


## Research Article

# Recognition of Electromagnetic Signals Based on the Spiking Convolutional Neural Network

Shifei Tao <sup>1,2</sup>, Shuai Xiao,<sup>2</sup> Shuaige Gong,<sup>1</sup> Huifu Wang,<sup>2</sup> Hao Ding,<sup>3</sup> and Hao Wang <sup>2</sup>

<sup>1</sup>State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE), Luoyang 471003, China

<sup>2</sup>School of Electrical and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

<sup>3</sup>Engineering Training Center, Nanjing University of Science and Technology, Nanjing 210094, China

Correspondence should be addressed to Shifei Tao; [s.tao@njust.edu.cn](mailto:s.tao@njust.edu.cn)

Received 20 May 2022; Accepted 10 June 2022; Published 24 June 2022

Academic Editor: Mingqian Liu

Copyright © 2022 Shifei Tao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feature extraction and recognition of signals are the bases of a cognitive radio. Traditional manual extraction for signals' features becomes difficult in the complex electromagnetic environment. Although convolutional neural networks (CNNs) can extract signal features automatically, they have low accuracy in recognizing electromagnetic signals at low signal-to-noise ratios (SNRs) due to the agility of signals. Considering the great potential of spiking neural networks (SNNs) in classification, a spiking convolution neural network (SCNN) for the recognition of electromagnetic signals is proposed in this paper. The SCNN effectively integrates the extraction ability of spatial features in CNNs and temporal features in SNNs. Since the SCNN is difficult to train, the strategy of surrogate gradient is proposed to train it. By taking the 2-dimensional time-frequency distribution of 6 signals as input, the SCNN can effectively identify different signals at low SNRs. The method proposed in this paper contributes to promote the research and application of SNNs in the recognition of electromagnetic signals.

## 1. Introduction

Modern electronic warfare is rapidly developing with the rise of electronic information technology. Electromagnetic signal identification becomes a critical part of a cognitive radio (CR) [1]. However, electromagnetic signal waveforms appear to be agile and heavily interfered in a diverse electromagnetic environment. The agility of signals reduces the performance of earlier recognition methods based on the interpulse feature [2]. Therefore, in-pulse features such as time-frequency features, wavelet packet features, wavelet ridge-frequency features, and higher-order spectrum are widely used for signal recognition by combining support vector machines (SVM) and deep learning (DL) [3]. As a type of DL, convolutional neural networks (CNNs) are widely used in target recognition and image classification. Due to their powerful capability for feature extraction and generalization, CNNs are quite effective in the recognition of radar signals [4, 5] and communication signals [6, 7].

In recent years, many studies have been done on electromagnetic signal recognition with CNNs. Some methods focus on preprocessing the input signals of CNNs to reduce noise interference [8–10]. Ye et al. use the time-frequency distribution preprocessed with binarization as the input of a 3-layer CNN for signal recognition. The recognition accuracy of signals at  $-6$  dB is still above 90% [8]. Yao and Wang preprocess the time-frequency maps with more methods including symmetric mapping, primary energy ridge extraction, binarization, and image reset to reduce noise interference. Then, they use a pretrained CNN to identify the preprocessed time-frequency maps and obtain a higher accuracy than methods with manual feature extraction [9]. Denoising is also an optional method for preprocessing. When using a residual neural network (ResNet) to denoise time-frequency maps of radar signals, the Inception-V4 network which combines the inception structure and residual connections achieves a recognition accuracy of over 90% for signals at  $-10$  dB [10]. Meanwhile, different features are constructed for signal recognition [11, 12]. Pu et al. use

orthogonal slices of Gaussian-smoothed fuzzy function as a feature extraction target. By learning the features of the orthogonal slices, the employed CNN achieves a recognition accuracy of 88.5% for signals at  $-6$  dB [11]. Xie et al. firstly use chirp decomposition for preliminary classification of signals. Then, they take the Zernike matrix extracted from time-frequency maps as the input feature of a ResNet to complete further classification of the signals. Their method finally achieves good robust performance on signals' parameters [12]. Some other methods are committed to improving the structure of networks for improving the recognition accuracy [13–15]. Lin et al. propose a deep residual shrinkage attention network for signal recognition. They use the attention mechanism in ResNet to reduce the impact of redundant information. Finally, the deep residual shrinkage attention network achieves excellent recognition accuracy and good scalability at the same time [13]. In addition, better performance than either network alone can be achieved by combining the long- and short-term memory network (LSTM) and CNN. The combined network fuses time-domain features with time-frequency domain features, providing some improvement in the performance for signal recognition at low SNRs [14, 15].

Nonetheless, there are still some challenges to be considered and addressed. (1) Most of the CNN-based methods have low recognition accuracy for signals with low SNRs. (2) Recognizing electromagnetic signals with CNNs requires preprocessing methods or manually constructed features to reduce noise interference, increasing the complexity of signal processing. (3) Rare electromagnetic signal data, especially radar signal data, is difficult to meet the requirement for training deep CNNs. Furthermore, too large networks cannot easily be applied to mobile devices.

In contrast, spiking neural networks (SNNs) have great potential for development due to their biointerpretability and low power consumption. Li et al. use Gaussian-tuned coding to convert the time-frequency maps into pulses. Then, a single-layer SNN composed of tempotron neurons is used to process the pulses for signal recognition. Their simulations show that a single-layer SNN performs better than a 3-layer CNN [16]. Li et al.'s results suggest that a simple SNN may have better performance than a deeper CNN. Although deeper SNNs were less used due to difficulties in building and training in the beginning, some studies have shown that SNNs can be transformed from CNNs [17–19] or trained with a surrogate gradient [20, 21]. New methods for building and training SNNs enable SNNs to have good performance in classification. SNNs have achieved superior performance in efficiently processing complex and noisy spatial-temporal information [22]. SNNs show a higher potential for robustness in many ways compared to CNNs [23]. The information is less susceptible to random background noise in SNNs [24].

Thus, the purpose of this study is twofold. First, we propose a spiking convolution neural network (SCNN) for signal recognition in this paper. The SCNN combines the SNN with the CNN. It integrates the temporal and spatial information of the feature maps for better feature extraction. At the same time, the SCNN has a smaller model size com-

pared to CNNs with the same accuracy. Second, the method of surrogate gradient for training the SCNN is given in this paper. In addition, two key parameters that affect the performance of the SCNN are analyzed and explained in detail. The remainder of this paper is summarized as follows. Section 2 describes the principles of time-frequency transform and SNNs besides the methods used in this paper. Section 3 determines the optimal values of two key parameters of the SCNN and compares the SCNN's performance with those of other networks. The noise generalization performance of the SCNN is analyzed at the end of this section. Finally, the conclusion is given in Section 4.

## 2. Theory and Methods

Figure 1 illustrates the process of identifying electromagnetic signals using the SCNN. The signals are first transformed into time-frequency maps. Then, the time-frequency maps are used as the input of the SCNN to complete the classification. Therefore, the first two layers of the SCNN are convolutional layers which are each followed by a layer composed of leaky integrate and fire (LIF) neurons. The feature maps after LIF neurons are then mapped to a pooling layer to reduce the dimension. And the two fully connected layers are each followed by a layer consisting of integrate and fire (IF) neurons. The number of LIF and IF neurons is the same as the size of the feature map of the previous layer. So, the LIF and IF layers do not change the size of the feature map.

*2.1. Time-Frequency Distribution of Electromagnetic Signals.* In order to avoid the features of one-dimensional signals being affected by noise, we use time-frequency distributions to get the features of signals. Different kernel functions correspond to different time-frequency distributions. A series of time-frequency distributions have been proposed, mainly including the Choi-Williams distribution (CWD), the smoothed pseudo-Wigner-Ville distribution (SPWVD), and the reduced interference distribution (RID) [25]. The CWD with minimal cross-terms among all unprocessed Cohen-like distributions is chosen in this paper. Its kernel function is shown in equation (1) as follows:

$$\phi(\theta, \tau) = e^{-\theta^2 \tau^2 / \sigma}. \quad (1)$$

The corresponding CWD is described in equation (2) as follows:

$$C(t, w) = \iiint \phi(\theta, \tau) s\left(u + \frac{\tau}{2}\right) s^*\left(u - \frac{\tau}{2}\right) e^{-j2\pi w\tau} e^{j2\pi(\theta u - \theta\tau)} dud\theta d\tau. \quad (2)$$

We use six signals commonly used in communication and radar, including the binary phase shift keying signal (2PSK), binary frequency shift keying signal (2FSK), quadratic frequency shift keying signal (4FSK), continuous wave signal (CW), linear frequency modulated signal (LFM), and nonlinear frequency modulated signal (NLFM). The Choi-Williams distributions of the 6 signals at 0 dB are given in Figure 2.

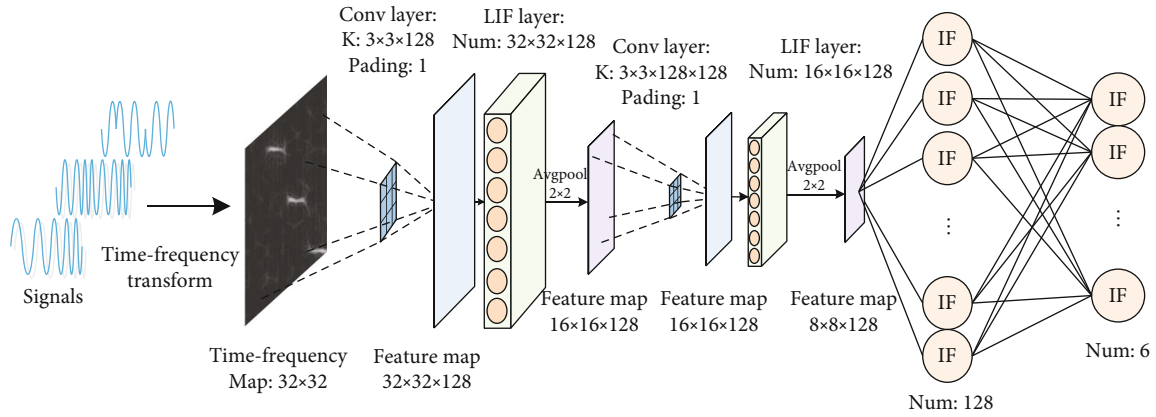


FIGURE 1: The process of identifying electromagnetic signals with the SCNN.

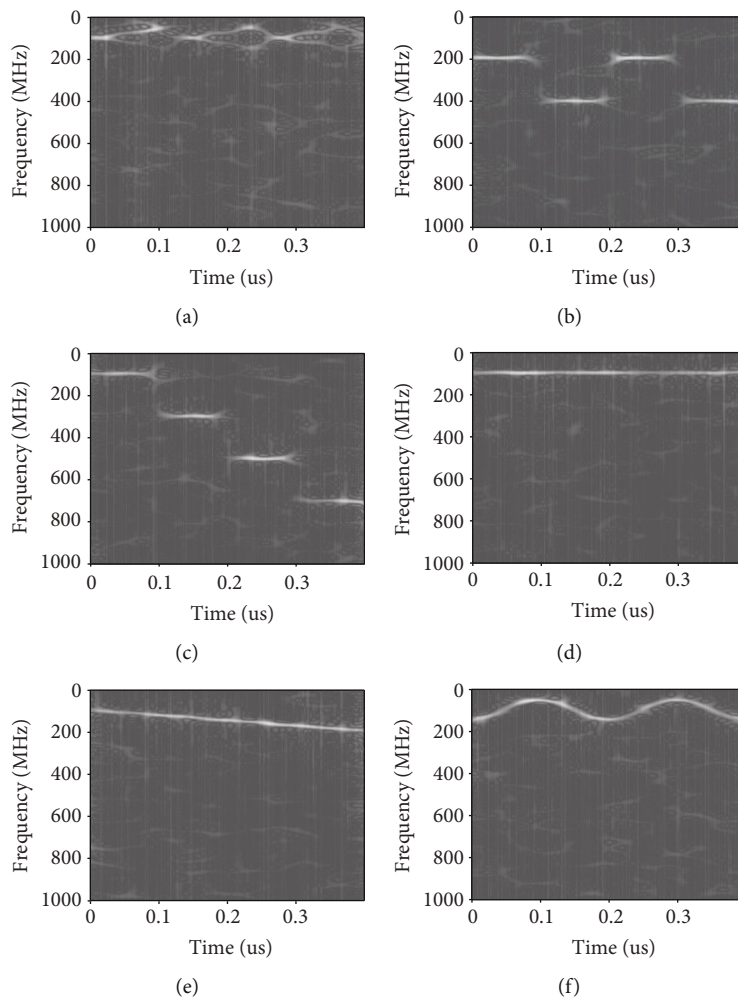


FIGURE 2: The Choi-Williams distributions of 6 electromagnetic signals at 0 dB. (a) 2PSK. (b) 2FSK. (c) 4FSK. (d) CW. (e) LFM. (f) NLFM.

2.2. *Spiking Neural Network.* Various spiking neuron models have emerged, including the Hodgkin-Huxley (HH) model, integrate-and-fire (IF) model, leaky-integrate-and-fire (LIF) model, and Izhikevich model [22]. Although its biological approximation is favourable, the HH model is computationally complex and inconvenient to use in practice. The sim-

plified LIF and IF models have become common and widely used models, since they reduce the computational effort of the model while retaining the threshold-based membrane potential change rules of biological neurons. Both LIF and IF neurons are integral models. Their membrane potential is influenced not only by the input at the

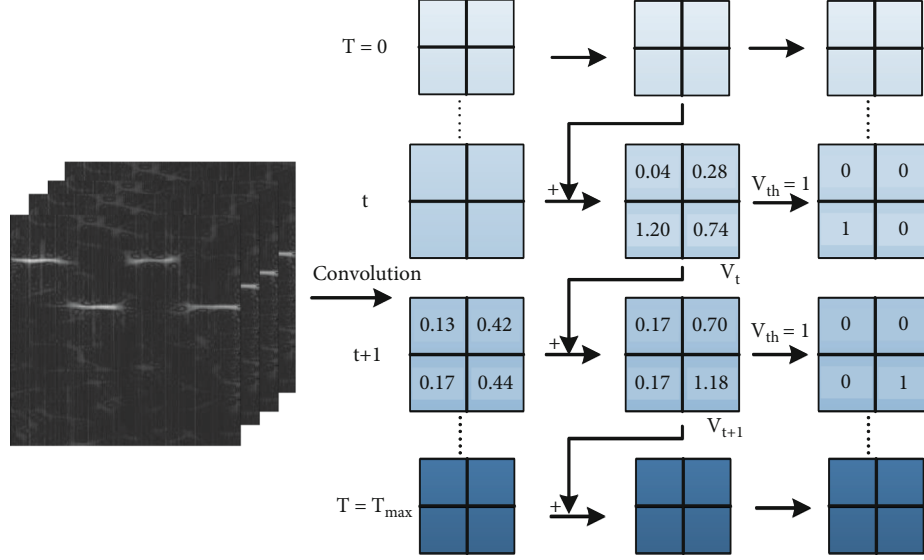


FIGURE 3: Activation of the SCNN based on threshold-based membrane potential change rules.

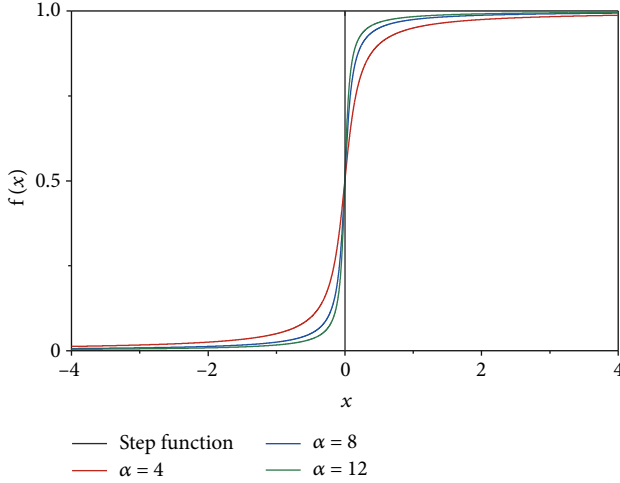


FIGURE 4: Approximation effect of the Atan function with different  $\alpha$ .  $f(x)$  is a step function or Atan function.

current moment but also by the membrane potential at the end of the previous moment. When the membrane potential does not exceed the threshold, the charging process of a continuous-time neuron can be expressed as equation (3) as follows:

$$\frac{dV(t)}{dt} = f(V(t), X(t)). \quad (3)$$

Equations (4) and (5) show the charging process of LIF and IF neurons, respectively.

$$\tau_n \frac{dV(t)}{dt} = -(V(t) - V_{\text{reset}}) + X(t), \quad (4)$$

$$\frac{dV(t)}{dt} = \frac{1}{C} X(t), \quad (5)$$

where  $\tau_n$  is the time constant and  $V_{\text{reset}}$  is the reset voltage. An analytical solution to equation (4) cannot be obtained as the input  $X(t)$  is a variable. The continuous differential equation is approximated by the discrete differential equation (6).

$$\begin{aligned} \tau_n (V[t] - V[t-1]) &= -(V[t-1] - V_{\text{reset}}) + X[t], \\ V[t] &= V[t-1] + X(t). \end{aligned} \quad (6)$$

For both discrete LIF and IF neurons, their forward propagation includes two processes, discharge and reset, in addition to the charging process. The discharge process can be expressed in equation (7).

$$S(t) = H(V_{\text{tmp}} - V_{\text{threshold}}), \quad (7)$$

where  $V_{\text{tmp}}$  is the voltage of the neuron before releasing the pulse after charging.  $V_{\text{threshold}}$  is the threshold voltage at which the neuron is activated.  $S(t)$  is the output pulse with a value of 1 for a released pulse and 0 for an unreleased pulse.  $H(x)$  is the step function. For LIF and IF neurons, the difference lies in the charging process. The membrane potential of IF neurons remains unchanged in the absence of input, whereas the membrane potential of LIF neurons gradually falls below the resting potential in the absence of input due to leakage and then rises back to the resting potential.

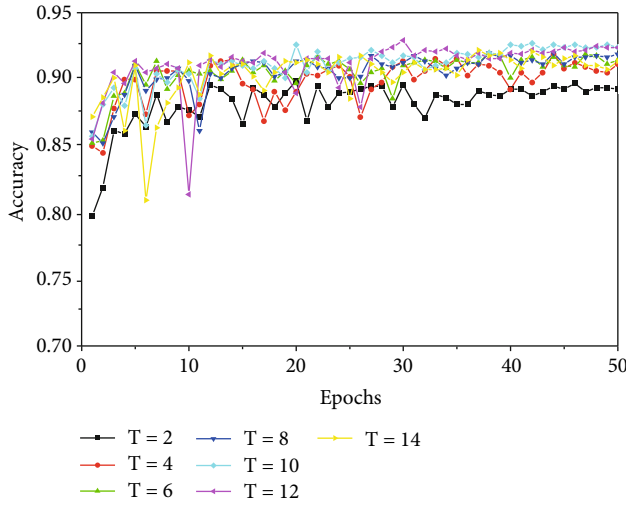
**2.3. Forward Propagation.** As shown in Figure 1, the first convolutional layer is used directly for feature extraction and has no temporal state. It forms an autocoding layer with the LIF neurons in the second layer. Since the input of the first layer does not vary with time, additional computational effort can be reduced by placing this layer outside the process of time steps. In addition, batch normalization is

TABLE 1: Parameters of 6 signals.

Signals	Frequency (MHz)	Other parameters	Sampling rate (GHz)
2PSK	100	13-bit barker code encoding	2
2FSK	200/400	/	2
4FSK	100/300/500/700	/	2
CW	100	/	2
LFM	100	Time width: 0.4 $\mu$ s, bandwidth 60 MHz	2
NLFM	100	Time width: 0.4 $\mu$ s, bandwidth 60 MHz	2

TABLE 2: Simulation environment.

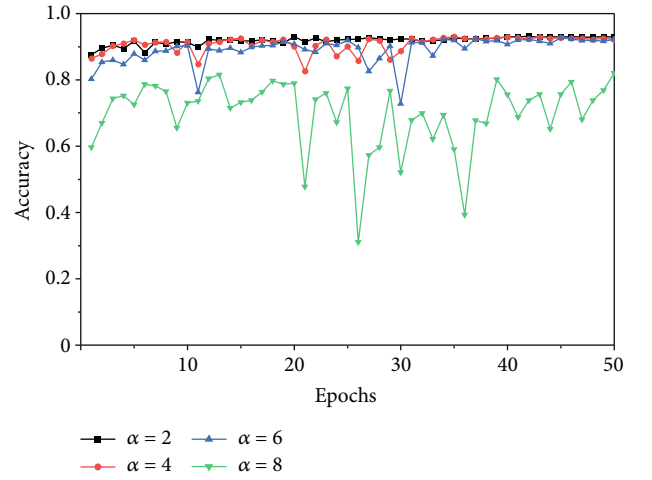
Environment	CPU	GPU	Coding language	Framework
Version	Intel i7-8700K@3.7 GHz	RTX2080	Python3.8.5	Pytorch1.8.1

FIGURE 5: Impact of  $T$  on SCNN's performance when tested with signals at  $-12$  dB.

performed after each convolution layer to accelerate the convergence of the SCNN.

Figure 3 shows the activation process of the SCNN in forward propagation. The eigenvalues of the time-frequency maps are assigned to the LIF neurons and then start to accumulate at the moment  $T = 0$ . When the activation threshold is reached, the LIF neuron releases a pulse and sets the membrane potential to  $V_{\text{reset}} = 0$ . If the neuron does not activate, the membrane potential is superimposed directly to the next moment  $T = 1$ . At the same time, if there is no input, the membrane potential will decay at a certain rate. This process is repeated until the end of the given time step. The size of the output feature map after the LIF neurons remains the same as the input size. This process completes the extraction of temporal features and achieves the fusion of spatial and temporal information.

Unlike the fully connected layers of CNNs which use probability for classification, the SCNN classifies signals by the frequency of pulses issued in time  $T$ . A higher frequency indicates a higher probability. The pulse frequency can be

FIGURE 6: Impact of  $\alpha$  on SCNN's performance when tested with signals at  $-12$  dB.

obtained as in equation (8), where  $j$  is the class of the signals and  $S_{\text{sum}}^j$  denotes the cumulative number of pulses issued by the  $j_{\text{th}}$  neuron in time  $T$ .

$$\text{Out}_j = \frac{S_{\text{sum}}^j}{T}. \quad (8)$$

Among all neurons in the last layer, the neuron with the highest pulse frequency corresponds to the predicted class of the input signal. Finally, the recognition accuracy of the SCNN can be determined by equation (9).

$$\text{Acc} = \frac{N_{\text{pre}}}{N_{\text{total}}}, \quad (9)$$

where  $N_{\text{pre}}$  is the number of signals correctly predicted and  $N_{\text{total}}$  is the total number of signals.

**2.4. Backward Propagation.** If the SCNN updates the weight  $w$  by backpropagating the error  $E_{\text{total}}$  with gradient descent, the chain rule in equation (10) gives its gradient transfer.

TABLE 3: Comparison of the composition of different networks.

Model	Layers	Model size (MB)
SNN1 [16]	1	0.376
CNN3 [8]	3	0.0094
LeNet5 [26]	5	0.239
Improved AlexNet8 [27]	8	99.4
VGG16 [28]	16	512
ResNet18 [29]	18	42.7
The proposed SCNN	8	4.58

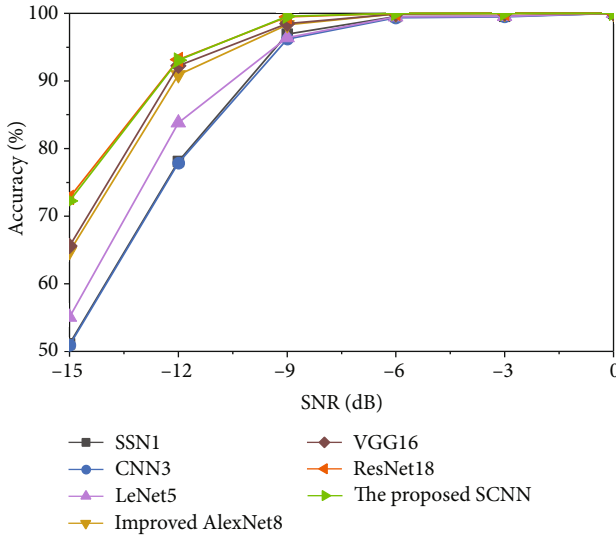


FIGURE 7: Comparison of the recognition accuracy of different networks for signals at different SNRs.

$$\frac{\partial E_{\text{total}}}{\partial \omega} = \frac{\partial E_{\text{total}}}{\partial S} \frac{\partial S}{\partial U} \frac{\partial U}{\partial \omega}, \quad (10)$$

where  $S$  denotes the output pulse and  $U$  denotes the membrane potential. As we know from equation (7),  $\partial S/\partial U$  in equation (10) cannot be derived since  $S(t)$  is a step function. The weights cannot be updated directly by gradient descent. Therefore, the method of surrogate gradient is proposed to train the SCNN. The commonly used surrogate functions are the sigmoid function and the Atan function. The approximation effects of the two functions are similar. The Atan function can be expressed as in equation (11).

$$f_A(x) = \frac{1}{\pi} \arctan\left(\frac{\pi}{2}\alpha x\right) + \frac{1}{2}. \quad (11)$$

The approximation effect of the Atan function on the step function under different  $\alpha$  is shown in Figure 4.  $\alpha$  can be regarded as the gradient factor. The larger its value, the larger the gradient is and the closer it is to the step function. However, too large  $\alpha$  tends to gradient explosion at the threshold  $x=0$  and the gradient disappears far from the threshold, which finally leads to the weights not being updated in training. In this paper, the Atan function is used

as the surrogate function. We analyze the effect of  $\alpha$  to get its best value for optimal network performance.

### 3. Results and Discussion

In order to verify the advantages of the proposed SCNN in recognizing signals at low SNRs, 800 time-frequency maps of electromagnetic signals are generated for every 3 dB interval in the SNR range from 0 dB to -15 dB. 600 time-frequency maps are used for training, and 200 time-frequency maps are used for testing. To reduce computational effort, we used grayed-out time-frequency maps for training and testing. The time-frequency maps are scaled to a size of  $32 \times 32$  using bicubic interpolation. Tables 1 and 2 show the signals' parameters and simulation environment, respectively. 64 time-frequency maps are taken as a batch. The learning rate is 0.1 at first and then changes by cosine annealing. Finally, the SCNN is optimized with the SGD optimizer.

**3.1. Impact of the Time Step.** As the SCNN accumulates pulses over time, its performance is affected by the length of the accumulation time. If the accumulation time is too short, the spiking neurons cannot be activated to fuse temporal and spatial information. If the accumulation time is too long, the network becomes very deep in time. The computational effort and inference delay of the SCNN are too much to meet the real-time requirement for signal recognition. In this paper, we should first determine an optimal time step  $T$  for accumulating pulses. Theoretically,  $T$  only affects the depth of the SCNN in time. And the gradient factor  $\alpha$  only affects the backpropagation of inaccuracies. Actually, numerous experiments also show that  $T$  and  $\alpha$  independently affect the performance of the SCNN. As a result, the optimal values of  $T$  and  $\alpha$  can be obtained by controlling the variables  $T$  and  $\alpha$ . First,  $\alpha$  is kept unchanged.  $T$  is changed from 2 to 14. When recognizing 6 signals from 0 dB to -15 dB, the accuracy trends with iterations are consistent for the same  $T$  at different SNRs. Only the results of  $\alpha = 2$  at -12 dB are shown in Figure 5 for limited pages. The accuracy of the SCNN can reach the maximum value and remains relatively stable after 40 iterations when  $T = 10$  and  $T = 12$ . Since an increase in  $T$  will lead the increasing of delay,  $T = 10$  is chosen as the optimal time step.

**3.2. Impact of the Gradient Factor.** Since  $T$  and  $\alpha$  independently affect the performance of the SCNN, we can keep the time step  $T = 10$  as we have gotten. The gradient factor  $\alpha$  is changed from 2 to 8. When recognizing 6 signals from 0 dB to -15 dB, the trends of accuracy with iterations are consistent for the same  $\alpha$  at different SNRs. Only the results at -12 dB are shown in Figure 6 for limited pages. As shown in Figure 6, the convergence of the SCNN worsens with increasing  $\alpha$ . It can be seen that the recognition accuracy is the highest and most stable when  $\alpha = 2$ . And  $\alpha = 2$  is finally adopted.

**3.3. Result Comparison of Different Networks.** Six electromagnetic signals at SNRs from 0 dB to -15 dB are identified with the SCNN when  $T = 10$  and  $\alpha = 2$ . Other simulation

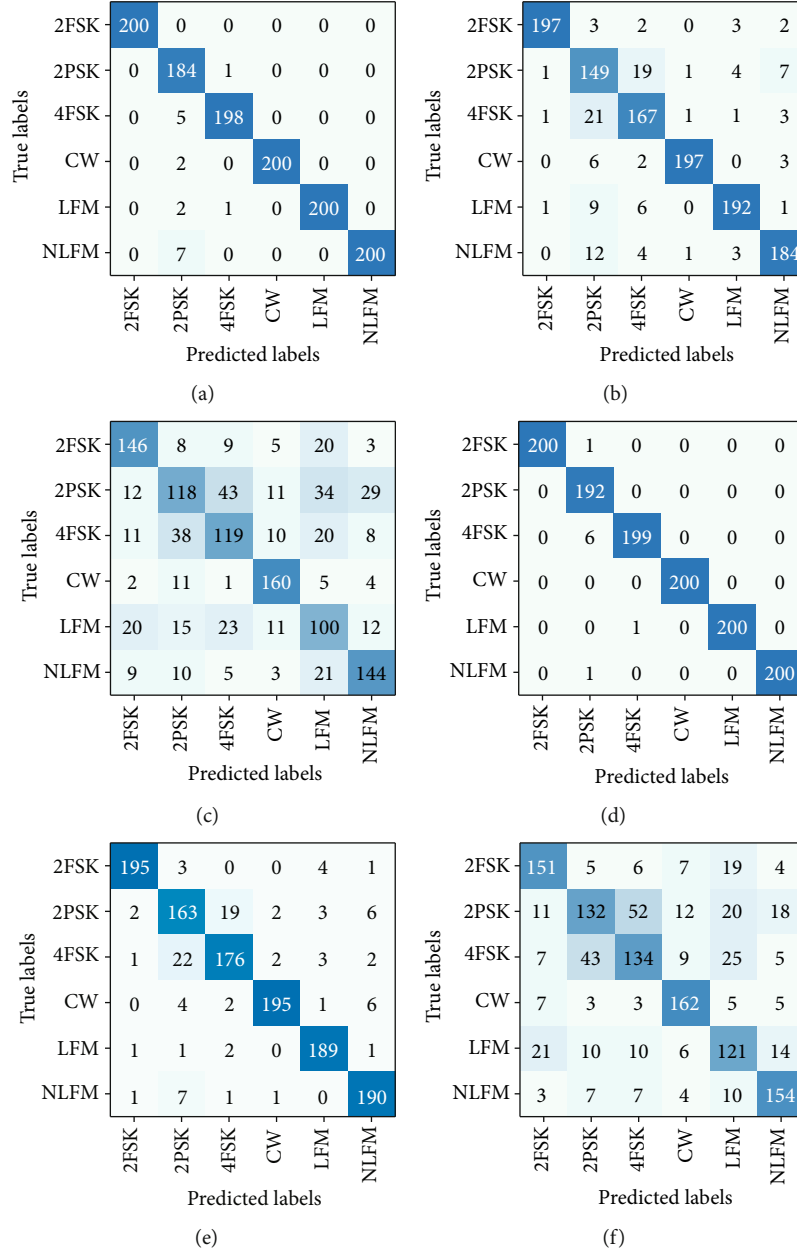


FIGURE 8: Confusion matrices of VGG16 and SCNN for 6 electromagnetic signals at -9 dB/-12 dB/-15 dB. (a) VGG16 tested at -9 dB. (b) VGG16 tested at -12 dB. (c) VGG16 tested at -15 dB. (d) SCNN tested at -9 dB. (e) SCNN tested at -12 dB. (f) SCNN tested at -15 dB.

settings are shown at the beginning of Section 3. Table 3 shows the composition and size of different networks. SNN1 is a single-layer SNN consisting of the temptron neurons [16]. CNN3 is a 3-layer CNN proposed by Ye et al. [8]. 5-layer LeNet5 is used by Guo et al. [26]. The improved AlexNet8 is proposed by Yang et al. [27]. VGG16 is used by Li and Zhu [28]. Furthermore, as the residual structure gives better performance of deep CNNs, we make a further comparison with an 18-layer ResNet18 with a residual structure [29].

As the recognition accuracy of different networks shown in Figure 7, it is found that the proposed SCNN and ResNet18 own better recognition accuracy among all the models at different SNRs. The reason that ResNet18 per-

forms well is because of the inclusion of the residual structure [29]. Although the performance of ResNet18 is comparable to that of the SCNN, the ResNet18's model size is about 9 times of the proposed SCNN, which has been shown in Table 3. In Figure 7, we can also find out that the recognition accuracy of CNN3 is relatively low due to its simple structure. With the increasing of depth, the fit ability of CNN-based models is enhanced, resulting in the increasing of recognition accuracy for the six signals at different SNRs. Even so, the recognition accuracy of CNN-based models with a large number of layers, such as VGG16, is still lower than that of the proposed SCNN. The outstanding performance of the SCNN compared to CNNs is attributed to its accumulated information in time which

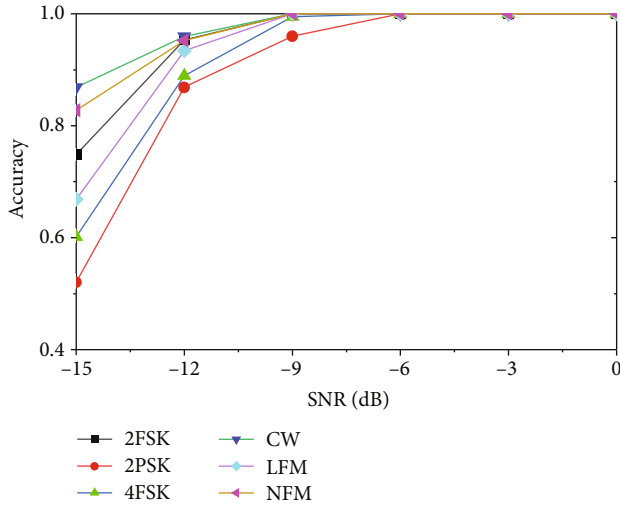


FIGURE 9: Recognition accuracy of the SCNN for each signal at different SNRs.

TABLE 4: Recognition accuracy of SCNN with different training and testing SNRs.

SNR (dB)	Accuracy (%)			
	0 (tested)	-3 (tested)	-6 (tested)	-9 (tested)
0 (trained)	100	100	91.25	50.66
-3 (trained)	100	100	98.08	60.08
-6 (trained)	100	100	99.83	90.58
-9 (trained)	100	100	99.91	99.16

compensates for its lack of spatial depth. The fusion of temporal and spatial information allows the SCNN to have good antinoise ability.

Based on the previous analysis, we know that VGG16 has the best performance for all CNNs without a residual structure. To further illustrate the advantages of the SCNN over ordinary CNNs without a residual structure, the confusion matrices of VGG16 and the SCNN are given in Figure 8. As the SNR decreases, the energy of noise covers the energy of signals. The energy distribution of different signals becomes blurred in the time-frequency maps, leading to a decrease in the recognition accuracy of both the VGG16 and SCNN for various signals. But in all, the overall recognition accuracy of the SCNN is still higher than that of the VGG16.

Figure 9 shows the recognition accuracy of the SCNN for each signal at different SNRs, in which we can see that the recognition of 2PSK is most susceptible to noise. This is because the energy of frequency components of 2PSK is relatively dispersed compared with other signals, which has been shown in Figure 2.

**3.4. Analysis of Generalization Performance for Noise.** In order to verify the generalization performance of the SCNN for noise, 6 electromagnetic signals at 0 dB, -3 dB, -6 dB, and -9 dB are used for training and testing. As shown in Table 4, the recognition accuracy is very high when the

SNRs of trained signals are lower than or comparable with those of tested signals. Only when the SNRs of tested signals are much lower than those of trained signals will the recognition accuracy decline. So, it indicates that the SCNN has good generalization performance for noise in a certain SNR range.

## 4. Conclusions

The SCNN proposed in this paper utilizes the convolutional layer as visual perception. The LIF neurons and IF neurons are used to construct the convolutional layers and fully connected layers of the SCNN, respectively. The feature maps are converted into pulses to obtain temporal information, which enables the SCNN to perform well in the recognition of electromagnetic signals. And the outstanding recognition performance of the SCNN for electromagnetic signals at low SNRs is verified through the recognition of 6 signals at different SNRs. As a third-generation neural network, information transmission in the form of pulses in SNNs has stronger biological interpretability. The low power consumption of SNNs can get greater benefits in signal recognition.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the State Key Laboratory of CEMEE (Grant no. CEMEE2022K0102A).

## References

- [1] J. S. Mu, Y. H. Tan, D. L. Xie, F. Zhang, and X. Jing, "CNN and DCGAN for spectrum sensors over rayleigh fading channel," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9970600, 12 pages, 2021.
- [2] G. M. Wang, S. W. Chen, J. Huang, X. Qin, and J. J. Yuan, "Radar emitter sorting and recognition based on multi-synchrosqueezing transform," *Modern Radar*, vol. 42, no. 3, pp. 49–56, 2020.
- [3] X. F. Li, F. W. Dong, S. Zhang, and W. Guo, "A survey on deep learning techniques in wireless signal recognition," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5629572, 12 pages, 2019.
- [4] X. Qin, X. Zha, J. Huang, and L. Luo, "Radar waveform recognition based on deep residual network," in *The 8th IEEE Joint International Information Technology and Artificial Intelligence Conference*, pp. 892–896, Chongqing, China, 2019.
- [5] T. Huynh-The, C. H. Hua, V. S. Doan, Q. V. Pham, and D. S. Kim, "Accurate deep CNN-based waveform recognition for intelligent radar systems," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2938–2942, 2021.
- [6] R. Ahmed, Y. Chen, and B. Hassan, "Deep learning-driven opportunistic spectrum access (OSA) framework for cognitive



- 5G and beyond 5G (B5G) networks,” *Ad Hoc Networks*, vol. 123, pp. 102632–102645, 2021.
- [7] M. Xu, J. Hou, P. Wu, Y. Liu, and Z. Lv, “Convolutional neural networks based on time frequency characteristics for modulation classification,” *Journal of Computer Science and Technology*, vol. 47, no. 2, pp. 175–179, 2020.
- [8] W. Q. Ye, Z. F. Yu, H. B. Wang, and K. Zhang, “Recognition algorithm of emitter signals based on CNN,” *Computer Simulation*, vol. 36, no. 9, pp. 33–37, 2019.
- [9] Y. Yao and Z. H. Wang, “Radar signal recognition based on time-frequency and CNN preprocessing,” *Journal of Detection & Control*, vol. 40, no. 6, pp. 99–105, 2018.
- [10] Y. H. Xiao, L. Wang, and Y. X. Guo, “Radar signal modulation type recognition based on denoising convolutional neural network,” *Journal of Electronics & Information Technology*, vol. 43, no. 8, pp. 2300–2307, 2021.
- [11] P. U. Yun-Wei, G. U. Jiang, L. I. Shou-Shou, and W. U. Hai-Xiao, “A recognition method for radar emitter signals based on convolutional neural network with multiple learning units,” *Journal of Beijing University of Posts and Telecommunications*, vol. 44, no. 6, pp. 74–82, 2021.
- [12] C. X. Xie, L. M. Zhang, and Z. G. Zhong, “Radar signal recognition based on time frequency feature extraction and residual neural network,” *Systems Engineering and Electronics*, vol. 43, no. 4, pp. 917–926, 2021.
- [13] P. Y. Cao, C. Z. Yang, C. Z. Chen, Z. S. Chen, L. Wang, and L. M. Shi, “Radar signal recognition method based on deep residual shrinkage attention network “In press,”” *Systems Engineering and Electronics*, 2022.
- [14] N. Wang, Y. Liu, L. Ma, Y. Yang, and H. Wang, “Multidimensional CNN-LSTM network for automatic modulation classification,” *Electronics*, vol. 10, no. 14, p. 1649, 2021.
- [15] L. M. Shi, C. Z. Yang, M. L. Wang, and B. Xu, “Recognition method of radar signal modulation method based on deep network,” *Journal of Ordnance Equipment Engineering*, vol. 42, no. 6, 2021.
- [16] W. Li, W. G. Zhu, and B. K. Zhu, “Modulation pattern recognition of radar emitter based on spiking neural network,” *Telecommunication Engineering*, vol. 62, no. 1, pp. 11–16, 2022.
- [17] X. Yang, Z. X. Zhang, W. P. Zhu, S. Yu, L. Liu, and N. Wu, “Deterministic conversion rule for CNNs to efficient spiking convolutional neural networks,” *Science China (Information Sciences)*, vol. 63, no. 2, pp. 196–214, 2020.
- [18] Y. Shang, Y. Li, F. You, and R. Zhao, “Conversion-based approach to obtain an SNN construction,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 30, no. 11, pp. 1801–1818, 2020.
- [19] S. Takuya, R. Y. Zhang, and Y. Nakashima, “Training low-latency spiking neural network through knowledge distillation,” in *IEEE Symposium on Low-Power and High-Speed Chips (IEEE COOL CHIPS)*, pp. 14–16, Tokyo, Japan, 2021.
- [20] C. Zhang and F. Z. Tang, “Self-adaptive coding for spiking neural network,” *Application Research of Computers*, vol. 39, no. 2, pp. 593–597, 2022.
- [21] M. Y. Xu, “Spiking neural network training algorithm based on gradient descent,” *Modern Computer*, vol. 27, no. 35, pp. 1–11, 2021.
- [22] Y. F. Hu, G. Q. Li, Y. J. Wu, and L. Deng, “Spiking neural networks: a survey on recent advances and new directions,” *Control and Decision*, vol. 36, no. 1, pp. 1–26, 2021.
- [23] T. L. Zhang and B. Xu, “Research advances and perspectives on spiking neural networks,” *Chinese Journal of Computers*, vol. 44, no. 9, pp. 1767–1785, 2021.
- [24] X. Cheng, Y. Hao, J. Xu, and B. Xu, “LISNN: improving spiking neural networks with lateral interactions for robust object recognition,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1519–1525, Yokohama, Japan, 2020.
- [25] Y. P. Liu, C. Wang, and H. Y. Xia, “Application progress of time-frequency analysis for lidar,” *Laser & Optoelectronics Progress*, vol. 55, no. 12, pp. 68–83, 2018.
- [26] P. P. Guo and L. Y. Wu, “LPI radar signal recognition with convolution feature and discrimination dictionary learning,” *Acta Armamentarii*, vol. 40, no. 9, pp. 1881–1889, 2019.
- [27] J. Yang and H. Zhang, “LPI radar signal recognition based on improved Alex Net,” *Modern Electronic Technology*, vol. 43, no. 5, pp. 57–60, 2020.
- [28] K. Li and W. G. Zhu, “Radar emitter identification method based on deep learning,” *Electronic Design Engineering*, vol. 28, no. 12, pp. 99–104, 2020.
- [29] C. Dong, C. W. Zhuang, Y. Gao et al., “Mixed communication signal recognition based on residual neural network,” *Radio Engineering*, vol. 50, no. 9, pp. 727–731, 2020.