly improve the recognition accuracy, they have paid a high price (increase in parameters and computational cost). We further compare the experimental results of the IVGG13-1FC network with other deep learning methods, proposed by Wang et al. [17], Pei et al. [18], and Chen et al. [19], as shown in Table 7. These literature studies use the same SAR image dataset with this paper. Wang et al. [17] proposed a method for SAR images target recognition by combining two-dimensional principal component analysis (2DPCA) and L2 regularization constraint stochastic configuration network (SCN). They applied the 2DPCA method to extract the features of SAR images. By combining 2DPCA and SCN (random learning model with a single hidden layer), the 2DPCA-SCN algorithm achieved good performance. Due to the limited original SAR images, it is difficult to effectively train the neural networks. To solve this problem, Pei et al. [18] proposed a multiview deep neural network. This deep neural network includes a parallel network topology with multiple inputs, which can learn the features of SAR images with different views layer by layer. Chen et al. [19] used all convolutional neural networks (A-CNNs) [27] to the target recognition of SAR images. Under the standard operating condition, the recognition accuracy on the SAR-SOC image dataset is remarkably high, but the recognition accuracy has declined under extended operating condition.

Although some methods such as A-CNN can achieve accuracy of 99.41% on the SAR-SOC, it is difficult to achieve satisfactory results on SAR-EOC-1 data which have a greater difference in pitch angles. The 2DPCA-SCN method achieves 98.49% accuracy on SAR-EOC-1, but only 95.80% on SAR-SOC. Other methods on the SAR-EOC-1 also achieve lower recognition accuracies than our methods. It can be found from Table 6 that IVGG networks achieve exceedingly high accuracies on both SAR-SOC and SAR-EOC-1 datasets. In particular, on the SAR-EOC-1 dataset, IVGG13 can achieve higher accuracy and more stable performance, which shows that our network has stronger generalization ability and better robustness.

IVGG13-1FC is also compared with traditional recognition methods such as KNN, SVM, and SRC [6, 23], and the results are shown in Table 8. The method proposed in reference [6] is a new classification approach of clustering multitask learning theory (I-CMTL), and SRC is a recognition method based on sparse representation-based classifier (SRC) proposed in 2016 [23]. From Table 8, we can see that our network is better than those of all the traditional recognition methods.

Table 8 shows that some traditional approaches are not so effective, such as KNN and SVM methods. Although many complex classifiers have been designed, they cannot fully utilize the potential correlation between multiple radar categories. On the other hand, large-scale and complete SAR datasets are difficult to collect, so the samples obtained are usually limited or unbalanced.

The classification algorithm approaches under the multitask framework have higher recognition accuracies, such as CMTL, MTRL, and I-MTRL. The multitask relational learning (MTRL) method proposed in [6] can autonomously learn the correlation between positive and negative tasks, and it can be easily extended to the nonlinear field. The MTRL is further improved by adding a projection regularization term to the objective function [7], which can independently learn multitask relationships and cluster information of different tasks and can also be easily extended to the nonlinear field. However, the Trace-norm Regularized multitask learning (TRACE), which is also under the multitask framework, has the lowest recognition accuracy because the TRACE method learns the linear prediction function and cannot accurately describe the nonlinear structure of SAR image, which also proves the importance of extending the multitask learning method to the nonlinear field.

The IVGG networks proposed in this paper can adaptively learn the nonlinear structure of SAR images and reduce the difficulty in redesigning the classifier when the SAR image conditions change. In contrast, the artificially designed feature extraction approach is complex, and sometimes, it can only be effective for certain fixed problems. Its generalization ability is not so ideal. Therefore, our networks enhance the feature extraction capability of sparse data.

Table 7: Accuracy rates (%) on the MSTAR SAR dataset of different CNNs.

Method	SAR-SOC	SAR-EOC-1
2DPCA-SCN [17]	95.80	98.49
2-view DCNNs [18]	97.81	93.29
3-view DCNNs [18]	98.17	94.34
4-view DCNNs [18]	98.52	94.61
A-CNN [19]	**99.41**	97.13
IVGG13-1FC	**99.34**	**99.22**

Table 8: Accuracy rates (%) of different methods on the SAR dataset.

Method	SAR-SOC	SAR-EOC-1
KNN [1]	92.71	91.42
SVM [1]	90.17	86.73
SRC [23]	89.76	—
TRACE [2]	75.04	67.42
RMTL [3]	92.09	92.03
CMTL [4]	93.91	94.72
MTRL [5]	95.84	95.46
I-CMTL [6]	97.34	98.24
IVGG13-1FC	**99.34**	**99.22**

4.4. Recognition Result of the HRRP Dataset. The recognition accuracy rates on the HRRP dataset are shown in Table 9

On the HRRP-1 dataset, the optimal recognition accuracies of GoogLeNet, ResNet18, and DenseNet121 are 98.7132%, 98.5234%, and 98.7299%, respectively, and the performance of the activation function Tanh is slightly better than that of ReLU. The best recognition results (accuracy > 99.05%) are all obtained by the activation function Tanh. The networks with recognition rate higher than 99.05% are VGG13 (Tanh), IVGG16-3FC (Tanh), and IVGG19-3FC (Tanh). Among them, the recognition rate of IVGG16-3FC (Tanh) is the highest, reaching 99.24%.

In the identification of the HRRP-1 dataset, the networks which are deeper have better recognition results. IVGG16 and IVGG19 can achieve better recognition effects.

The network with the best recognition accuracy on the HRRP-2 dataset is IVGG19-3FC (ReLU). The VGGNet and IVGG-3FC have higher recognition accuracies. The recognition results of IVGG networks and VGGNets have no obvious difference, among which IVGG19-3FC (ReLU) achieves the best recognition accuracy of 98.98%.

On the HRRP-1 dataset, our method is also compared with other methods such as SVM, Maximum Correlation Criterion-Template Matching Method (MCC-TMM) [28], Bayesian Compressive Sensing (BCS) [29], Joint Sparse Representation (JSR) [30], and a CNN method with SVM as its classifier [20], as shown in Table 10.

4.5. Comprehensive Analysis of Results. In conclusion, we find that DenseNet121 also has high performance in the SAR dataset (still slightly inferior to our method), but its recognition performance for HRRP is obviously reduced. In HRRP recognition, ResNet18 has a high performance (still slightly inferior to our method), but the performance of SAR image recognition is exceptionally low (only 80%). Different from the above two methods, our method has high recognition performance for SAR and HRRP signals, which means that the method in this paper is efficient and stable. VGG network achieves good performance for radar target recognition, but IVGG reduces the parameters significantly and improves the computation and recognition efficiency.

The performances of IVGG networks are better than those of VGGNets on the HRRP-1 dataset and SAR-EOC-1 dataset and better than those of other neural networks and traditional algorithms on all the experimental datasets.

In fact, the SAR image dataset used in this paper is a public dataset published by MSTAR, and the HRRP dataset also has been published in other papers. The radar is sensitive to the pitch angles, and the radar echo data of the same target at different pitch angles are quite different. This is also the difficulty of radar target recognition. On the SAR-EOC dataset, the difference of pitch angles between the test set and the training set is greater than that on SAR-SOC, and the recognition accuracy on the SAR-EOC test set is slightly lower than that on SAR-SOC.

In addition, we also found a problem in the experiment. When the network comes very deep, the recognition algorithm may be invalid. For example, when we use ResNet50, it will cause the method loss efficacy. The reason is that the data amount of each sample is small (especially HRRP is one-dimensional data), and the downsampling layers in the ResNet50 are too many for HRRP. This problem may also occur in SAR images. But overall, SAR images will be slightly better. Solving this problem has two points, one feasible method is to reduce the downsampling layers, but it will undoubtedly weaken the robustness of the network, which may lead to insufficient results and waste in computing costs. Another effective solution is to design shallow convolutional neural networks for radar target recognition, such as the IVGG networks proposed in this paper.

For target recognition in radar signals, the IVGG networks and VGGNets perform better than several convolutional neural networks recently proposed. The main reasons are as follows.

The noise of the optical image is usually additive noise, while the noise of the SAR image is mostly speckled multiplicative noise. HRRP data are a one-dimensional array, which is the vector sum of projection of the target scattering point echoes in the radar ray direction. Neither of them has obvious edge features and texture information like the traditional optical image. SAR image is sensitive to the azimuth of the target when it is imaged. When the azimuth is different, even for the same target, there are still excessively big differences in SAR images.

The data amount of HRRP and SAR images is less than that of traditional optical images. In this paper, only 256 data per HRRP target and $128 \times 128 = 16384$ data per SAR image are sent into the networks. However, a slightly larger optical image can often reach $256 \times 256 = 65536$ pixels. For this reason, the CNN models for radar target recognition cannot be too deep. Otherwise, they may fall into overfitting. So,

Table 9: Accuracy rates (%) on the HRRP dataset.

Method	HRRP-1		HRRP-2	
	Tanh	ReLU	Tanh	ReLU
GoogLeNet	98.71	97.95	98.48	97.85
ResNet18	98.52	98.02	98.48	98.20
DenseNet121	98.73	97.94	98.15	97.65
VGG11	98.32	98.18	98.56	97.51
VGG13	99.05	98.89	98.76	98.79
VGG16	98.75	98.55	98.94	98.88
VGG19	98.90	98.40	98.66	98.76
IVGG11-3FC	98.79	97.76	98.35	98.19
IVGG11-1FC	98.52	98.28	97.95	98.42
IVGG13-3FC	98.75	98.86	98.65	98.80
IVGG13-1FC	98.46	98.43	98.33	98.54
IVGG16-3FC	**99.24**	98.99	**98.90**	98.67
IVGG16-1FC	98.54	98.79	98.50	98.63
IVGG19-3FC	**99.06**	99.05	**98.84**	**98.98**
IVGG19-1FC	98.35	98.84	98.02	98.11

Table 10: Accuracy rates (%) of different methods on the HRRP-1 dataset.

Method	Accuracy rate (%)
SVCA + SVM [28]	94.24
MCC-TMM [28]	92.81
BCS [29]	92.76
JSR [30]	91.49
CNN + SVM [20]	96.45
IVGG16-3FC	**99.24**

compared with ResNet and DenseNet, IVGG networks and VGGNets with fewer network layers have better recognition ability.

In the experiment, the activation function Tanh has excellent performance on the SAR-EOC-1 and HRRP datasets. The radar data itself have sparsity, which is enhanced by the activation function ReLU, while too sparse data will weaken the ability of the convolution layer to extract target features. Activation function Tanh has better nonlinearity and works better when the feature difference is obvious.

5. Conclusion

In this paper, we propose the IVGG networks and use them for target recognition on HRRP data and SAR images. The first improvement in this paper is to propose the IVGG networks. Then we simplify the fully connected layers which can significantly reduce parameters. Experiments show that our methods have the best recognition effect. At the same time, with the improvement of the networks, there are fewer parameters in the networks, which can improve the processing efficiency of target recognition and make the method more suitable for the real-time requirements.

In addition, we also find that for radar target recognition, Tanh's performance is generally better than that of ReLU, which is different from image recognition.

Data Availability

All datasets in this article are public datasets and can be found on public websites.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Defense Pre-Research Foundation of China under Grant 9140A01060314KG01018, National Natural Science Foundation of China under Grant 61471370, Equipment Exploration Research Project of China under Grant 71314092, Scientific Research Fund of Hunan Provincial Education Department under Grant 17C0043, and Hunan Provincial Natural Science Fund under Grant 2019JJ80105.

References

[1] Q. Zhao and J. C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 2, pp. 643–654, 2001.

[2] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statistics and Computing*, vol. 20, no. 2, pp. 231–252, 2010.

[3] T. Evgeniou and M. Pontil, "Regularized multi-task learning, knowledge discovery and data mining," in *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 109–117, Seattle, WC, USA, 2004.

[4] J. Zhou, J. Chen, J. Ye et al., "Clustered multi-task learning via alternating structure optimization," *Neural Information Processing Systems*, vol. 2011, pp. 702–710, 2011.

[5] Y. Zhang and D.-Y. Yeung, "A regularization approach to learning task relationships in multitask learning," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 3, pp. 1–31, 2014.

[6] L. Cong, B. Weimin, X. Luping et al., "Clustered multi-task learning for automatic radar target recognition," *Sensors*, vol. 17, no. 10, p. 2218, 2017.

[7] Z. He, J. Lu, and G. Kuang, "A fast SAR target recognition approach using PCA features," in *Proceedings of the Fourth International Conference on Image and Graphics (ICIG 2007)*, IEEE, Chengdu, China, pp. 580–585, August 2007.

[8] D. Meng and L. Sun, "Some new trends of deep learning research," *Chinese Journal of Electronics*, vol. 28, no. 6, pp. 1087–1090, 2019.

[9] W. Wang, C. Tang, X. Wang et al., "Image object recognition via deep feature-based adaptive joint sparse representation," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8258275, 9 pages, 2019.

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

[11] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, 2015.

[12] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[13] G. Huang, Z. Liu, V. D. M. Laurens et al., "Densely connected convolutional networks," in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, pp. 2261–2269, Honolulu, HI, USA, 2017.

[14] W. Wang, Y. Li, T. Zou et al., "A novel image classification approach via dense-MobileNet models," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 18 pages, 2020.

[15] W. Wang, Y. Yang, X. Wang et al., "The development of convolution neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, Article ID 040901, 2019.

[16] D. Ciresan, U. Meier, J. Masci et al., "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 30, pp. 1237–1242, Barcelona, Spain, 2011.

[17] Y. Wang, Y. Zhang, L. Yang et al., "Target recognition method based on 2DPCA-SCN regularization for SAR images," *Journal of Signal Processing*, vol. 35, no. 5, pp. 802–808, 2019, in Chinese.

[18] J. Pei, Y. Huang, W. Huo, Y. Zhang, J. Yang, and T.-S. Yeo, "SAR automatic target recognition based on multiview deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2196–2210, 2018.

[19] Y. Cheng, L. Yu, and X. Xie, "SAR image target classification based on all congvolutional neural network," *Radar Science and Technology*, vol. 16, no. 3, pp. 242–248, 2018, in Chinese.

[20] S. He, R. Zhang, J. Ou et al., "High resolution radar target recognition based on convolution neural network," *Journal of Hunan University (Natural Sciences)*, vol. 46, no. 8, pp. 141–148, 2019, in Chinese.

[21] J. C. Mossing and T. D. Ross, "An evaluation of SAR ATR algorithm performance sensitivity to MSTAR extended operating conditions," in *Proceedings of SPIE-The International Society for Optical Engineering*, pp. 554–565, Moscow, Russia, 1998.

[22] S. Liu, R. Zhan, Q. Zhai et al., "Multi-view polarization HRRP target recognition based on joint sparsity," *Journal of Electronics and Information Technology*, vol. 38, no. 7, pp. 1724–1730, 2016.

[23] H. Song, K. Ji, Y. Zhang, X. Xing, and H. Zou, "Sparse representation-based SAR image target classification on the 10-class MSTAR data set," *Applied Sciences*, vol. 6, no. 1, p. 26, 2016.

[24] L. Du, H. Liu, Z. Bao et al., "Radar automatic target recognition based on complex high range resolution profile feature extraction," *Science in China Series F-Information Sciences*, vol. 39, no. 7, pp. 731–741, 2009, in Chinese.

[25] W. Wang, Y. Jiang, Y. Luo et al., "An advanced deep residual dense network (DRDN) approach for image super-resolution," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592–1601, 2019.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the International Conference on Machine Learning*, pp. 807–814, Haifa, Israel, 2010.

[27] J. T. Springenberg, A. Dosovitskiy, T. Brox et al., "Striving for simplicity: the all convolutional net," 2014, https://arxiv.org/abs/1412.6806.

[28] A. Zyweck and R. E. Bogner, "Radar target classification of commercial aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 2, pp. 598–606, 1996.

[29] S. Ji, Y. Xue, L. Carin et al., "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[30] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Joint sparse representation for robust multimodal biometrics recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 113–126, 2014.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8842221, 10 pages
https://doi.org/10.1155/2020/8842221

Research Article

Common Laws Driving the Success in Show Business

Chong Wu,[1] **Zhenan Feng**,[2] **Jiangbin Zheng**,[3] **and Houwang Zhang**[2]

[1]*Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong*
[2]*School of Automation, China University of Geosciences, Wuhan 430074, China*
[3]*School of Informatics, Xiamen University, Xiamen 361005, China*

Correspondence should be addressed to Chong Wu; chongwu2-c@my.cityu.edu.hk

Received 16 May 2020; Revised 10 June 2020; Accepted 23 June 2020; Published 10 July 2020

Academic Editor: Nian Zhang

In this paper, we want to find out whether gender bias will affect the success and whether there are some common laws driving the success in show business. We design an experiment, set the gender and productivity of an actor or actress in a certain period as the independent variables, and introduce deep learning techniques to do the prediction of success, extract the latent features, and understand the data we use. Three models have been trained: the first one is trained by the data of an actor, the second one is trained by the data of an actress, and the third one is trained by the mixed data. Three benchmark models are constructed with the same conditions. The experiment results show that our models are more general and accurate than benchmarks. An interesting finding is that the models trained by the data of an actor/actress only achieve similar performance on the data of another gender without performance loss. It shows that the gender bias is weakly related to success. Through the visualization of the feature maps in the embedding space, we see that prediction models have learned some common laws although they are trained by different data. Using the above findings, a more general and accurate model to predict the success in show business can be built.

1. Introduction

"Do I need to change a job?" is one of the major concerns to most actors and actresses since the show business is really competitive [1]. Matthew effect [2] or the so-called "rich-get-richer" phenomenon is proved to exist in the show business which demonstrates the scarcity of the resources [1]. Luck is proved to be a key element in driving the success [3]. It is well known that the effect of rich-get-richer is quite arbitrary and unpredictable [4]. Hence, most actors and actresses will meet a problem of avoiding the famine and building a sustainable career in acting [1]. Some studies have found that boosting productivity is a key metric to evaluate the success of an actor or actress, and it can be more of a network effect [5, 6] than a consequence of acting skills; in other words, success is not highly related to the acting skills [1]. And, some studies show the relationship between the dynamic collaboration network and success [7]: success is a collective phenomenon [8]. Startup network is proved to have predictive power in show business [9]. And, future success can be predicted by monitoring the behavior of a small set of individuals [10]. To study the law of success, a great deal of work has been done [11–19].

Recently, a study shows that the success in show business is predictable and uses a heuristic threshold-based binary classifier to achieve an accuracy up to 85% [1]. In their study, they find a strong gender bias in the waiting time statistics, the location of annus mirabilis, and the career length distribution of these data. However, we have some questions here: Whether gender bias is one of the key elements driving the success? Can we find some common laws driving the success in show business? Since we want to build a general prediction model, the common laws which determine the growth and the shape of the series are more important than the differences.

To solve our questions, we design this study. The data we use are collected from the International Movie Database (IMDb), http://www.imdb.com in [1]. It consists of millions of profile sequences of actors and actresses from the birth of the film in 1888 up to the present day [1]. Each sequence records the yearly time series of credited jobs over the entire working life of the actor or actress [1]. We just consider the

number of credited jobs regardless of the impact of the work, the screen time, and so on, which is the same as in [1]. The original feature space is a non-Euclidean space. We must to do the representation learning to map these features to a Euclidean space. To do this, we construct a deep model which consists of an encoder and a classifier. Since gender is an independent variable in our experiment, we train three models: (1) MAO, (2) MAE, and (3) MM. They all have the same structure but are trained by different datasets (MAO is trained by the data of an actor, MAE is trained by the data of an actress, and MM is trained by the mixed data). Our problem can be reconstructed like follows: (1) if MAO can achieve nondegradation performance on the data of an actress like MAE and MAE can achieve nondegradation performance on the data of an actor like MAO, then it can be proved that there are common features in the series which are unrelated to the gender. (2) If MM can achieve similar and nonsuperior performance against MAO and MAE, then these features which have gender bias are not dominative features in this prediction problem; that is to say, gender bias may cause some differences into the resource allocation, but it is weakly related to success. The contributions of this paper can be concluded as follows:

(1) We found that there are some common laws/features driving the success in show business by extracting and understanding the data.

(2) Using these common features, a more general prediction model with an accuracy up to 90% can be built.

(3) Our experiment shows that gender bias is weakly related to success despite a recent study which shows that it affects strongly the waiting time statistics, the location of annus mirabilis, the career length distribution, etc.

2. Materials and Methods

2.1. Data. The data we use consist of the careers of 1,512,472 actors and 896,029 actresses from 1888 up to 2016 and are collected from the International Movie Database (IMDb) http://www.imdb.com. Each career is viewed as a profile sequence: the yearly time series of acting jobs in films or TV series over the entire working life of the actor or actress [1]. We refer to [1] and relax their selection constraint to select the sequences of actors and actresses with working lives $L \geq 5$ years, and the number of credited jobs in the annus mirabilis (AM) is ≥ 5. The sequences obtained by some more relaxed cutoffs are too short to be analyzed, and they are considered as the outliers and not included in the experiment. Then, the subset we use consists of 37896 (2.51%) sequences of actors and 22025 (2.46%) sequences of actresses which is larger than the data used in the prediction model in [1]. We divide this subset into several groups for experiment: (1) Group 1: the data of an actor with $\mathrm{AM} \geq 5$ and $L \geq 20$, including 21994 sequences; (2) Group 2: the data of an actress with $\mathrm{AM} \geq 5$ and $L \geq 20$, including 9034 sequences; (3) Group 3: the data of an actor with $\mathrm{AM} \geq 5 \leq L < 20$, including 15902 sequences; (4) Group 4: the data of an actress with $\mathrm{AM} \geq 5.5 \leq L < 20$, including 12991 sequences. Group 1 and Group 2 can be considered as some very successful actors which are used to train the prediction model mainly. Group 3 and Group 4 can be considered as some actors who are not very successful, and they might need a prediction model more than previous groups, and these data will be used to test the prediction model.

2.2. Data Preprocessing. To do an early prediction, we need to do some preprocessing on the data before training the model. At first, we refer to [1] to truncate each sequence into several subsequences or called subcareer series. For each sequence, we randomly sample several subsequences with a sampling rate n. The subsequences which are sampled before the annus mirabilis are regarded as class 1. The subsequences which are sampled after the annus mirabilis are regarded as class 2. Hence, it is a binary classification problem. The aim of this sampling is to get some samples of class 1 since we only have the entire working life of the actor or actress. An example of the sampling process with a sampling rate $r = 4$ is shown in Figure 1. NatComm19 uses the following function [1] to transfer these subsequences to scalars for the training:

$$D(\overline{w}_T) = -\sum_{y=1}^{T-1} \min\left(0, \overline{w}_{y+1} - \overline{w}_y\right), \tag{1}$$

where $\overline{w}_T$ is the number of credited jobs at year T and T is the length of the subsequence.

The above transformation will lose some information like the increasing or decreasing trend. In this paper, we revise equation (1) as follows to get a new sequence and not a scalar which will protect these information:

$$D(\overline{w}_T) = -\sum_{y=1}^{k-1} \min\left(0, \overline{w}_{y+1} - \overline{w}_y\right). \tag{2}$$

Then, we use the new sequence D to train the model.

Since gender is an independent variable, we construct three prediction models which will be trained by different subsets of the whole data. The details of separation of training data and test data for each model are shown in Table 1.

2.3. Prediction Model. Recurrent neural network (RNN) or long short-term memory (LSTM) [20, 21] is powerful to solve the time series prediction problem with sequential data. Compared to the standard feedforward neural network, RNN is a kind of neural networks which is as the feedback connections (memory), as shown in Figure 2. It can process not only single data points, but also the entire sequences of data. For example, LSTM is applied in some tasks such as speech recognition [22], sign language translation [23], object cosegmentation [24, 25], and airport passenger management [26]. Hence, here, we use RNN with LSTM units to build an end-to-end prediction model, where the LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. Figure 3 shows the structure of our

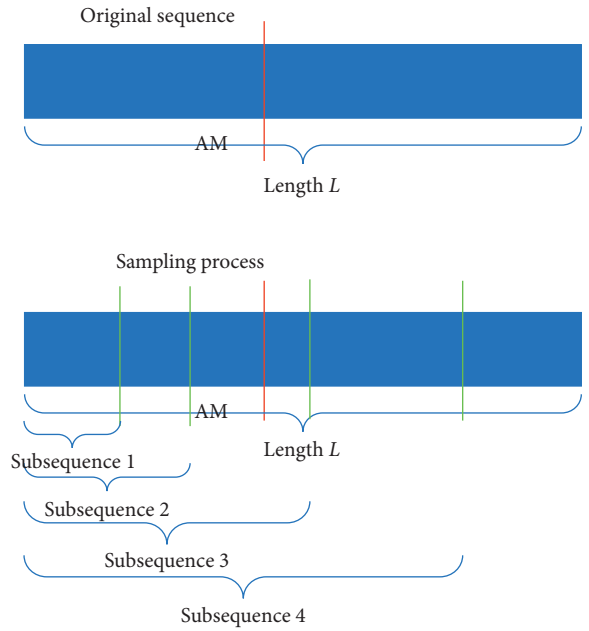


Figure 1: The process of subsequence generation.

model. Sequentially, our model can be divided into two parts: (1) encoder; (2) binary classifier. The encoder consists of an LSTM layer with 30 hidden units and outputs at the last time step. And, the classifier consists of a fully connected layer, a softmax layer, and a classification layer with the cross entropy as the loss function. Our model is trained in a supervised fashion, on a set of training sequences, using an optimization algorithm, gradient descent. Since sequences have different lengths as shown in Figure 4, the feature space of these sequences is a non-Euclidean space. It is difficult to train a classifier in this feature space. Hence, each input sequence will be embedded by the encoder to a Euclidean space using the following transformation:

$$f: D \longrightarrow H, \tag{3}$$

where H is an n-dim sequence. Through the encoder, the dimension of the feature is also reduced. Then, the following loss function is minimized to get the optimized parameters:

$$L(C, \widehat{C}) = -C\log(\widehat{C}) - (1 - C)\log(1 - \widehat{C}), \tag{4}$$

where C is the real label and $\widehat{C}$ is the label predicted by the classifier.

In the process of forward propagation, LSTM does not simply compute a weighted sum of the input signal. It applies a nonlinear function. For each j-th LSTM unit, it maintains a memory c_i^j at time j and an output gate weight o_i^j. Then, the output h_i^j is

$$h_i^j = o_i^j \tanh\left(c_i^j\right). \tag{5}$$

The memory cell c_t^j is updated by partially forgetting the existing memory and adding a new memory content $c_t^{j'}$:

$$c_t^j = f_t^j c_{t-1}^j + p_t^j c_t^{j'}, \tag{6}$$

where f_t^j is the weight of the forget gate and p_t^j is the weight of the input gate.

The details of each layer's configuration are shown in Table 2. The training settings for the prediction model: max epoch is set to 15, size of the minibatch is set to 100, optimizer is Adam, and gradient threshold is set to 1. More complex models like the models with deep layers and the models with complex structures (biLSTM) have also been tested, but there is no obvious performance improvement. That is to say, these are all fairly "off the shelf" classifiers. Since simpler is better, we just use the simplest model to show the results.

3. Results

Table 3–5 show the comparison between our model and a recent study NatComm19 [1] on the test data. MM_ours denotes the prediction model trained by the mixed data of an actor and actress, MAO_ours denotes the prediction model trained by the data of an actor only, and MAE_ours denotes the prediction model trained by the data of an actress only. MM_NatComm19 denotes the model of NatComm19 [1] trained by the mixed data of an actor and actress, and the learned threshold $d = 6.1523$; MAO_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actor only, and the learned threshold $d = 6.9580$; and MAE_-NatComm19 denotes the model of NatComm19 [1] trained by the data of an actress only, and the learned threshold $d = 5.6640$. All models are trained on the training data with a cutoff value ($AM \geq 5$, $L \geq 20$). We can see that our models outperform NatComm19 in terms of all quantity metrics in all subsets of the test data. Our models are more general than NatComm19 and can still maintain the performance on the

Table 1: The details of training data and test data for each model.

Training data including validation data	Test data
Model 1: MAO_ours	
70% data of an actor (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 6$; the total number of subsequences in the training set: $6^{*}0.7^{*}21994 = 92374$	30% data of an actor (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 6$; 100% data of an actor (AM ≥ 5, $5 = < L < 20$), the sampling rate of subsequence generation: $n = 5$; 100% data of an actress (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 12$; 100% data of an actress (AM ≥ 5, $5 = < L < 20$), the sampling rate of subsequence generation: $n = 5$; the total number of subsequences in the test set: $6 * 0.3 * 21994 + 5 * 1 * 15902 + 12 * 1 * 9034 + 5 * 1 * 12991 = 292462$
Model 2: MAE_ours	
70% data of an actress (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 12$; the total number of subsequences in the training set: $12^{*}0.7^{*}9034 = 75885$	30% data of an actress (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 12$; 100% data of an actress (AM ≥ 5, $5 = < L < 20$), the sampling rate of subsequence generation: $n = 5$; 100% data of an actor (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 6$; 100% data of an actor (AM ≥ 5, $5 = < L < 20$), the sampling rate of subsequence generation: $n = 5$; the total number of subsequences in the test set: $12 * 0.3 * 9034 + 5 * 1 * 12991 + 6 * 1 * 21994 + 5 * 1 * 15902 = 308951$
Model 3: MM_ours	
70% mixed data of an actor and actress (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 3$ for the data of an actor in the mixed data, $n = 6$ for the data of an actress in the mixed data; the total number of subsequences in the training set: $3^{*}0.7^{*}21994 + 6^{*}0.7^{*}9034 = 46187 + 37942 = 84129$	30% mixed data of an actor and actress (AM ≥ 5, $L \geq 20$), the sampling rate of subsequence generation: $n = 3$ for the data of an actor in the mixed data, $n = 6$ for the data of an actress in the mixed data; 100% data of an actress (AM ≥ 5, $5 = < L < 20$), the sampling rate of subsequence generation: $n = 5$; 100% data of an actor (AM ≥ 5, $5 = < L < 20$), the sampling rate of subsequence generation: $n = 5$; the total number of subsequences in the test set: $3 * 0.3 * 21994 + 6 * 0.3 * 9034 + 5 * 1 * 12991 + 5 * 1 * 15902 = 180520$

The validation data are included in the training data. *Note*. MM_ours denotes the prediction model trained by the mixed data of an actor and actress, MAO_ours denotes the prediction model trained by the data of an actor only, and MAE_ours denotes the prediction model trained by the data of an actress only.

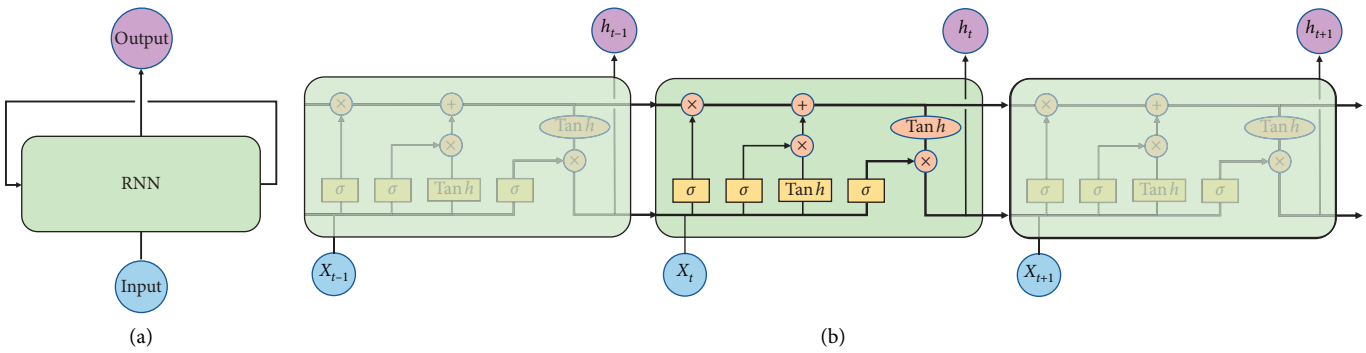


Figure 2: The structure of RNN and details of the LSTM unit.

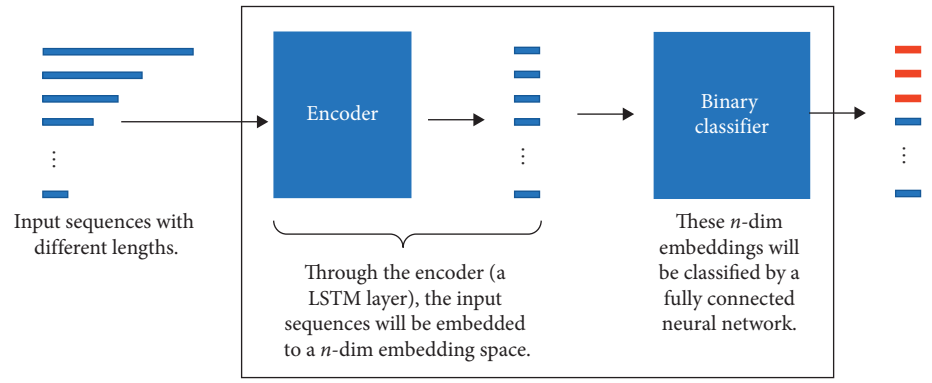


Figure 3: The workflow of our model. It has an end-to-end structure and can be divided into two parts: (1) encoder; (2) binary classifier. The encoder of our model is a single LSTM layer which is used to embed different sequences to an n-dim embedding space. The binary classifier is a fully connected neural network.

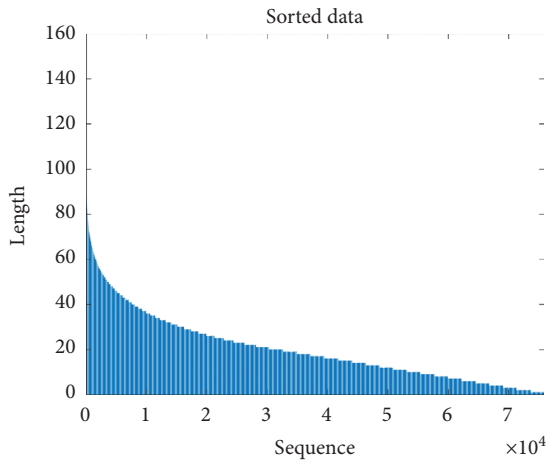


Figure 4: Sequences sorted by the sequence length.

new data ($AM \geq 5$, $5 \geq L < 20$ and $AM \geq 10$, $5 \geq L < 20$ and $AM \geq 15$, $5 \geq L < 20$), whereas the performance of three models of NatComm19 degrades to near the baseline. The details of the baseline model can be found in [1]. There is almost no difference between the performance of our three models. And, interestingly, the difference between the performance of the three models of NatComm19 can also be ignored.

Table 2: The details of each layer's configuration in our model.

Layer's name	Input size	Output size	No. of hidden units
Sequence input	1	1	
LSTM	1	—	
Fully connected layer	30	2	30
Softmax layer	2	2	
Classification layer	2	2	

4. Discussion

Two MAE models (MAE_ours and MAE_NatComm19) can achieve similar results compared to two MAO models (MAO_ours and MAO_NatComm19) on the test data of an actor. Similarly, two MAO models (MAO_ours and MAO_NatComm19) can also achieve similar results compared to two MAE models (MAE_oursandMAE_-NatComm19) on the test data of an actress. The case of MAE_ours and MAO_ours shows that our models can

Table 3: Performance comparison of our methods and a recent study NatComm19 [1] in the prediction of the AM on the subset (AM ≥ 5) of the test data.

	Actor $L \geq 20$, AM ≥ 5	Actress $L \geq 20$, AM ≥ 5	Actor $5 = < L < 20$, AM ≥ 5	Actress $5 = < L < 20$, AM ≥ 5
$C1 : C2$	0.8074	0.6136	0.7888	0.5655
Baseline accuracy	0.6702	0.7221	0.7034	0.7487
		MM_ours		
$F1$ score	0.9102	0.9262	0.8891	0.9173
Precision	0.8866	0.9079	0.8570	0.9037
Recall	0.9350	0.9452	0.9237	0.9313
Accuracy	0.8978	0.9067	0.8702	0.8917
		MAO_ours		
$F1$ score	0.9082	0.9254	0.9045	0.9272
Precision	0.9010	0.9267	0.8845	0.9203
Recall	0.9156	0.9241	0.9254	0.9341
Accuracy	0.8992	0.9077	0.8897	0.9048
		MAE_ours		
$F1$ score	0.9104	0.9268	0.9021	0.9265
Precision	0.8958	0.9203	0.8537	0.8966
Recall	0.9255	0.9334	0.9564	0.9584
Accuracy	0.8992	0.9087	0.8828	0.9020
		NatComm19MM		
$F1$ score	0.7956	0.7878	0.7442	0.7436
Precision	0.8930	0.8346	0.6092	0.6074
Recall	0.7174	0.7459	0.9562	0.9585
Accuracy	0.8338	0.8453	0.7100	0.7099
		NatComm19MAO		
$F1$ score	0.7942	0.7872	0.7457	0.7438
Precision	0.8902	0.8347	0.6103	0.6075
Recall	0.7169	0.7448	0.9582	0.9588
Accuracy	0.8332	0.8464	0.7116	0.7111
		NatComm19MAE		
$F1$ score	0.7707	0.7770	0.7766	0.7409
Precision	0.9176	0.8803	0.6630	0.6057
Recall	0.6643	0.6954	0.9371	0.9540
Accuracy	0.8238	0.8474	0.7622	0.7591

MM_ours denotes the prediction model trained by the mixed data of an actor and actress; MAO_ours denotes the prediction model trained by the data of an actor only; MAE_ours denotes the prediction model trained by the data of an actress only; MM_NatComm19 denotes the model of NatComm19 [1] trained by the mixed data of an actor and actress, and the learned threshold $d = 6.1523$; MAO_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actor, and the learned threshold $d = 6.9580$; MAE_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actress, and the learned threshold $d = 5.6640$.

Table 4: Performance comparison of our methods and a recent study NatComm19 [1] in the prediction of the AM on the subset (AM ≥ 10) of the test data.

	Actor $L \geq 20$, AM ≥ 10	Actress $L \geq 20$, AM ≥ 10	Actor $5 = < L < 20$, AM ≥ 10	Actress $5 = < L < 20$, AM ≥ 10
$C1 : C2$	0.6481	0.4169	0.6053	0.3348
Baseline accuracy	0.7275	0.7968	0.7668	0.8173
		MM_ours		
$F1$ score	0.9409	0.9591	0.9202	0.9530
Precision	0.9355	0.9612	0.9418	0.9780
Recall	0.9463	0.9571	0.8995	0.9293
Accuracy	0.9279	0.9422	0.9024	0.9313
		MAO_ours		
$F1$ score	0.9389	0.9557	0.9276	0.9563
Precision	0.9551	0.9729	0.9538	0.9836
Recall	0.9232	0.9391	0.9029	0.9306
Accuracy	0.9270	0.9387	0.9118	0.9359
		MAE_ours		
$F1$ score	0.9396	0.9559	0.9377	0.9643
Precision	0.9414	0.9664	0.9338	0.9761

Table 4: Continued.

	Actor $L \geq 20$, AM ≥ 10	Actress $L \geq 20$, AM ≥ 10	Actor $5 =< L < 20$, AM ≥ 10	Actress $5 =< L < 20$, AM ≥ 10
Recall	0.9378	0.9457	0.9415	0.9528
Accuracy	0.9264	0.9386	0.9217	0.9467
		NatComm19MM		
*F*1 score	0.7688	0.8008	0.8114	0.7607
Precision	0.9299	0.8879	0.7321	0.6371
Recall	0.6552	0.7292	0.9101	0.9439
Accuracy	0.8460	0.8916	0.8367	0.8478
		NatComm19MAO		
*F*1 score	0.7681	0.7989	0.8085	0.7559
Precision	0.9280	0.8841	0.7250	0.6297
Recall	0.6552	0.7287	0.9137	0.9453
Accuracy	0.8453	0.8909	0.8371	0.8449
		NatComm19MAE		
*F*1 score	0.7377	0.7790	0.8127	0.7616
Precision	0.9373	0.8956	0.7518	0.6431
Recall	0.6082	0.6892	0.8843	0.9337
Accuracy	0.8330	0.8823	0.8429	0.8502

MM_ours denotes the prediction model trained by the mixed data of an actor and actress; MAO_ours denotes the prediction model trained by the data of an actor only; MAE_ours denotes the prediction model trained by the data of an actress only; MM_NatComm19 denotes the model of NatComm19 [1] trained by the mixed data of an actor and actress, and the learned threshold $d = 6.1523$; MAO_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actor, and the learned threshold $d = 6.9580$; MAE_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actress, and the learned threshold $d = 5.6640$.

Table 5: Performance comparison of our methods and a recent study NatComm19 [1] in the prediction of the AM on the subset (AM ≥ 15) of the test data.

	Actor $L \geq 20$, AM ≥ 15	Actress $L \geq 20$, AM ≥ 15	Actor $5 =< L < 20$, AM ≥ 15	Actress $5 =< L < 20$, AM ≥ 15
$C1 : C2$	0.5271	0.3253	0.6292	0.3429
Baseline accuracy	0.7683	0.8439	0.7940	0.8467
		MM_ours		
*F*1 score	0.9563	0.9725	0.9236	0.9583
Precision	0.9600	0.9756	0.9548	0.9883
Recall	0.9527	0.9694	0.8945	0.9301
Accuracy	0.9434	0.9584	0.9021	0.9358
		MAO_ours		
*F*1 score	0.9533	0.9697	0.9336	0.9618
Precision	0.9750	0.9832	0.9692	0.9934
Recall	0.9326	0.9566	0.9005	0.9322
Accuracy	0.9401	0.9548	0.9159	0.9414
		MAE_ours		
*F*1 score	0.9560	0.9710	0.9340	0.9638
Precision	0.9632	0.9794	0.9306	0.9758
Recall	0.9489	0.9627	0.9375	0.9520
Accuracy	0.9425	0.9562	0.9179	0.9458
		NatComm19MM		
*F*1 score	0.7647	0.7990	0.8161	0.7425
Precision	0.9303	0.8555	0.7581	0.6161
Recall	0.6492	0.7495	0.8837	0.9340
Accuracy	0.8600	0.9099	0.8593	0.8614
		NatComm19MAO		
*F*1 score	0.7608	0.8043	0.7978	0.7588
Precision	0.9230	0.8660	0.7282	0.6345
Recall	0.6470	0.7508	0.8822	0.9437
Accuracy	0.8610	0.9096	0.8491	0.8639

Table 5: Continued.

	Actor $L \geq 20$, AM ≥ 15	Actress $L \geq 20$, AM ≥ 15	Actor $5 = < L < 20$, AM ≥ 15	Actress $5 = < L < 20$, AM ≥ 15
		NatComm19MAE		
$F1$ score	0.7361	0.7883	0.7954	0.7552
Precision	0.9268	0.8586	0.7455	0.6391
Recall	0.6105	0.7286	0.8525	0.9230
Accuracy	0.8530	0.9059	0.8489	0.8671

MM_ours denotes the prediction model trained by the mixed data of an actor and actress; MAO_ours denotes the prediction model trained by the data of an actor only; MAE_ours denotes the prediction model trained by the data of an actress only; MM_NatComm19 denotes the model of NatComm19 [1] trained by the mixed data of an actor and actress, and the learned threshold $d = 6.1523$; MAO_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actor, and the learned threshold $d = 6.9580$; MAE_NatComm19 denotes the model of NatComm19 [1] trained by the data of an actress, and the learned threshold $d = 5.6640$.

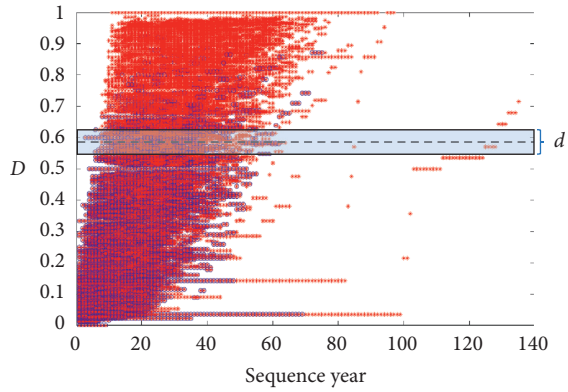


Figure 5: The workflow of the model in NatComm19 [1]. d is a scalar threshold which is learnable. The target of this model is to get an optimal d to separate two classes in the original feature space.

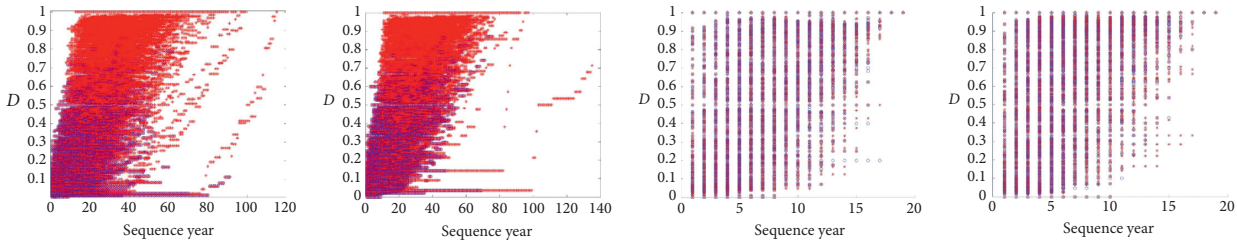


Figure 6: Feature maps of the original feature space. *Note*. There are a few outliers (sequences with a length over 100). It is caused by a few films that in some sense exist but have not been released. Since they are so rare and are the correct data, they are also considered as in [1]: (a) actor, AM ≥ 5, $L \geq 20$; (b) actress, AM ≥ 5, $L \geq 20$; (c) actor, AM $\geq 5.5 \leq L < 20$; (d) actress, AM $\geq 5.5 \leq L < 20$.

learn some common features that are used to classify. Since the model of NatComm19 uses a learnable threshold to classify the original feature space as shown in Figure 5, the case of MAE_NatComm19 and MAO_NatComm19 shows that the distribution and the shape of the original feature space of the data of an actor and the data of an actress are similar just as shown in Figure 6. MM_ours achieves similar and nonsuperior results compared to MAE_ours and MAO_ours, and MM_NatComm19 also achieves similar and nonsuperior results compared to MAE_-NatComm19 and MAO_NatComm19. It shows that these features which have gender bias are not dominative features in this prediction problem; that is to say, gender bias may cause some differences in some aspects like resource allocation, but it is weakly related to success. To further validate our conclusion, we visualize the embedding space in Figure 7. It seems that three models learn some different features. But, it was caused by the randomness of the neural network, and the order of these features has no meaning because it is like the eigen decomposition. From the weight of each embedding feature which is obtained in the fully connected layer, we can see that most of these embedding features are unimportant. And interestingly, all three models have only one dominative feature. The floating range of the corresponding feature in three models is also similar $[-1, s]$, where s is a positive scalar. We can believe that they have learned a similar feature that is used to classify.

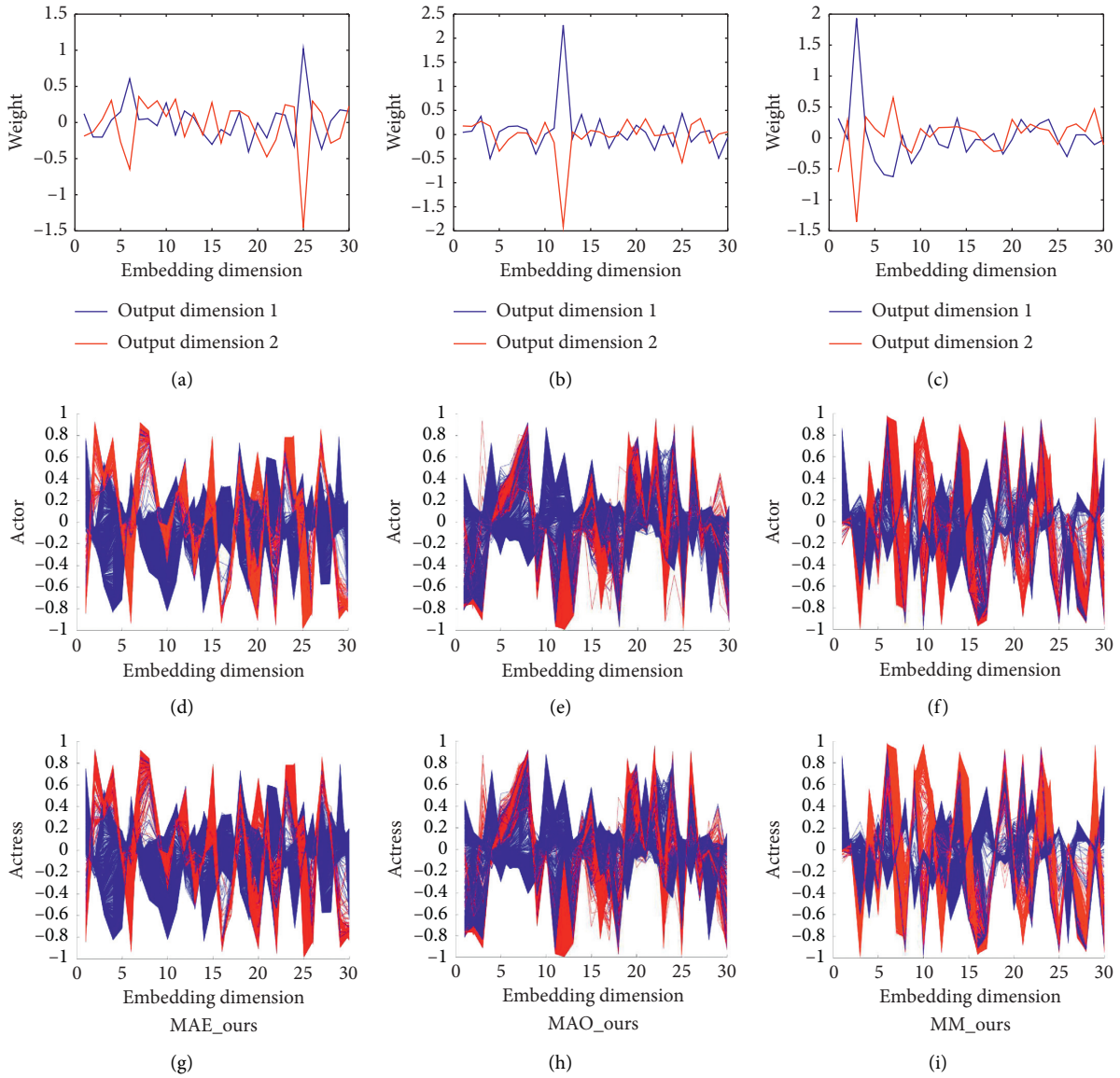


Figure 7: Feature maps of the testing data of an actor and actress in the embedding space obtained by different models. Blue line denotes class 1, and red line denotes class 2. It can be seen that the curves of different datasets show the same distribution and shape in the same embedding space. And, the boundary between two classes is clearer than the original feature space. Although it seems that the embedding spaces of different models are different, they are actually equivalent because they are different approximations of the global optimum obtained by the neural network. And, the curves of each feature's weight show that there is one feature dominating the classification. Note that it is like the eigen decomposition. Hence, the order of these weights has no meaning. And, the dominative feature of each model shows a similar floating range, and there is a clear boundary between two classes in this feature. It further proves that three models have learned a similar feature.

5. Conclusion

In this paper, we design a data-driven research to find out whether the gender bias is a key element and try to find some common laws/features driving the success in show business. The experiment results show that there are some common features between the success of an actor and the success of an actress. And, gender bias is weakly related to the success. We use this property to build a general model to predict the success in show business. Compared to the benchmark, the improvement of the model is obvious. In the future, we plan to do a further research on whether gender bias is a key element and try to find some common laws driving the success in other fields.

Data Availability

The data used in this study can be accessed at https://doi.org/10.17605/OSF.IO/NDTA3.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

The first author designed the study and wrote the paper. The remaining authors contributed equally to this paper in data analysis.

Acknowledgments

Chong Wu acknowledges the constructive suggestions from Prof. Jonathan Zhu. This work was supported by Mr. Jiangbin Zheng, the third author of this paper.

References

[1] O. E. Williams, L. Lacasa, and V. Latora, "Quantifying and predicting success in show business," *Nature Communications*, vol. 10, no. 1, pp. 1–8, 2019.

[2] A. M. Petersen, W.-S. Jung, J.-S. Yang, and H. E. Stanley, "Quantitative and empirical demonstration of the matthew effect in a study of career longevity," *Proceedings of the National Academy of Sciences*, vol. 108, no. 1, pp. 18–23, 2011.

[3] M. Janosov, F. Battiston, and R. Sinatra, "Success and luck in creative careers," *EPJ Data Science*, vol. 9, no. 1, 2020.

[4] D. Easley and J. Kleinberg, "Networks, crowds, and markets: reasoning about a highly connected world," *Significance*, vol. 9, pp. 43-44, 2012.

[5] A.-L. Barabási, *The Formula: The Universal Laws of Success*, Hachette Book Group, Hachette UK, 2018.

[6] S. P. Fraiberger, R. Sinatra, M. Resch, C. Riedl, and A.-L. Barabási, "Quantifying reputation and success in art," *Science*, vol. 362, no. 6416, pp. 825–829, 2018.

[7] S. Juhász, G.Ő. Tóth, and B. Lengyel, "Brokering the core and the periphery: creative success and collaboration networks in the film industry," *PLoS One*, vol. 15, no. 2, Article ID e0229436, 2020.

[8] L. Wu, D. Wang, and J. A. Evans, "Large teams develop and small teams disrupt science and technology," *Nature*, vol. 566, no. 7744, pp. 378–382, 2019.

[9] B. Moreno, V. Ciotti, P. Panzarasa, S. Liverani, L. Lacasa, and V. Latora, "Predicting success in the worldwide start-up network," *Scientific Reports*, vol. 10, no. 1, pp. 1–6, 2020.

[10] M. S. Mariani, Y. Gimenez, J. Brea, M. Martin, R. Algesheimer, and C. J. Tessone, The Wisdom of the Few: Predicting Collective Success from Individual Behavior, 2020.

[11] R. Sinatra, D. Wang, P. Deville, C. Song, and A.-L. Barabasi, "Quantifying the evolution of individual scientific impactfic impact," *Science*, vol. 354, no. 6312, Article ID aaf5239, 2016.

[12] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National Academy of Sciences*, vol. 102, no. 46, pp. 16569–16572, 2005.

[13] A. Kozbelt, "One-hit wonders in classical music: evidence and (partial) explanations for an early career peak," *Creativity Research Journal*, vol. 20, no. 2, pp. 179–195, 2008.

[14] D. K. Simonton, "Creative productivity: a predictive and explanatory model of career trajectories and landmarks," *Psychological Review*, vol. 104, no. 1, pp. 66–89, 1997.

[15] H. C. Lehman, *Age and Achievement*, Princeton University Press, Princeton, NJ, USA, 2017.

[16] D. K. Simonton, "Age and outstanding achievement: what do we know after a century of research?" *Psychological Bulletin*, vol. 104, no. 2, pp. 251–267, 1988.

[17] A. Spitz and E.Ő-Á. Horvát, "Measuring long-term impact based on network centrality: unraveling cinematic citations," *PLoS One*, vol. 9, no. 10, 2014.

[18] D. E. Acuna, S. Allesina, and K. P. Kording, "Predicting scientific success," *Nature*, vol. 489, no. 7415, pp. 201-202, 2012.

[19] O. Penner, R. K. Pan, A. M. Petersen, K. Kaski, and S. Fortunato, "On the predictability of future impact in science," *Scientific Reports*, vol. 3, no. 1, pp. 1–8, 2013.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] A. Voelker, I. Kaji´c, and C. Eliasmith, "Legendre memory units: continuous-time representation in recurrent neural networks," in *Advances in Neural Information Processing Systems*, pp. 15544–15553, MIT Press, Cambridge, MA, USA, 2019.

[22] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, Vancouver, Canada, May 2013.

[23] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA USA, February 2018.

[24] L. Wang, X. Duan, Q. Zhang, Z. Niu, G. Hua, and N. Zheng, "Segment-tube: spatio-temporal action localization in untrimmed videos with per-frame segmentation," *Sensors*, vol. 18, no. 5, p. 1657, 2018.

[25] X. Duan, Le Wang, C. Zhai et al., "Joint spatiotemporal action localization in untrimmed videos with per-frame segmentation," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 918–922, Athens, Greece, October 2018.

[26] F. Orsini, M. Gastaldi, L. Mantecchini, and R. Rossi, "Neural networks trained with WiFi traces to predict airport passenger behavior," in *Proceedings of the 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 1–7, Cracow, Poland, June 2019.

Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 8859172, 9 pages
https://doi.org/10.1155/2020/8859172

Research Article

A SAR Image Target Recognition Approach via Novel SSF-Net Models

Wei Wang,[1] Chengwen Zhang,[1] Jinge Tian,[1] Jianping Ou,[2] and Ji Li[1]

[1]*School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2]*ATR Key Lab., National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Jianping Ou; oujianping@nudt.edu.cn and Ji Li; hangliji@163.com

Received 18 May 2020; Revised 3 June 2020; Accepted 16 June 2020; Published 9 July 2020

Academic Editor: Nian Zhang

With the wide application of high-resolution radar, the application of Radar Automatic Target Recognition (RATR) is increasingly focused on how to quickly and accurately distinguish high-resolution radar targets. Therefore, Synthetic Aperture Radar (SAR) image recognition technology has become one of the research hotspots in this field. Based on the characteristics of SAR images, a Sparse Data Feature Extraction module (SDFE) has been designed, and a new convolutional neural network SSF-Net has been further proposed based on the SDFE module. Meanwhile, in order to improve processing efficiency, the network adopts three methods to classify targets: three Fully Connected (FC) layers, one Fully Connected (FC) layer, and Global Average Pooling (GAP). Among them, the latter two methods have less parameters and computational cost, and they have better real-time performance. The methods were tested on public datasets SAR-SOC and SAR-EOC-1. The experimental results show that the SSF-Net has relatively better robustness and achieves the highest recognition accuracy of 99.55% and 99.50% on SAR-SOC and SAR-EOC-1, respectively, which is 1% higher than the comparison methods on SAR-EOC-1.

1. Introduction

Radar Automatic Target Recognition (RATR) technology can achieve the target's attributes, categories, models, and other key characteristics. It can work around the clock and is robust to the environment changes. In order to obtain richer target information from radar signals, RATR technology is increasingly focused on the research of high-resolution radar. Synthetic Aperture Radar (SAR) image is a kind of high-resolution radar image. Compared with High Range Resolution Profile (HRRP), it can provide two-dimensional resolution information of targets and contain more detailed features. However, SAR images are sensitive to the changes of target attitude and speckle noise, which makes it difficult to recognize the SAR targets accurately. So, how to accurately judge the target category of SAR images has become the research focus of RATR technology.

There are two main difficulties existing in SAR image recognition: First, the scattering characteristics between different targets within the same angle may be very similar, which makes it difficult to cluster radar targets. The second is that the geometric structure information hidden in radar images, such as target size and scatter distribution, are complex and nonlinear, which leads to difficulty in information extraction.

Traditional RATR methods include K-nearest neighbor classifier (KNN) and support vector machine learning (SVM). The Principal Component Analysis (PCA) adopted by He et al. [1] has realized the rapid recognition of SAR image targets. Zhao et al. [2] applied SVM to automatic target recognition of SAR images. Trace-norm Regularized multitask learning (Trace), proposed by Obozinskiet et al. [3], assumed that all models share a common low-dimensional subspace, but its method cannot be extended to the nonlinear domain. Evgeniou and Pontil et al. [4] proposed regularized multitask learning (RMTL), which extended the existing kernel based on learning methods for single-task learning, such as SVM. Clustered Multitask Learning (CMTL) approach proposed by Zhou et al. [5] was used to replace Multitask Learning (MTL), which assumed that

multiple tasks followed a clustered structure, and it achieved a high accuracy of SAR image recognition. Zhang et al. [6] proposed the Multitask relationship learning (MTRL) approach, which can autonomously learn the positive and negative task correlation, and its recognition accuracy was higher than that of CMTL. Cong et al. [7] proposed a new classification method for clustered multitask learning theory. The method improved MTRL and learned multitask relationships autonomously. It can cluster information of different tasks and easily extended to nonlinear domain.

However, traditional SAR image target recognition technologies often require artificially designing complex feature extraction algorithms, which is difficult to implement and has poor generalization ability. The performance of target recognition algorithm is unstable when the generating environment of radar signal is different. With the development of artificial intelligence, there are more and more applications of target recognition based on deep learning [8]. In the field of optical image recognition, Convolutional Neural Networks (CNNs) have achieved great success. They are widely used in object detection and localization, semantic segmentation, speech recognition, natural language processing, image classification, and target recognition. Compared with other classification algorithms, convolutional neural networks have better robustness for translational changes [9]. Wang et al. [10] proposed a method for SAR image target recognition that combines two-dimensional principal component analysis (2DPCA) and L2 regularization constraint stochastic configuration network (SCN). They applied the 2DPCA method to extract the features of SAR images. Combining 2DPCA and SCN (random learning model with a single hidden layer), the 2DPCA-SCN algorithm have achieved good performance. Due to the limited original SAR images, it is difficult to effectively train the neural networks. In order to solve this problem, multiview deep neural network is proposed by Pei et al. [11]. The framework of this deep neural network includes a parallel network topology with multiple inputs, which can learn the features of SAR images with different views layer by layer. Chen [12] used All Convolutional Neural Network (A-CNN) [13] to the target recognition of SAR images and achieved very high recognition accuracy on the SAR image dataset under standard operating condition, but the recognition performance on SAR image dataset under extended operating condition has declined. Zou et al. [14] proposed another convolutional neural network structure for SAR image target recognition, which uses multiazimuth SAR images to improve the recognition accuracy.

Both the sparsity of SAR images and the limited SAR datasets increase the difficulty of recognition tasks. In response to the above problems, a Sparse Data Feature Extraction (SDFE) module is first designed in this paper. Based on the SDFE module, a small sample sparse data feature extraction network (SSF-Net) is proposed. In order to minimize the network parameters and improve the recognition efficiency, the network has further made improvements of the classifier. The approach in this paper is compared with those in [3–7, 10–12] and achieves higher recognition accuracy and stronger generalization ability.

2. SSF-Net Based on SDFE Module

2.1. CNNs. In recent years, CNN has been widely used in computer vision recognition tasks, and the basis structure of CNN is shown in Figure 1. In 2012, Hinton and Alex Krizhevsky proposed AlexNet [16], which successfully applied ReLU [17], Dropout [18], and LRN [17] in CNN for the first time. Visual geometry group networks (VGGNets) proposed by Simonyan and Zisserman [19] have significantly improved image recognition performance by deepening the network to 19 layers. The application of 3×3 small convolution filters is the main contribution of VGGNets. By stacking small convolutional filters, VGGNets not only increases the depth of the network but also enhances the nonlinearity of the convolution layers. Compared with large convolution filters, small filters can also effectively reduce the amount of parameters [20]. Before the VGG network was proposed, An et al. [21] also used small convolution filters, but the network was not as deep as VGGNets. In extracting target features, VGG network has very excellent performance.

Deepening the network will lead to the degradation problems. That is, after sufficient number of training, the accuracy of the training set is saturated or even decreased. In addition, the problems of gradient and information loss also hinder the increase of network depth. Residual net (ResNet) [22] solved this problem to some extent by using skip connections.

Inspired by the ResNet, Dense Convolutional Network (DenseNet) was proposed by Huang et al. [23]. By constructing dense blocks which adopt dense connections, DenseNet can deepen to more than 200 layers. Each layer in a dense block can directly access the gradient value from the loss function and the original input signal. By changing the growth rate, DenseNet can reduce the amount of parameters, but increase the computational cost [24].

2.2. SDFE Module and SSF-Nets. SAR images contain many different features from optical images. The traditional feature extraction methods need to consider the geometric features, statistical gray scale features, electromagnetic scattering features, transform domain features, local invariant features [25, 26], and so on. CNNs can adaptively learn the features of SAR images for recognition, which reduces the complexity of the recognition algorithm.

Although many studies have proved that, in the field of optical image recognition, deeper networks have better performance [22, 23]. However, the amount of SAR image data is relatively less. An overly complex network cannot significantly improve the recognition performance, and it may also carry the risk of overfitting. Therefore, the depth of the network proposed for SAR image recognition is not as deep as those of the ResNet and the DenseNet, so as to avoid the gradient disappearance problem that may appear in the late stage of training. The convolutional layer and pooling layer alternately and linearly propagate in our network. So, it can avoid skip connections to simplify the network complexity as much as possible. Due to the sparse feature of SAR

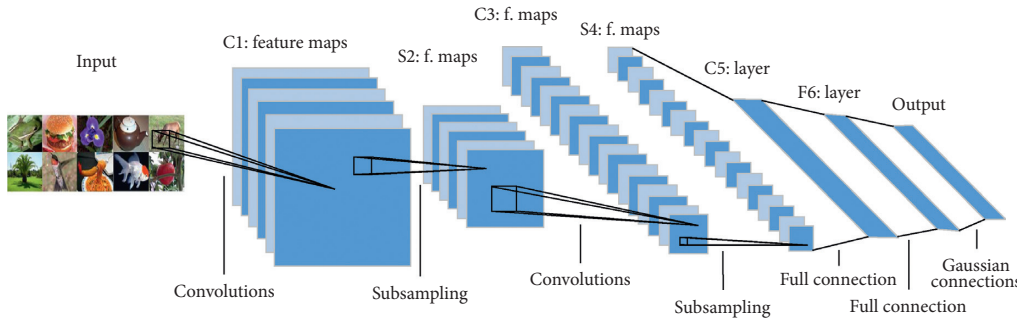


Figure 1: The basic structure of convolution neural network [15].

images, when all the features are extracted by using small convolution filters, it may not be able to fully represent all the characteristics information of the target. Therefore, a Sparse Data Feature Extraction (SDFE) module is proposed in this paper, which contains a parallel convolution layer and a point convolution layer. Convolution filters with different sizes are introduced into the parallel convolutional layer to improve the ability of the network to extract sparse features in SAR images. The SDFE structure is shown in Figure 2, where "Conv7," "Conv5," "Conv3," and "Conv1" represent convolutional layers with the filters size of 7×7, 5×5, 3×3, and 1×1, respectively. "MaxPool (3)" is the 3×3 max pooling layer with stride of 1.

The parallel convolutional layer of SDFE module utilizes 4 different filters with size of 7×7, 5×5, 3×3, and 1×1. The largest "7×7" convolutional filter in SDFE is crucial to improve the network's ability to extract feature from sparse data. The parallel convolutional layer in SDFE widens the network structure and further increases the depth of the network. The parallel convolutional layer is different from the Inception [27] module. In the Inception module, the largest convolutional filter size is 5×5, and following a point convolution layer, so its ability of sparse features extraction is limited. The SDFE parallel convolutional layer involves 7×7 convolution filters, and its input does not need to go through the point convolution layer to compress depth, which can directly extract features from the output of the upper network layer. The output of the parallel convolution module is followed by a point convolution layer after "depth concat". The output depth of the point convolution layer is consistent with the input depth to increase the nonlinearity of the network and ensure that the SDFE module does not lose the feature information generated by the parallel convolution layer.

The large-scale convolution kernel can effectively extract the target features if the input data is sparse. The sparsity of the convolutional layer would bring many benefits, such as better robustness and higher feature extraction efficiency. However, if the input data is excessive sparse, feature extraction will become more difficult. Therefore, after repeated experiments, instead of the larger convolution kernel, the 7×7, 5×5, 3×3, and 1×1 filters are used in the parallel convolutional layer to alleviate this problem.

Based on the SDFE module, we propose 4 small sample sparse data feature extraction networks (SSF-Nets), as shown in Table 1. In Table 1, a SDFE structure is counted as two layers. The

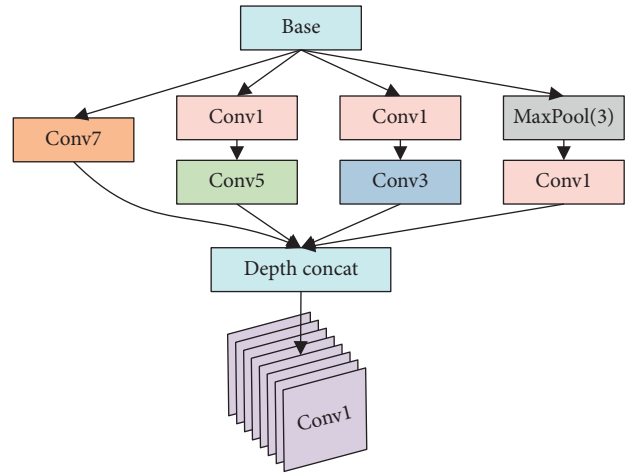


Figure 2: Structure of SDFE.

depth of the classifier in SSF-Net is set as 1. The "Conv" module in Figure 2 and Table 1 is a composite function containing "convolution," "batch normalization," and "activation function".

AlexNet, VGGNets, and some other networks' classifiers are three Fully Connected layers (3-FC), which contain more than 80% of the parameters in the whole networks [16, 19] and need high memory requirements. RATR puts forward high requirements for real-time computing, and the recognition system should minimize the consumption of hardware. In order to reduce the amount of parameters and simplify the network, our network introduces one Fully Connected layers (1-FC) as classifier to concentrate the learning tasks into the convolutional layer and lighten the burden of the fully connected layer.

In addition, we introduce the Global Average Pooling (GAP) proposed by Lin et al. [28] to replace the FC layer as the classifier. This classifier does not require fully connected layers, which can greatly reduce the number of parameters and avoid overfitting problems in the SSF-Net under certain conditions. The SSF-Nets combined with the above three classifiers are represented by "SSF-NetX-GAP," "SSF-NetX-1FC," and "SSF-NetX-3FC," where "X" indicates network's depth.

2.3. Network Complexity. If there are 4 types of targets, when using "3-FC" as the classifier, the size of output feature map generated by the last pooling (or convolution) layer of the

Table 1: SSF-Net configuration.

SSF-Net12	SSF-Net14	SSF-Net17	SSF-Net20
conv3-64	conv3-64	conv3-64	conv3-64
	conv3-64	conv3-64	conv3-64
2×2 MaxPool, stride:2			
conv3-128	conv3-128	conv3-128	conv3-128
	conv3-128	conv3-128	conv3-128
2×2 MaxPool, stride:2			
SDFE-256	conv3-256	conv3-256	conv3-256
Conv3-256	**SDFE-256**	**SDFE-256**	**SDFE-256**
		conv3-256	conv3-256
			conv3-256
2×2 MaxPool, stride:2			
SDFE-512	conv3-512	conv3-512	**SDFE-512**
SDFE-512	conv3-512	**SDFE-512**	conv3-512
		SDFE-512	conv3-512
			SDFE-512
2×2 MaxPool, stride:2			
conv3-512	**SDFE-512**	conv3-512	conv3-512
conv3-512	**SDFE-512**	conv3-512	conv3-512
		conv3-512	conv3-512
			conv3-512
2×2 MaxPool, stride:2			
Classifier, soft-max			

network is $H \times W \times D$. The parameters in the classifier are calculated as follows:

$$\begin{aligned} 3-\text{FC: Parameters} &= H \times W \times D \times 4096 \\ &\quad + 4096 + 4096 \times 4096 + 4096 + 4096 \\ &\quad \times 4 + 4 \\ &= 16,801,796 + 4096 \times H \times W \times D. \end{aligned} \tag{1}$$

When using the single layer fully connected layer "1-FC", the parameters in the classifier is calculated as follows:

$$1-\text{FC: Parameters} = H \times W \times D \times 4 + 4. \tag{2}$$

When using "GAP" as the classifier, the global average pooling is used to replace the fully connected layer. Since the pooling layer has no parameters, it can further reduce the amount of parameters. The calculation formula is as follows:

$$\text{GAP: Parameters} = D \times 4 + 4. \tag{3}$$

Through the above calculation, using the "1-FC" and "GAP" classifiers can save about 86%–92% of the parameters compared to that of the networks with "3-FC", and the networks with "GAP" can further save about 100,000 parameters than the "1-FC" networks. The parameters of the SSF-Nets with different depths and different classifiers are shown in Figure 3.

It can be seen from Figure 3 that the type of the classifier has the greatest influence on the number of network parameters, followed by the network depth. As the network depth increases gradually, the amount of network parameters only increases slowly. If the RATR system hardware conditions are poor and the memory is insufficient, using "3-FC" as the network classifier would be a bad choice.

Figure 4 shows the comparison of floating points of operations (FLOPs) of SSF-Net12, SSF-Net14, SSF-Net17, and SSF-Net20. According to Figure 4, the computation cost is most affected by the network depth. SSF-Net17 and SSF-Net20 are very computation-intensive. Compared to that of SSF-Net12, the FLOPs of SSF-Net14 has an increase of 19.82%. The FLOPs of SSF-Net17 has an increase of 53.31% compared to that of SSF-Net14, and the FLOPs of SSF-Net20 has an increase of 15.13% compared to that of SSF-Net17. So, if there is no significant difference in recognition accuracy, SSF-Net14 has the highest cost performance.

In addition, when the network depth is the same, the "3-FC" classifier has the highest computational cost, which is a fixed increase of 238.9×10^6 compared to the other two classifiers. The calculation cost of the "1-FC" is the lowest, but it is not much different from "GAP".

3. Experimental Results

3.1. Dataset. The Moving and Stationary target acquisition and recognition (MSTAR) dataset are used for the experiments. There are many research studies on radar automatic target recognition based on the MATAR SAR data set, such as [2–7, 10–12, 29]. The experimental results in this paper are compared with the above methods. The MSTAR dataset are classified into two datasets: Standard Operating Condition (SOC) dataset and Extended Operating Condition (EOC) dataset. In EOC-1 dataset, there are 4 kinds of ground targets, in which the targets with side view angle of 17° are used for training and the targets with side view angle of 30° are used for test. There are 10 kinds of targets in SOC dataset, each of which contains Omni-directional SAR image data at 15° and 17° pitch angles. In the experiments, observation data at 17° are used for training, and the observation data at 15° pitch angle are used for testing. The SAR images of MSTAR SAR-SOC dataset are shown in Figure 5.

SAR images are extremely sensitive to changes in pitch angle, so it is more difficult to identify the targets under EOC-1 conditions. The pitch angle difference between the SOC training set and test set is 2°, while the difference under the EOC-1 is increased to 13°. There is a big deviation of the same target in SAR images under the same posture, which increases the difficulty of recognition. The method in this paper has especially better recognition accuracy for SAR EOC-1 dataset and therefore has greater practical significance [7, 10, 30].

3.2. Preprocessing and Experiment Setup. In the experiments, each sample in the test set or the training set is resized to a fixed resolution of 128×128, and then the center cut and random horizontal rotation are performed. After this preprocessing, the number of SAR images has been expanded by 3 times, which compensates for the shortage of SAR images and alleviates the overfitting problem of the network to some extent.

In order to verify the validity of our approach, the experiments are completed on the same platform and environment, as shown in Table 2. The "batchsize" should be set

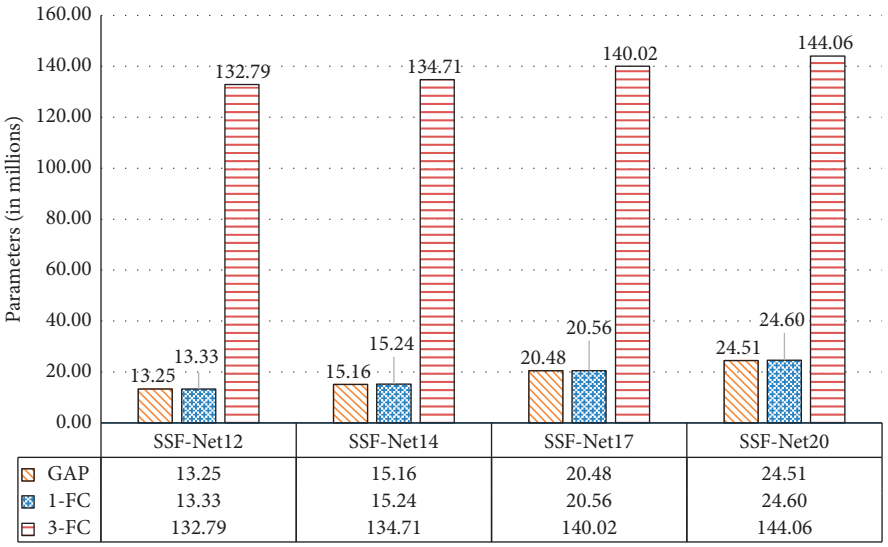


Figure 3: The parameters comparison of SSF-Net.

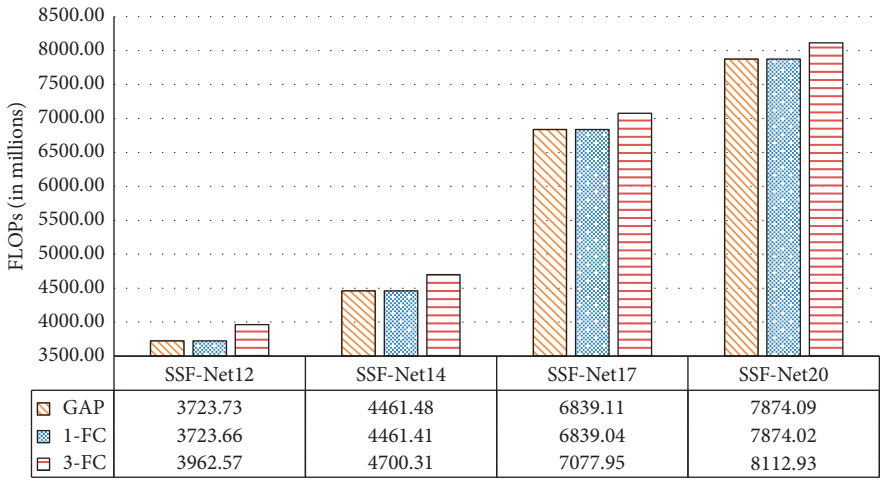


Figure 4: Comparison of floating points of operations (FLOPs).

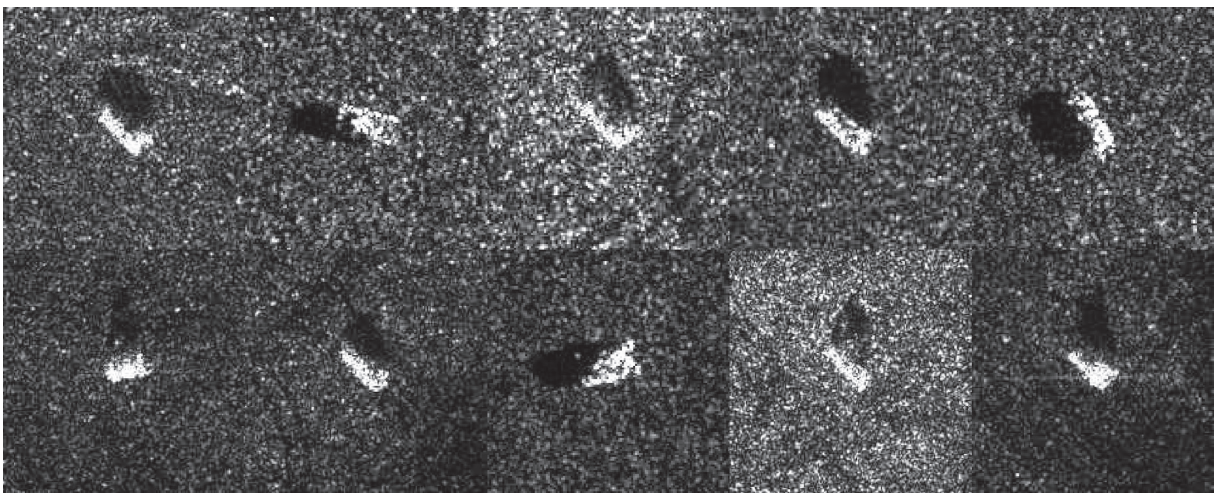


Figure 5: SAR images of MSTAR SAR-SOC dataset.

to an appropriate value. Our original intention is to set the "batchsize" as large as possible within a suitable range to make the gradient calculation of the network more accurate. However, too large "batchsize" will make the model converge to the local optimum easily. Secondly, the "batchsize" is limited to the graphics card memory. After repeated experiments, we set the "batchsize" of the training set to 16 and that of the test set to 32.

Considering that the radar data is sparse, activation function Rectified Linear Unit (ReLU) [24] will undoubtedly increase this sparseness and reduce the useful information of the target, which is unfavorable for recognition. So, we use another activation function, Hyperbolic Tangent function (Tanh), as the activation function. The resulting impact will be further analyzed in the experiments.

The learning rate attenuation method is also introduced in the training processing. As the number of iterations increases, the learning rate gradually decreases. This can ensure that the model does not fluctuate greatly in the later period of training and closer to the optimal solution. After repeated experiments, the parameters are finally adjusted as follows: the initial learning rate is set as 0.01, and 200 epochs are used for training. The learning rate decreases by 2 times since the first 50 epochs and then decreased by 2 times every 20 epochs. The average recognition accuracy of the last 100 epochs is calculated as the final results.

Table 2: Experimental platform configuration.

Attribute	Configuration information
OS	Ubuntu 14.04.5 LTS
CPU	Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30 GHz
GPU	GeForce GTX TITAN X
CUDNN	CUDNN 6.0.21
CUDA	CUDA 8.0.61
Framework	PyTorch

Table 3: Recognition accuracy rates of different depth SSF-Nets (%).

Method	SAR-SOC		SAR-EOC-1	
	Tanh	ReLU	Tanh	ReLU
SSF-Net12-3FC	98.49	99.19	95.32	97.55
SSF-Net12-1FC	97.47	99.09	97.02	96.58
SSF-Net12-GAP	99.33	98.99	97.17	97.02
SSF-Net14-3FC	99.27	99.34	**99.50**	98.59
SSF-Net14-1FC	**99.37**	99.20	**99.24**	97.96
SSF-Net14-GAP	99.18	99.43	99.05	97.55
SSF-Net17-3FC	99.39	99.37	99.36	98.92
SSF-Net17-1FC	99.31	99.35	98.81	98.02
SSF-Net17-GAP	**99.55**	**99.45**	98.78	95.67
SSF-Net20-3FC	99.43	99.35	98.47	**99.16**
SSF-Net20-1FC	99.54	99.34	98.69	98.63
SSF-Net20-GAP	99.42	99.30	99.33	98.11

3.3. Experimental Results. To focus on the impact of SSF-Net depth on recognition performance, we conducted experiments on the two MSTAR SAR datasets with the 4 depth SSF-Nets in Table 1 and the results are shown in Table 3.

According to Table 3, the recognition performance of SSF-Net12 is lower than that of the other 3 deeper networks. Because its structure is too simple to fully learn SAR image features for recognition, on SAR-EOC-1, SSF-Net14-3FC achieves the highest accuracy of 99.50%. The accuracies of SSF-Net14 with three different classifiers are 99.50%, 99.24%, and 99.05%, respectively, which are better than those of SSF-Net17 and SSF-Net20. On the SAR-SOC dataset, although SSF-Net17-GAP achieves the highest accuracy of 99.55%, most of the networks (except SSF-Net12) also achieve the accuracies higher than 99.3%. Because the difference of pitch angle between training set and test set of SAR-EOC-1 dataset is far greater than that of SAR-SOC dataset, the identification difficulty is greater, which requires the network to have strong generalization ability. Therefore, simply increasing the network depth does not significantly improve the networks' recognition performance, which also verifies that the excessively deep convolutional neural network is not conducive to SAR image recognition.

Based on the experimental results of SAR-EOC-1 in Table 3, we believe that SSF-Net14 has the best overall performance. SSF-Net14-1FC achieves 99.37% accuracy rates on SOC, only 0.18% lower than the highest accuracy achieved by SSF-Net14-3FC. On EOC-1, SSF-Net14-1FC also achieves 99.24% accuracy rates, only 0.26% lower than the highest accuracy achieved by SSF-Net17-GAP. "3-Fc" classifier has a large number of parameters and calculation, while "1-FC" classifier has a small number of parameters and calculation. Although "1-FC" has slightly more parameters than "GAP", the computational cost is less. Next, we will compare the results of the SSF-Net14-1FC with GoogLeNet [27], ResNet-18 [18], and DenseNet-121 [19]. The results are shown in Table 4.

GoogLeNet achieves high recognition accuracies on SAR-SOC, but its recognition accuracies on SAR-EOC-1 are poor, which only 90.62% and 90.19%. This shows that its generalization ability is not so ideal. ResNet-18 and DenseNet-121 further deepen the network and apply skip connections to alleviate the gradient disappearance problem. However, the accuracy rates on SAR image recognition are still lower than that of our proposed network. Shallow networks have good capabilities of feature extraction and learning, so the networks with complex structures such as DenseNet-121 and ResNet-18 may bring overfitting problems to a certain extent. Based on horizontal comparison of the recognition accuracies of the activation functions, Tanh and ReLU in Tables 3 and 4, we can see the performance of Tanh on SAR-EOC-1 is generally stronger, indicating that Tanh has better effect on sparse data processing.

We further compare SSF-Net14-1FC with the methods proposed by Wang [10], Pei [11], and Chen [12], et al., and the results are shown in Table 5.

Although some methods such as A-CNN can achieve accuracy of 99.41% on the SAR-SOC, it is difficult to achieve satisfactory results on SAR-EOC-1 data which have greater difference in pitch angles. The 2DPCA-SCN method achieves 98.49% accuracy on SAR-EOC-1, but only 95.80% on SAR-SOC. Other methods on the SAR-EOC-1 also achieve

Table 4: Recognition accuracy rates of other CNNs (%).

Method	SAR-SOC		SAR-EOC-1	
	Tanh	ReLU	Tanh	ReLU
GoogLeNet	98.87	98.65	90.62	90.19
ResNet-18	97.20	97.90	78.45	82.25
DenseNet-121($k = 32$)	98.66	98.93	96.41	98.66
SSF-Net14-1FC	**99.37**	99.20	**99.24**	97.96

Table 5: Recognition accuracy rates of other CNNs (%).

Method	SAR-SOC	SAR-EOC-1
2DPCA-SCN [10]	95.80	98.49
2-Views DCNNs [11]	97.81	93.29
3-Views DCNNs [11]	98.17	94.34
4-Views DCNNs [11]	98.52	94.61
A-CNN [12]	**99.41**	97.13
SSF-Net14-1FC	**99.37**	**99.24**

lower recognition accuracies than SSF-Net. It can be found from Table 3 that SSF-Net achieves very high accuracy on both SAR-SOC and SAR-EOC-1 dataset. Especially on SAR-EOC-1 dataset, SSF-Net can achieve higher accuracy and more stable performance, which shows that our network has stronger generalization ability and better robustness.

SSF-Net is also compared with nondeep learning approaches (such as KNN, SVM, and SRC [7, 29]), and the results are shown in Table 6. Among them, "I-MTRL" is a new classification approach of clustering multitask learning theory. SRC [29] is a recognition approach based on Sparse Representation-based Classifier proposed in 2016.

Table 6 shows that some traditional approaches are not so effective, such as KNN and SVM methods. Although many complex classifiers have been designed, they cannot fully utilize the potential correlation between multiple radar categories. On the contrary, large-scale and complete SAR datasets are difficult to collect, so the samples obtained are usually limited or unbalanced. Traditional approaches are not able to share all the information, making it difficult to get good training results. Dong et al. [31] proposed a joint sparse representation model to take advantage of the correlation between multiple tasks of SAR ATR, and comparative experiments have demonstrated the superiority of multitask learning.

The classification algorithm approaches under the multitask framework has higher recognition accuracies, such as CMTL, MTRL, and I-MTRL. The multitask relational learning (MTRL) method proposed in [6] can autonomously learn the correlation between positive and negative tasks, and it can be easily extended to the nonlinear field. The MTRL is further improved by adding a projection regularization term to the objective function [7], which can independently learn multitask relationships, cluster information of different tasks, and can also be easily extended to nonlinear field. However, the Trace-norm Regularized multitask learning (TRACE), which is also under the multitask framework, has the lowest recognition accuracy. Because the TRACE method learns the linear

Table 6: Recognition accuracies rate of traditional approaches (%).

Method	SAR-SOC	SAR-EOC-1
KNN [2]	92.71	91.42
SVM [2]	90.17	86.73
SRC [29]	89.76	—
TRACE [3]	75.04	67.42
RMTL [4]	92.09	92.03
CMTL [5]	93.91	94.72
MTRL [6]	95.84	95.46
I-MTRL [7]	97.34	98.24
SSF-Net14-1FC	**99.37**	**99.24**

prediction function and cannot accurately describe the nonlinear structure of SAR image, it also proves the importance of extending the multitask learning method to the nonlinear field.

The SSF-Net proposed in this paper can adaptively learn the nonlinear structure of SAR images and reduce the difficulty of redesigning the classifier when the SAR image conditions change. In contrast, the artificially designed feature extraction approach is complex, and sometimes it can only be effective for certain fixed problems. Its generalization ability is not so ideal. Therefore, our networks enhance the feature extraction capability of sparse data.

3.4. Experiments Analysis. SSF-Net17-GAP and SSF-Net14-3FC achieved the highest accuracy rates, 99.55% and 99.50%, on SAR-SOC and SAR-EOC-1 dataset, respectively. After a comprehensive selection, we compare the SSF-Net14-1FC with a variety of methods. It has achieved recognition accuracies, 99.37% and 99.24%, on SAR-SOC and SAR-EOC-1 dataset, which are higher than most of the accuracies achieved by other approaches.

By analyzing the different network structures and comparing the experimental results, the following conclusions are obtained:

(1) The networks should not be too deep, and the structure should be as concise as possible. Due to the small amount of data in radar signal, some complex and deep networks, such as ResNets and DenseNets, may face the problem of overfitting.

(2) Due to the sparsity of SAR images, large convolutional filters can be considered for feature extraction in the network. Different from the traditional sparse signal processing method [32], the SDFE module is designed to improve the network's ability to extract features from sparse data. However, the convolution filters in the first layer should not be too large. In this paper, we adopt 3×3 filters in the first convolution layer. Different from traditional optical images, SAR images do not have obvious edge features and texture information, so in the first layer, large-scale convolution filters cannot be used at quickly capture SAR image target edges and other features. On the contrary, the use of large-scale convolution filters at the first layer may cause excessive loss of detail information, which is not conducive to identification.

(3) The network for SAR image targets recognition should increase the ability to learn nonlinear structures. Drawing on the view that the multitask learning method should be extended to the field of nonlinearity, the SDFE module increases the nonlinearity of the network with a point convolution layer that has no compression depth.

On SAR-EOC-1, Tanh has generally better performance. The main reason is that the SAR images have sparsity and the activation function ReLU may overenhance this nature. Excessively sparse data will weaken the ability of the convolutional layer to extract target features. And Tanh has a slightly better nonlinearity, so its performance is better when the original data features are significantly different. Overall, Tanh has better activation for radar signals.

4. Conclusions

In this paper, a feature extraction SDFE module and SSF-Net for sparse data is designed, which has good performance for radar targets recognition.

One of the advantages of SSF-Net is that it can achieve high accuracy on both SAR-SOC and SAR-EOC-1. On SAR-SOC, the accuracy rate of SSF-Net14-1FC has only 0.18% lower than the highest accuracy rate achieved by the SSF-Net17-GAP. However, it saves 25.84% parameters and 34.77% FLOPs than SSF-Net17-GAP. On SAR-EOC-1, the accuracy rate of SSF-Net14-1FC is only 0.26% lower than the highest accuracy rate, but it saves more than 88.6% of the parameters. SSF-Net14-1FC saves at least 36.97% FLOPs than SSF-Net17-3FC and SSF-Net20-GAP. Therefore, SSF-Net can achieve better recognition performance for SAR images with a shallow network, improves the computational efficiency, and saves parameter space.

The SDFE module, as the most important part in SSF-Net, has three advantages. Firstly, the SDFE module can effectively extract the target features when the input data is sparse. Secondly, the SDFE module improves the nonlinearity of SSF-Net, which can strengthen the SSF-Net's ability to fit the nonlinear structure of SAR images. Lastly, the SDFE module increases the robustness and computational efficiency of SSF-Net, so the SSF-Nets can achieve high accuracies on SAR-EOC-1 with fewer layers.

When deepening the network, the recognition algorithm may be invalid. It is because the down-sampling layers in the deep neural network are too many for SAR images. To solve this problem, one feasible method is to reduce the down-sampling layers of the deep neural network, but it will weaken the robustness of the network and increase the computational cost. Another solution is to design shallow convolutional neural networks, such as our SSF-Nets proposed in this paper.

According to the imaging characteristics of SAR images, another feasible method to improve the target recognition rate is target classification and recognition based on image superresolution reconstruction [33], which is also a key research direction at present.

Data Availability

All datasets in this article are public datasets and can be found on public websites.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

W.W. and J.O. conceptualized the study; J.O. and C.Z. carried out the methodology; C.Z. and J.T. helper with software; J.L. carried out formal analysis; C.Z. carried out investigation; W.W. and J.O carried out writing and preparing the original draft.

Acknowledgments

This research was funded by National Defense Pre-Research Foundation of China under Grant 9140A01060314KG01018, National Natural Science Foundation of China under Grant 61471370, Equipment Exploration Research Project of China under Grant 71314092, Scientific Research Fund of Hunan Provincial Education Department under Grant 17C0043, and Hunan Provincial Natural Science Fund under Grant 2019JJ80105.

References

[1] Z. He, J. Lu, and G. Kuang, "Fast SAR target recognition approach using PCA features," in *Proceedings of the International Conference on Image & Graphics*, pp. 580–585, IEEE, Sichuan, China, August 2007.

[2] Q. Zhao and J. C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 2, pp. 643–654, 2001.

[3] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statistics and Computing*, vol. 20, no. 2, pp. 231–252, 2010.

[4] T. Evgeniou and M. Pontil, "Regularized multi-ttask learning," in *Proceedings of the Knowledge Discovery and Data Mining*, pp. 109–117, Washington, DC, USA, August 2004.

[5] J. Zhou, J. Chen, J. Ye et al., "Clustered multi-task learning via alternating structure optimization," *Neural Information Processing Systems*, vol. 2011, pp. 702–710, 2011.

[6] Y. Zhang and D.-Y. Yeung, "A regularization approach to learning task relationships in multitask learning," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 3, pp. 1–31, 2014.

[7] L. Cong, B. Weimin, X. Luping et al., "Clustered multi-task learning for automatic radar target recognition," *Sensors*, vol. 17, no. 10, Article ID s17102218, 2017.

[8] D. Meng and L. Sun, "Some new trends of deep learning research," *Chinese Journal of Electronics*, vol. 28, no. 6, pp. 1087–1090, 2019.

[9] D. Malmgren-Hansen, R. Engholm, and M. O. Pedersen, "Training convolutional neural networks for translational invariance on SAR ATR," in *Proceedings of the EUSAR 2016: 11th European Conference on Synthetic Aperture Radar*, pp. 1–4, Hamburg, Germany, September 2016.

[10] Y. Wang, Y. Zhang, Li. Yang et al., "Target recognition method based on 2DPCA-SCN regularization for SAR images [J]," *Journal of Signal Processing*, vol. 35, no. 5, pp. 802–808, 2019, in Chinese.

[11] J. Pei, Y. Huang, W. Huo, Y. Zhang, J. Yang, and T.-S. Yeo, "SAR automatic target recognition based on Multiview deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2196–2210, 2018.

[12] Y. Cheng, L. Yu, and X. Xie, "SAR image target classification based on all congvolutional neural network[J]," *Radar Science and Technology*, vol. 016, no. 3, pp. 242–248, 2018, in Chinese.

[13] J. T. Springenberg, A. Dosovitskiy, T. Brox et al., "Striving for simplicity: the all convolutional net," 2014, http://arxiv.org/abs/1412.6806.

[14] H. Zou, L. Yun, and H. Wen, "Research on multi-aspect SAR images target recognition using deep learning," *Journal of Signal Processing*, vol. 34, no. 5, pp. 512–522, 2018, in Chinese.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, Curran Associates Inc., Doha, Qatar, pp. 1097–1105, November 2012.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the International Conference on Machine Learning*, pp. 807–814, Haifa, Israel, January 2010.

[18] G. E. Hinton, N. Srivastava, A. Krizhevsky et al., "Improving neural networks by preventing co-adaptation of feature detectors," 2012, http://arxiv.org/abs:arxiv/1207.0580.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, Bejing, China, June 2014.

[20] W. Wang, Y. Yang, X. Wang et al., "The development of convolution neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, Article ID 040901, 2019.

[21] D. Ciresan, U. Meier, J. Masci et al., "Flexible, high performance convolutional neural networks for image classification," *International Joint Conference on Artificial Intelligence*, vol. 30, pp. 1237–1242, 2011.

[22] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[23] G. Huang, Z. Liu, V. D. M. Laurens et al., "Densely connected convolutional networks," in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, pp. 2261–2269, Honolulu, HI, USA, July 2017.

[24] W. Wang, Y. Li, T. Zou et al., "A novel image classification approach via dense-MobileNet models," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[26] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin, "SAR-SIFT: a SIFT-like algorithm for SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 453–466, 2015.

[27] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[28] M. Lin, Q. Chen, S. Yan et al., "Network in network," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.

[29] H. Song, K. Ji, Y. Zhang, X. Xing, and H. Zou, "Sparse representation-based SAR image target classification on the 10-class MSTAR data set," *Applied Sciences*, vol. 6, no. 1, p. 26, 2016.

[30] J. C. Mossing and T. D. Ross, "An evaluation of SAR ATR algorithm performance sensitivity to MSTAR extended operating conditions," in *Proceedings of SPIE Algorithms for Synthetic Aperture Radar Imagery*, pp. 554–565, Orlando, FL, USA, April 1998.

[31] G. Dong, G. Kuang, N. Wang et al., "SAR target recognition via joint sparse representation of monogenic signal," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, vol. 8, no. 7, pp. 3316–3328, 2017.

[32] W. Wang, C. Tang, X. Wang et al., "Image object recognition via deep feature-based adaptive joint sparse representation," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8258275, 9 pages, 2019.

[33] W. Wang, Y. Jiang, Y. Luo et al., "An advanced deep residual dense network (DRDN) approach for image super-resolution," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592–1601, 2019.