Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9970600, 12 pages
https://doi.org/10.1155/2021/9970600

Research Article

CNN and DCGAN for Spectrum Sensors over Rayleigh Fading Channel

Junsheng Mu,[1] **Youheng Tan**,[2] **Dongliang Xie**,[1] **Fangpei Zhang**,[3] **and Xiaojun Jing**[2]

[1]*The State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China*
[2]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*Information Science Research Institute of China Electronics Technology Group Corporation, Beijing 100876, China*

Correspondence should be addressed to Xiaojun Jing; jxiaojun@bupt.edu.cn

Received 16 March 2021; Accepted 15 July 2021; Published 10 August 2021

Academic Editor: Ding Xu

Spectrum sensing (SS) has attracted much attention in the field of Internet of things (IoT) due to its capacity of discovering the available spectrum holes and improving the spectrum efficiency. However, the limited sensing time leads to insufficient sampling data due to the tradeoff between sensing time and communication time. In this paper, deep learning (DL) is applied to SS to achieve a better balance between sensing performance and sensing complexity. More specifically, the two-dimensional dataset of the received signal is established under the various signal-to-noise ratio (SNR) conditions firstly. Then, an improved deep convolutional generative adversarial network (DCGAN) is proposed to expand the training set so as to address the issue of data shortage. Moreover, the LeNet, AlexNet, VGG-16, and the proposed CNN-1 network are trained on the expanded dataset. Finally, the false alarm probability and detection probability are obtained under the various SNR scenarios to validate the effectiveness of the proposed schemes. Simulation results state that the sensing accuracy of the proposed scheme is greatly improved.

1. Introduction

In recent years, the spectrum resource has been more and more scarce due to the great demand for wireless communication, Internet of Things (IoT), Artificial Intelligence (AI) [1–3], etc. One of the most important issues of wireless communication technology is to improve its spectrum efficiency in the near future. As a possible scheme to improve spectrum efficiency, cognitive radio (CR) [4] has attracted much attention.

The core idea behind CR is to realize dynamic spectrum allocation (DSA) and spectrum sharing by spectrum sensing and the intelligent learning ability of the system [5]. The most important technology behind CR is to periodically monitor the absence or the presence of the registered users within the observed bands, named spectrum sensing (SS) [6]. In SS, the registered users are the primary users (PU) of the observed bands and have the priority to the spectrum. The purpose of CR is to opportunistically access the registered spectrum when PU is absent. As a result, the cognitive users are the secondary users (SU). Once the PU is back, the SU will release the spectrum at once and wait for the other opportunity.

Classical SS contains matched filtering, energy detector (ED) [7], cyclic spectrum detection [8], covariance matrix detection [9], etc. The sensing performance of the matched filtering is optimal if the prior knowledge of the primary signal is known in advance. ED is the optimal blind detector considering both sensing performance and sensing complexity. However, it suffers from noise uncertainty under the low signal-to-noise ratio (SNR) regimes. The sensing performance of cyclic spectrum detection and covariance matrix detection is improved in the low-SNR case compared with ED at the expense of a higher complexity. However, these traditional SS schemes either have poor performance or have high complexity.

Recently, with the wide application of the deep learning (DL) in the field of computer vision [10], the wireless communication based on DL has been a hot topic [11–13]. The essence of DL is to provide a method of automatically learning

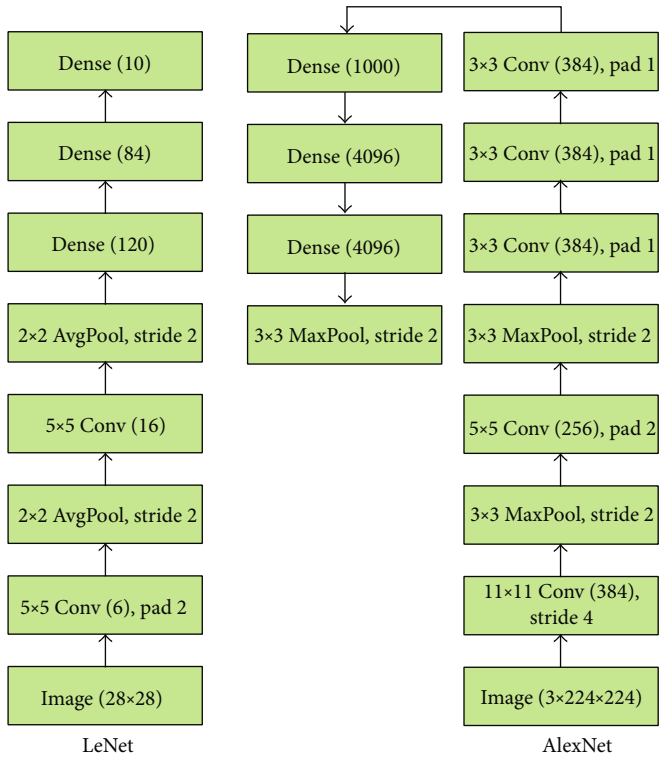

Figure 1: The network architecture of LeNet and AlexNet.

pattern features and combine the features, thus, reducing the incompleteness caused by artificial design features. In [14], a stacked autoencoder based spectrum sensing approach (SAE-SS) was proposed to relieve the influence from carrier frequency offset (CFO), timing delay, and noise uncertainty. A deep learning based signal detector was considered to exploit the underlying structural information of the modulated signals in [15]. The transfer learning strategies were used in [16] to improve the performance for real-world signals. In [17], the convolutional neural network- (CNN-) long short term memory network (LSTM) detector was proposed to extract the spatial and temporal features of the input.

Motivated by the mentioned above, DL is applied to SS in this paper, where the covariance matrix of the received signal is converted into the true color picture. Then, an improved deep convolutional generative adversarial network (DCGAN) is proposed to expand the training set for the issue of data shortage.

After that, the LeNet [18], AlexNet [19], VGG-16 [20], and a novel network are trained based on the extended data. Finally, the simulations are made to validate the effectiveness of the proposed schemes. The main contributions of this paper are concluded as follows.

(1) The two-dimensional dataset of the received signal is established under the various SNR conditions. Each SNR contains 4000 samples from -10 dB to 2 dB

(2) An improved DCGAN network is proposed to expand the obtained two-dimensional dataset. In the expanded dataset, each SNR contains 8000 samples

(3) The LeNet, AlexNet, and VGG-16 networks are trained on the expanded dataset. The corresponding false alarm probability and detection probability are given under the various SNR scenarios

(4) Based on the sensing performance of the LeNet, AlexNet, and VGG-16 networks, an improved network is provided in this paper to balance the sensing performance and the sensing complexity

The reminder of the paper is organized as follows. Section 2 introduces the related work and gives the system model. In Section 3, the improved DCGAN scheme is discussed to solve the issue of data shortage. The SS with the LeNet, AlexNet, VGG-16, and an improved network is conducted. Finally, conclusions are drawn in Section 5.

2. Related Work

In this section, three classical convolutional neural networks (CNN) and the deep convolutional generative adversarial network (DCGAN) [21] are reviewed. In addition, the system model of this paper is provided.

2.1. CNN. CNN is a feed-forward neural network. This network model uses a gradient descent method to minimize the loss function and reversely adjusts the weight parameters

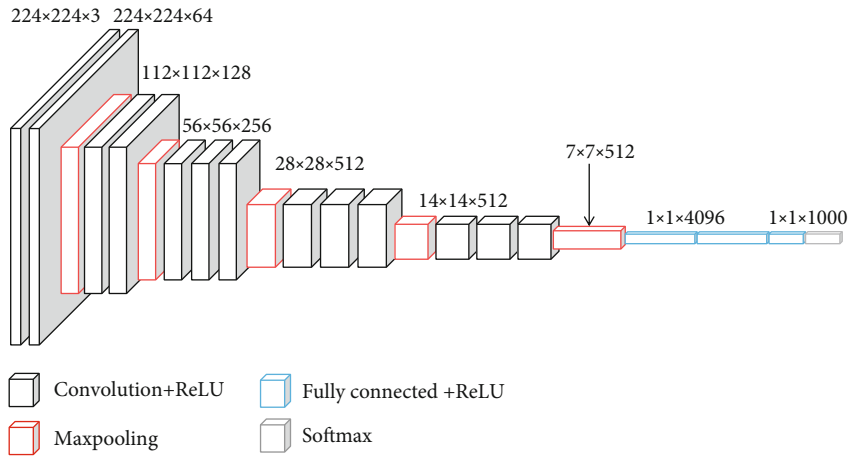

Figure 2: The network architecture of VGG-16.

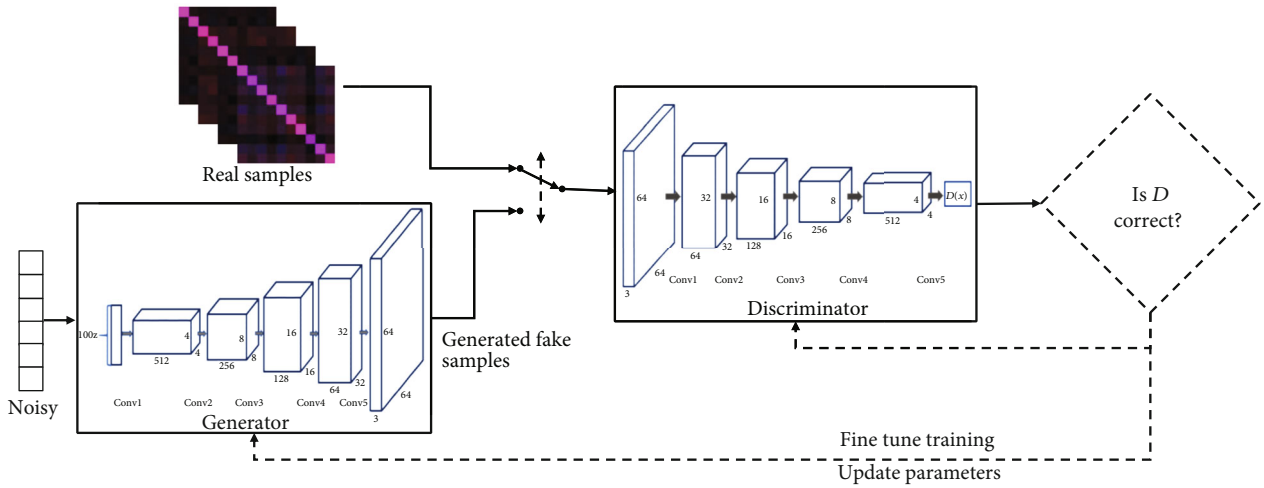

Figure 3: Basic framework of DCGAN.

in the network layer by layer. Three classical CNN networks are considered in this paper, LeNet, AlexNet, and VGG-16.

As it is shown in Figure 1, the LeNet network contains three convolutional layers with the size 5×5, two pooling layers with the size 2×2, and three fully connected layers [18]. The AlexNet network contains five convolutional layers, three pooling layers with the size 3×3, and three fully connected layers [19].

As it is shown in Figure 2, the VGG-16 network contains 13 convolutional layers with the size 3×3 and three fully connected layers [20]. The maximum polling is considered in VGG-16 network with the size 2×2. The first two pooling layers are followed by two convolutional layers while the rest pooling layers are followed by three convolutional layers.

2.2. DCGAN. The generative adversarial networks (GAN) [21] have attracted wide attention in the field of machine learning because of its great potential to imitate high-dimensional and complex real data. For scenarios where there is a lack of data, it can be used to generate more sample data. In order to solve the problem of high acquisition cost of training set samples, this paper utilizes GAN to generate more training set samples. Generative adversarial networks include generating network (Generator) and discriminating network (Discriminator). The generating network learns the real data distribution to generate new data under the guidance of the discriminating network. Deep convolutional generative adversarial networks are one of the more effective and stable networks based on GAN. The basic framework of DCGAN [22] is shown in Figure 3.

Assume the input of the Generator is the random Gaussian noise z and its output is the fake sample $G(z)$. The true sample x and fake sample $G(z)$ are input to the Discriminator, respectively, and the corresponding outputs are $D(x)$ and $D(\mathrm{G}(\mathrm{z}))$.$D(x)$ denotes the probability that the input x of the Discriminator is a real sample. The Discriminator is to make $D(x)$ tend to 1 and $D(\mathrm{G}(\mathrm{z}))$ tend to 0. At the same time, the Generator is to make $D(\mathrm{G}(\mathrm{z}))$ close to 1. The loss function of DCGAN is shown as

$$L(\mathrm{D}, \mathrm{G}) = E_{x \sim P_r}[\log D(x)] + E_{x \sim P_g}[\log (1 - D(x))], \quad (1)$$

where P_r represents the real sample distribution and P_g represents the fake sample distribution.

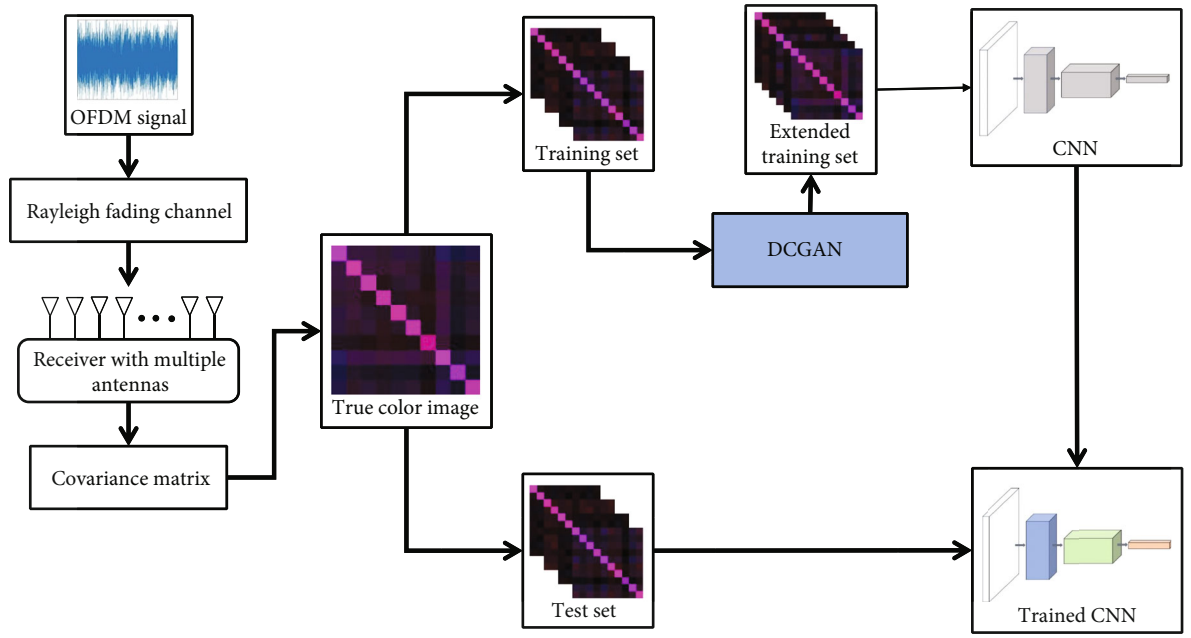

Figure 4: System model.

The objective functions of the Discriminator and the Generator are, respectively, written as

$$\max_{D} L(D, G), \tag{2}$$

$$\max_{G} \max_{D} L(D, G). \tag{3}$$

(3) can be further rewritten as

$$\max_{G} \max_{D} E_{x\sim P_g}[\log(1 - D(x))]. \tag{4}$$

2.3. System Model. In this paper, the signal is received by a multiantenna system, and then the covariance matrix of the signal is calculated. After that, the covariance matrix is transformed into a true color graph as a data set. The system model of this paper is shown in Figure 4.

An M-element antenna system is considered to receive the signals based on N observation vectors. Let $s_i(\mathrm{n})$, $n = 0, 1, \cdots, N-1$ denote the $n-$th discrete-time sample at the $i-$th antenna. Generally, the spectrum sensing can be regarded as a binary classification [21],

$$s_i(\mathrm{n}) = \begin{cases} w_i(\mathrm{n}), & H_0, \\ x_i(\mathrm{n}) + w_i(\mathrm{n}), & H_1, \end{cases} \tag{5}$$

where $w_i(\mathrm{n})$ denotes the background noise and $x_i(\mathrm{n})$ denotes primary signal vector with the Rayleigh fading [7]. H_0 and H_1, respectively, signify the absence and the presence of PU.

Let $s_i = [s_i(0), s_i(1), \cdots s_i(N-1)]$ denote the sampling sequence at the $i-$th antenna and its average can be expressed as

$$s_{i-a} = \frac{1}{N} \sum_{n=0}^{N-1} s_i(n). \tag{6}$$

The time series matrix of the received signal S can be formulated as

$$S = \begin{bmatrix} s_1(0) - s_{1-a} & s_1(1) - s_{1-a} & \cdots & s_1(N) - s_{1-a} \\ s_2(0) - s_{2-a} & s_2(1) - s_{1-a} & \cdots & s_2(N) - s_{1-a} \\ \vdots & \vdots & \cdots & \vdots \\ s_M(0) - s_{M-a} & s_M(1) - s_{M-a} & \cdots & s_M(N) - s_{M-a} \end{bmatrix}. \tag{7}$$

The corresponding covariance matrix is

$$R(N) = \frac{1}{N} S \times S^H. \tag{8}$$

For the real part and the imaginary part of S, the covariance matrix is considered as the two channels of the true color image, together with a zero matrix as the third channel of the true color image.

The samples are divided into training set and test set. Then, the training set is expanded by DCGAN. After that, CNN is trained by the expanded training set. Finally, the sensing performance is validated by the test set.

The false alarm probability and the detection probability can be formulated as

$$P_d = P\{\varphi[R(N)] > \lambda | H_1\}, \tag{9}$$

$$P_f = P\{\varphi[R(N)] > \lambda | H_0\}, \tag{10}$$

where $\varphi[R(N)]$ denotes the feature extraction operation of the proposed network and λ denotes the sensing threshold.

3. Data Enhancement with DCGAN

In this section, the data enhancement scheme with DCGAN is discussed and an improved DCGAN scheme is proposed, where the python3.7 and pytorch1.5 machine learning libraries are used to implement generative adversarial network and convolutional neural network. The hardware CPU is Inter(R) Core(TM) i5-6300HQ and the GPU is NVIDIA GeForce GTX 960 M.

3.1. Data Generation. In this subsection, the original training set and test set are generated, where the OFDM signal [23] is considered as the primary signal, and the Rayleigh fading [24] is regarded as the propagation channel. The size of the sampling covariance matrix is $10 \times 10 \times 2$. The number of antennas in the multiantenna system used in this paper is $M = 10$. The channel number of a true color image is 3, where the channel number of the sampling covariance matrix is 2 (the real part and the imaginary part), and the third channel is set to a zero matrix. The matrix size depends on the antenna number of the multiantenna system.

In the process of data generation, 8 datasets are generated, whose SNR varies from -10 dB to 2 dB with the step 2 dB. Each dataset is divided into two parts, H_0 and H_1. For the H_0 part, the real and imaginary parts are, respectively, sampled to obtain 3000 sets of data, where each set of data sampling points $N = 1000$ and contains two matrices. Then, this set of data is a $10 \times 1000 \times 2$ sampling time series matrix. Calculate the sampling covariance matrix of each matrix according to (7). Then, 3000 sets of dual-channel sampling covariance matrices can be obtained. After that, these 3000 sets of data are randomly sorted and converted into true color images, where the first channel of the true color image is the first channel of the sampling covariance matrix, the second channel is the zero matrix, and the third channel is the second channel of the sampling covariance matrix. Finally, take the first 1000 groups as the data of the H_0 category in the test set, and take the last 2000 groups as the data of the H_0 category in the original training set.

For the H_1 part, the same operations are conducted as the H_0. As a result, the 1000 sets of test set data and 2000 sets of original training set data for the H_1 part.

Figure 5 exhibits the obtained true color image in the H_0 and H_1 case under 2 dB, where the left image corresponds with the H_0 case and the right image corresponds with the H_1 case. From Figure 5, the color of the H_0 case is very dark except for the diagonal, which indicates that the value of the corresponding two-channel sampling covariance matrix is very small. While in the H_1 case, the color of the image is uneven and somewhat chaotic, which indicates that the corresponding two-channel sampling covariance matrix values are also chaotic.

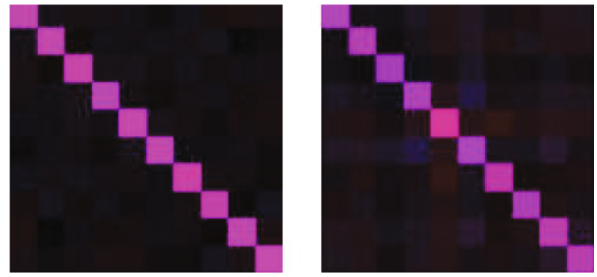

Figure 5: True color image in the H_0 and H_1 cases under 2 dB.

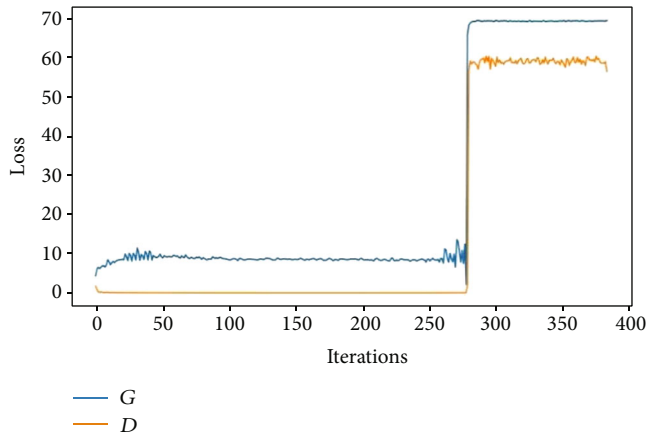

Figure 6: The loss of the training.

3.2. Data Enhancement. The sampled data is enhanced in this subsection based on the classical DCGAN scheme, where the sampled data is doubled, from 4000 images to 8000 images for each SNR.

In Figure 6, the loss of the training with the iterations is exhibited, where G denotes the Generator, D denotes the Discriminator, and the number of training cycles is 16. According to Figure 6, when it runs to the 10th cycle, a large loss occurs due to the gradient explosion. As a result, how to effectively reduce the loss determines the quality of data enhancement.

In Figure 7, the initial weights of the convolution kernel are adjusted, from the Gaussian distribution with the mean 0 and the variance 0.02 to the Gaussian distribution with the mean 0.05 and the variance 0.02. As a result, the loss function is enlarged, and the trend of the loss function can be better observed. From Figure 7, to increase the mean of the initial weight will not only increase the loss function value but also slow down the training speed and reduce the training gradient.

Figures 8 and 9, respectively, exhibit the training result with the mean of initial weight 0.05 and 0.02, where the left image denotes the original image and the right image denotes the generated image with DCGAN. From Figures 8 and 9, the smaller the mean of initial weight is, the better the generated image quality is. It can be seen that the initial weight average still has a great influence on the convergence speed of the model. The larger the initial weight of the convolution kernel,

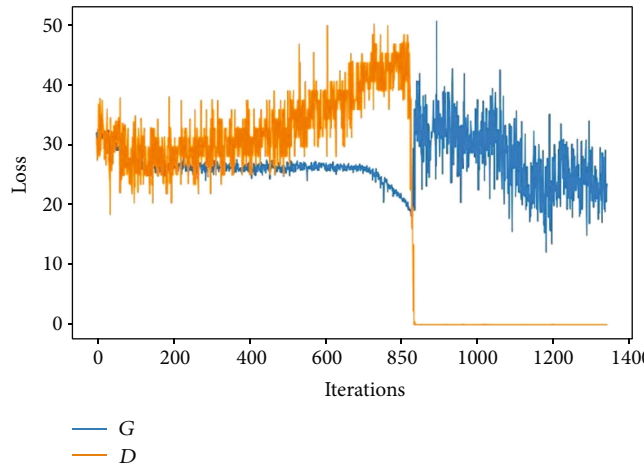

Figure 7: The training loss when the mean of initial weight is 0.05.

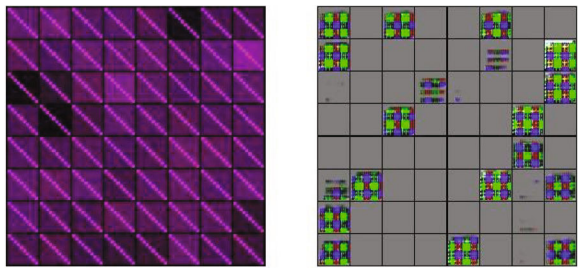

Figure 8: 82 times training result with the mean of initial weight 0.05.

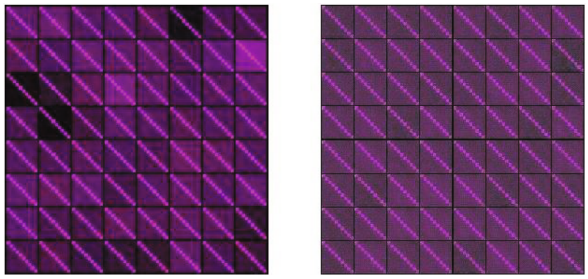

Figure 9: 82 times training result with the mean of initial weight 0.02.

the slower the convergence speed may be, and the slower the gradient will be. But at the same time, more useful information can be obtained.

3.3. Improved DCGAN. According to the discussion mentioned above, to increase the mean of the initial weight reduces the training gradient. However, it sacrifices the generated image quality. The improved DCGAN scheme is made to solve the above issue in this subsection.

The change of $D(G(z))$ before and after the gradient explosion is shown in Figure 10 with the mean of initial weight 0. From Figure 10, when the gradient explosion occurs, the loss functions of the generating network and the discriminating network become extremely large while $D(x)$ and $D(G(z))$ are both 0. At this point, both networks will be collapsed. Motivated by this, an adjustment scheme is proposed as it is shown in Table 1.

[0/16]	Loss_D: 16918	Loss_G: 4.3525	$D(x)$: 0.6500	$D(G(x))$: 0.6606/0.175
[2/16]	Loss_D: 0.0087	Loss_G: 9.4516	$D(x)$: 0.9994	$D(G(x))$: 0.0081/0.0001
[4/16]	Loss_D: 0.0025	Loss_G: 8.8306	$D(x)$: 0.9996	$D(G(x))$: 0.0020/0.0002
[6/16]	Loss_D: 0.0028	Loss_G: 8.6410	$D(x)$: 0.9996	$D(G(x))$: 0.0024/0.0002
[8/16]	Loss_D: 0.0034	Loss_G: 8.3489	$D(x)$: 0.9988	$D(G(x))$: 0.0022/0.0003
[10/16]	Loss_D: 58.5886	Loss_G: 69.2882	$D(x)$: 0.0000	$D(G(x))$: 0.0000/0.0000
[12/16]	Loss_D: 58.5617	Loss_G: 69.2911	$D(x)$: 0.0000	$D(G(x))$: 0.0000/0.0000

Figure 10: The operation status caparisons before and after the gradient explosion with the mean of initial weight 0.

Figures 11 and 12 show the improved DCGAN result with the mean of initial weight 0 and the improved DCGAN loss with the mean of initial weight 0 when the algorithm runs to the 40th training loop. It can be seen that the generated map has been able to reach the point of being fake. Moreover, from the point of view of the loss function, since the 100th training, the generating network and the discriminating network have also shown an obvious upward and downward jitter trend, that is, adversarial evolution. At the same time, it can be concluded that the mean value of the initialization parameter weight changes from 0.2 to 0, which speeds up the convergence of the model.

4. SS with CNN Network

For SS, two factors determine the sensing performance, detection probability (PD), and false alarm probability (PFA). As a result, the PD and PFA are provided with LeNet, AlexNet, VGG-16, and the proposed CNN-1 network in this section.

4.1. LeNet. Figures 13 and 14, respectively, show the PD [25] and PFA [26] under various SNR conditions. From Figures 13 and 14, the detection performance of the LeNet network is obviously different under different SNRs. When the SNR is small, PD is small and PFA is high due to the obtained nonsignificant features. However, when SNR ≥ -4 dB, PD can be maintained at or above 0.9, and PFA is basically lower than 0.1. In addition, after data enhancement with the proposed scheme, the mean value of PD becomes higher,

Table 1: The adjustment scheme of DCGAN.

	$D(G(z))$	Emergency action
Emergency action of judgement network	$0.50 < D(G(z)) \leq 0.65$	2 extra training times
	$0.65 < D(G(z)) \leq 0.75$	3 extra training times
	$0.75 < D(G(z)) \leq 0.85$	4 extra training times
	$0.85 < D(G(z)) \leq 0.95$	5 extra training times
	$0.95 < D(G(z))$	6 extra training times
Emergency action of judgement network	$0.14 < D(G(z)) \leq 0.19$	1 extra training time
	$0.10 < D(G(z)) \leq 0.14$	2 extra training times
	$0.07 < D(G(z)) \leq 0.10$	3 extra training times
	$D(G(z)) \leq 0.07$	4 extra training times

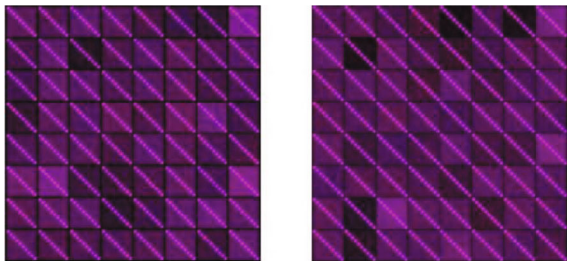

Figure 11: Improved DCGAN result with the mean of initial weight 0.

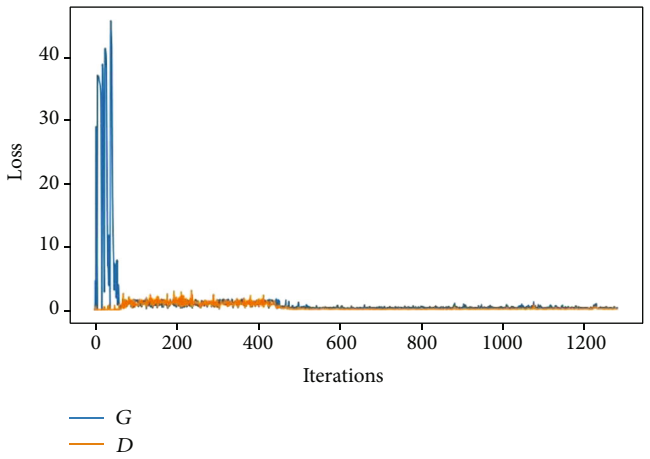

Figure 12: Improved DCGAN loss with the mean of initial weight 0.

and the maximum and minimum values of PD become significantly higher. Although the average value of PFA does not significantly decrease, but its maximum and minimum values are closer to the mean, which indicates its fluctuation range becomes smaller with the improved DCGAN.

4.2. AlexNet. Figures 15 and 16, respectively, give the PD and PFA comparisons with the AlexNet network under various SNRs, where the improved DCGAN is considered.

From Figure 15, the average value of PD is 0.95 or more, and the minimum value is 0.8 or more when $-2\text{dB} \leq \text{SNR} \leq$ 4dB. The average value of PD is close to the maximum, which means that most PD values are close to 1, and only a few are close to the minimum values. In addition, after the data enhancement, the average value of PD is also improved slightly.

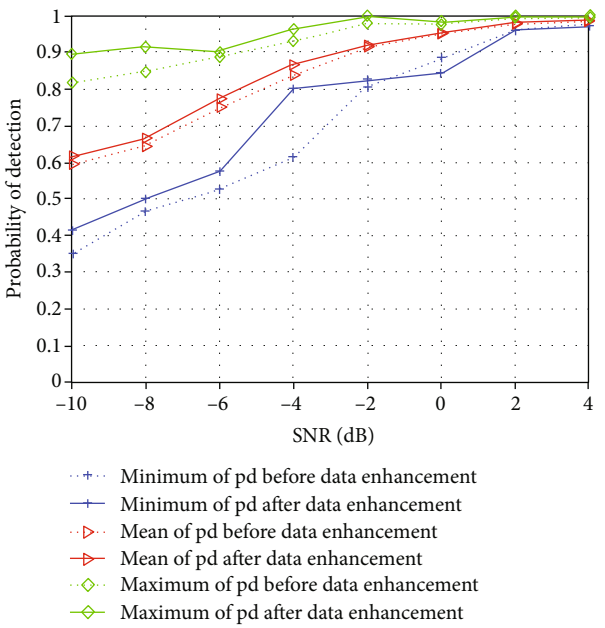

Figure 13: Detection probability with LeNet.

The PFA is provided with the AlexNet network in Figure 16. From Figure 16, the mean value of PFA is less than 0.05 and its curve almost coincides with the minimum curve. At the same time, after the data enhancement, the maximum value of PFA decreases obviously, which indicates that the vibration amplitude of PFA decreases obviously.

As a summary, when $-2\text{dB} \leq \text{SNR} \leq 4\text{dB}$, PD is close to 1 and PFA is close to 0, which indicates that the detection performance of the AlexNet is better than that of the LeNet. After the data enhancement, PD is improved slightly, and PFA has become more stable.

4.3. VGG-16. In Figures 17 and 18, the PD and PFA are, respectively, exhibited with the VGG-16 network. According

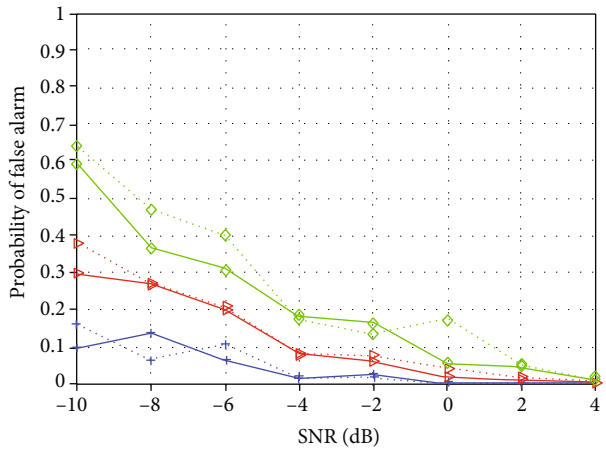

Figure 14: False alarm probability with LeNet.

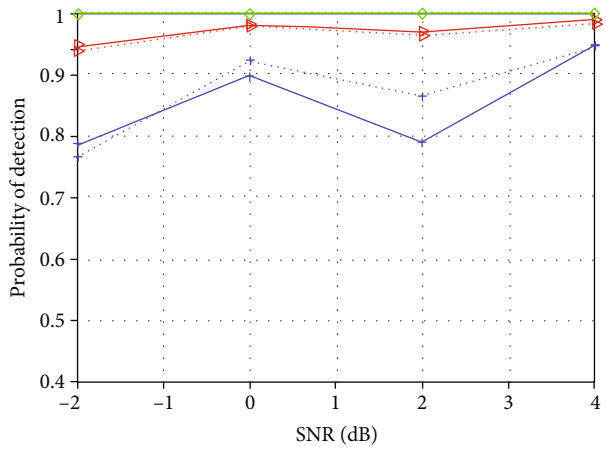

Figure 15: Detection probability with AlexNet.

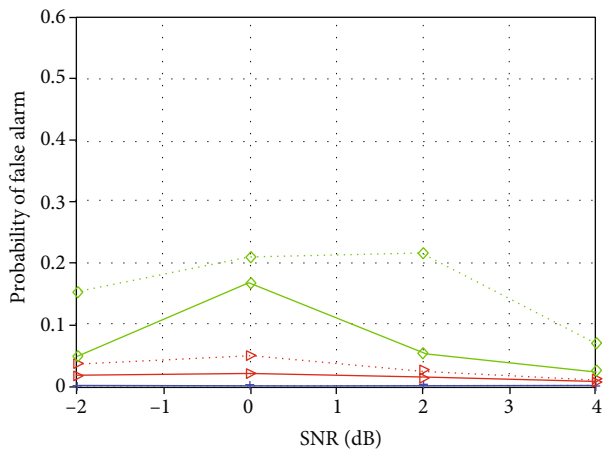

Figure 16: False alarm probability with AlexNet.

to Figure 17, when SNR = −10 dB or SNR = −4 dB, the average value of PD is improved by nearly 0.1. When SNR ≥ −4 dB, the average value of PD is above 0.9. After the data enhancement, the minimum value of PD is also significantly improved.

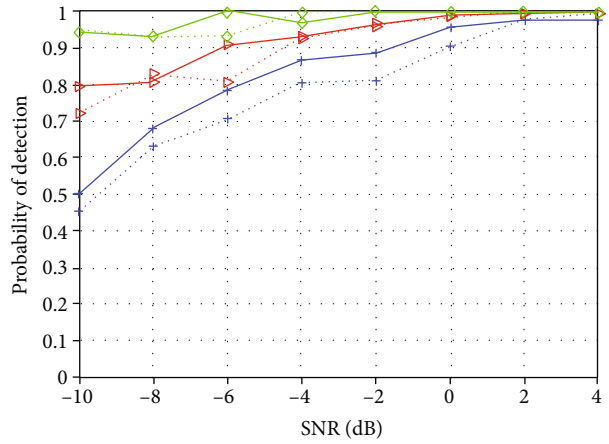

Figure 17: Detection probability with VGG-16.

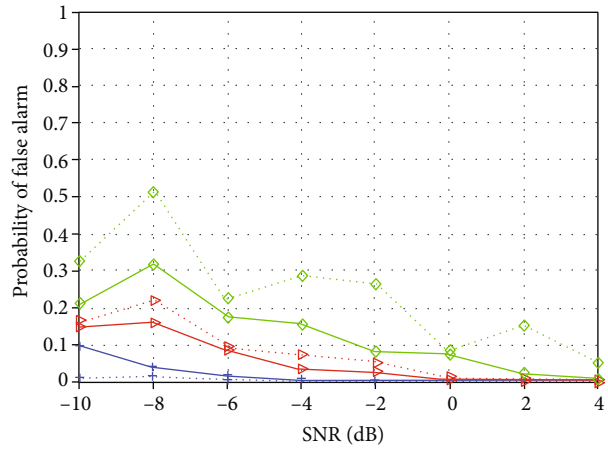

Figure 18: False alarm probability with VGG-16.

From Figure 18, the average value of PFA is less than 0.2 and decreases with the increase of SNR. After the data enhancement, the mean and maximum values of PFA decrease significantly.

To sum up, the detection performance of VGG-16 is better than that of LeNet-5 and AlexNet. In addition, the improvement of data enhancement on the performance of the model is also obvious, where the vibration amplitude of PD and PFA decreases obviously. However, the performance of VGG-16 is at the expense of computational complexity due to a large number of network parameters.

4.4. CNN-1. According to the sensing performance of the LeNet, AlexNet, and VGG-16 network, the network depth of VGG-16 is too large and the depth of Lenet-5 is too small to achieve the optimal result. Therefore, this paper designs a novel convolution neural network with the appropriate depth, named CNN-1. The network parameters of the proposed CNN-1 network are shown in Table 2.

As described in the system model, R_x is a real matrix with the dimension $M \times M \times 3$. Let $R_x(i, j, \tau)$ denote the element at position (i, j) of the τ-th dimension for R_x.

Table 2: The parameters of CNN-1.

Input: true color image (dimension: $110 \times 110 \times 3$)	
Layers	Kernel size
C_1 + ReLU	$32 \times 3 \times (3 \times 3)$
S_1: maximum pooling	2×2
C_2 + ReLU	$64 \times 32 \times (3 \times 3)$
S_2: maximum pooling	2×2
C_3 + ReLU	$128 \times 64 \times (3 \times 3)$
S_3: maximum pooling	2×2
C_4 + ReLU	$128 \times 128 \times (3 \times 3)$
S_4: maximum pooling	2×2
F_1 + ReLU	512×3200
F_2 + sigmoid	1×512
Output: score vector (dimension: 1×1)	

(1) *Convolutional Layer* C_1. C_1 contains 32 feature maps, and each feature map is gained by convolution operation with the kernel size of 3×3. Thus

$$C_1(i,j,\tau) = \varphi_{\mathrm{ReLU}}\left(\sum_{k=0}^{2}\sum_{i_0=0}^{2}\sum_{j_0}^{2} R_x(i+i_0, j+j_0, k)\right) k_\tau^{C_1} \cdot (3-i_0, 3-j_0, 2-k), \tag{11}$$

where $C_1(i,j,\tau)$ denotes the element at position (i,j) of the $\tau-$th feature map in C_1 layer, and $k_\tau^{C_1}$ denotes the kernel of the $\tau-$th feature map in C_1 layer. $\varphi_{\mathrm{ReLU}} = \max(0,x)$ denotes ReLU function, where $\max(\cdot)$ denotes the maximum value, x is the independent variable of the function.

(2) *Pooling Layer* S_1. S_1 contains 32 feature maps, and maximum pooling is conducted for each feature map

$$S_1(i,j,\tau) = \max\{C_1(2i-1, 2j-1, \tau), C_1(2i-1, 2j, \tau), C_1(2i, 2j-1, \tau), C_1(2i, 2j, \tau)\}. \tag{12}$$

(3) *Convolutional Layer* C_2. C_2 contains 64 feature maps, and each feature map is gained by convolution operation. Thus

$$C_2(i,j,\tau) = \varphi_{\mathrm{ReLU}}\left(\sum_{k=0}^{31}\sum_{i_0=0}^{2}\sum_{j_0}^{2} R_x(i+i_0, j+j_0, k)\right) k_\tau^{C_2} \cdot (3-i_0, 3-j_0, 31-k). \tag{13}$$

(4) *Pooling Layer* S_2. S_2 contains 64 feature maps, and maximum pooling is conducted for each feature map

$$S_2(i,j,\tau) = \max\{C_2(2i-1, 2j-1, \tau), C_2(2i-1, 2j, \tau), C_2(2i, 2j-1, \tau), C_2(2i, 2j, \tau)\}. \tag{14}$$

(5) *Convolutional Layer* C_3. C_3 contains 128 feature maps, and each feature map is gained by convolution operation

$$C_3(i,j,\tau) = \varphi_{\mathrm{ReLU}}\left(\sum_{k=0}^{127}\sum_{i_0=0}^{2}\sum_{j_0}^{2} R_x(i+i_0, j+j_0, k)\right) k_\tau^{C_3}(3-i_0, 3-j_0, 127-k). \tag{15}$$

(6) *Pooling Layer* S_3. S_3 contains 128 feature maps, and maximum pooling is conducted for each feature map

$$S_3(i,j,\tau) = \max\{C_3(2i-1, 2j-1, \tau), C_3(2i-1, 2j, \tau), C_3(2i, 2j-1, \tau), C_3(2i, 2j, \tau)\}. \tag{16}$$

(7) *Convolutional Layer* C_4. C_4 contains 128 feature maps, and each feature map is gained by convolution operation

$$C_4(i,j,\tau) = \varphi_{\mathrm{ReLU}}\left(\sum_{k=0}^{127}\sum_{i_0=0}^{2}\sum_{j_0}^{2} R_x(i+i_0, j+j_0, k)\right) k_\tau^{C_4}(3-i_0, 3-j_0, 127-k), \tag{17}$$

(8) *Pooling Layer* S_4. S_4 contains 128 feature maps, and each feature map is gained by convolution operation

$$S_4(i,j,\tau) = \max\{C_4(2i-1, 2j-1, \tau), C_4(2i-1, 2j, \tau), C_4(2i, 2j-1, \tau), C_4(2i, 2j, \tau)\}. \tag{18}$$

(9) *Fully Connected Layer* F_1. F_1 is fully connected with S_4, the number of neurons in F_1 is 512

(10) *Fully Connected Layer* F_2. F_2 is fully connected with F_1 with the neuron number 512. F_2 is a probability value because SS is a binary classification issue. This probability value can be expressed as $P_{L_1}(R_x)$. When $P_{L_1}(R_x) > 0.5, R_x \in H_1$; otherwise, $R_x \in H_0$.

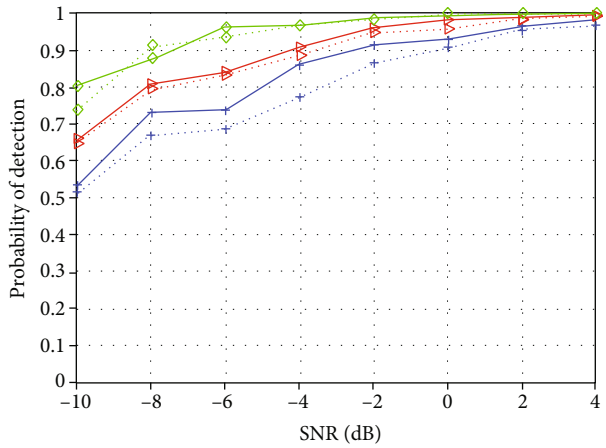

Figure 19: Detection probability with CNN-1.

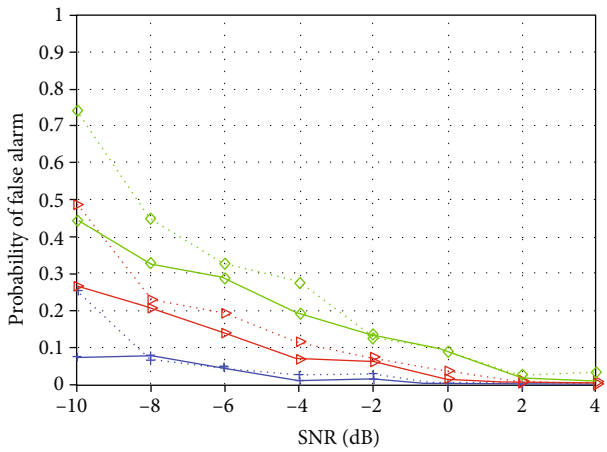

Figure 20: False alarm probability with CNN-1.

Figures 19 and 20, respectively, exhibit the PD and PFA with the proposed CNN-1 scheme under various SNRs. From Figures 19 and 20, the detection probability of the CNN-1 with the proposed data enhancement scheme is higher than that without the data enhancement while the false alarm probability of the CNN-1 with the proposed data enhancement scheme is lower than that without the data enhancement. In addition, when SNR ≥ -8dB, the mean value of PD is more than 0.8 and the mean value of PFA is less than 0.2, which is much better than that of the LeNet and the same as that of the VGG-16.

Figure 21 gives the performance comparisons of various SS schemes including the LeNet based scheme, the AlexNet based scheme, the VGG-16 based scheme, and the proposed CNN-1 based scheme. As it is shown for the black line, it, respectively, denotes -2 dB, 0 dB, 2 dB, and 4 dB from bottom to top, and the other lines are the same as the black one. From Figure 21, the sensing performance of the CNN-1 based SS scheme is highest compared to the other SS schemes under the same SNR, which corresponds with the performance in Figures 13–20.

In Figure 22, the computation time comparison of different CNN networks is discussed, where we use the time it

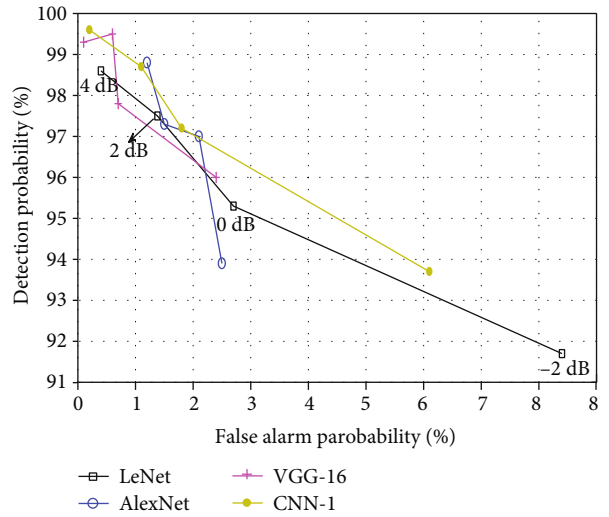

Figure 21: Performance comparisons of various SS schemes.

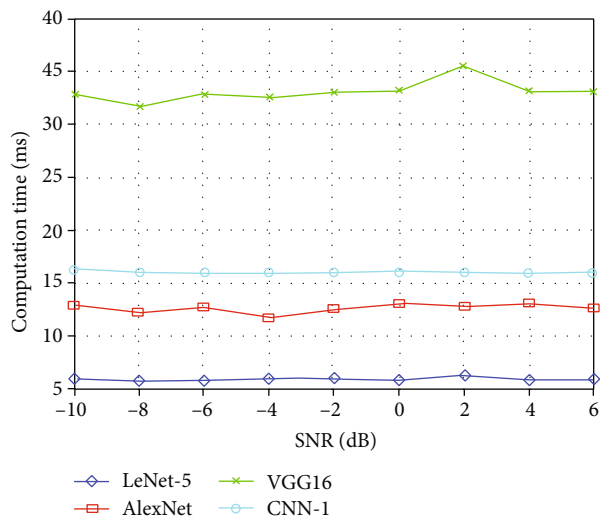

Figure 22: Computation time comparisons of different CNN networks.

takes for a model to process a true color image to represent computational complexity and evaluate the computational complexity of different CNN algorithms. From Figure 22, the computation time of the LeNet, AlexNet, CNN-1, and VGG-16 is about 6 ms, 12.5 ms, 16 ms, and 33 ms, which indicates that the LeNet is the simplest and the VGG-16 is the most complex.

As a summary, the detection performance of CNN-1 is similar to that of VGG-16, but the computation time of CNN-1 is nearly half that of VGG-16. This indicates that CNN-1 is a better convolution neural network model for SS.

As a supplement, performance comparisons are made between the proposed CNN-1 and the schemes in [16, 17]. As it is shown for the black line, it, respectively, denotes -2 dB, 0 dB, 2 dB, and 4 dB from bottom to top, and the other lines are the same as the black one. From Figure 23, the sensing performance of the proposed CNN-1 scheme is slightly higher than that of the scheme in [16]. Meanwhile, the

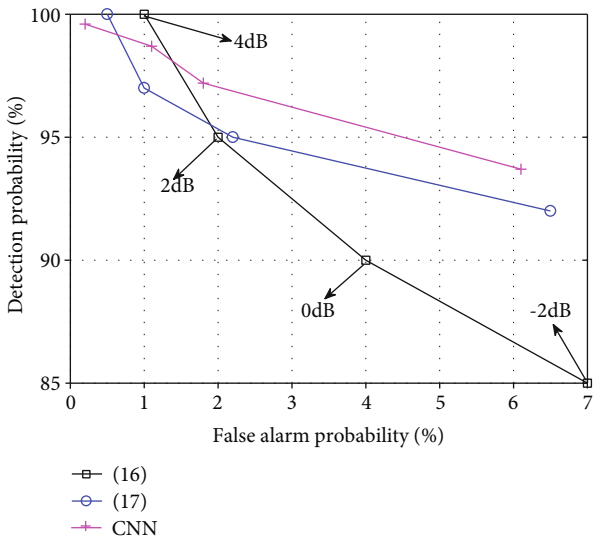

Figure 23: Performance comparisons between CNN-1 with the schemes in [16, 17].

sensing performance of the proposed CNN-1 scheme is significantly higher than that of the scheme in [17]. This states that the proposed CNN-1 scheme with data enhancement is more suitable for the detection of spectrum state.

5. Conclusions

In this paper, the deep learning based spectrum sensing is discussed for sustainable cities and society, where the LeNet, AlexNet, VGG-16, and the proposed CNN-1 network are considered. First, the two-dimensional dataset of the received signal is established and expanded by the proposed DCGAN scheme. Then, the four networks are trained on the expanded dataset. Finally, the test is made under the various SNR conditions. The experiment results show that the sensing performance is greatly improved by the proposed data enhancement scheme and the novel CNN network.

Data Availability

The data supporting the results of my study can be found at https://figshare.com/articles/dataset/__part1_rar/14245763.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Wu, B. Rong, K. Salehian, and G. Gagnon, "Cloud transmission: a new spectrum-reuse friendly digital terrestrial broadcasting transmission system," *IEEE Transactions on Broadcasting*, vol. 58, no. 3, pp. 329–337, 2012.

[2] N. Chen, B. Rong, X. Zhang, and M. Kadoch, "Scalable and flexible massive MIMO precoding for 5G H-CRAN," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 46–52, 2017.

[3] Y. Cui, F. Liu, X. Jing, and J. Mu, "Integrating sensing and communications for ubiquitous IoT: applications, trends and challenges," 2021, http://arxiv.org/abs/2104.11457.

[4] A. Bishnu and V. Bhatia, "LogDet covariance based spectrum sensing under colored noise," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6716–6720, 2018.

[5] K. Zheng, X.-Y. Liu, X. Liu, and Y. Zhu, "Hybrid overlay-underlay cognitive radio networks with energy harvesting," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4669–4682, 2019.

[6] J. I. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[7] K. Yadav, S. Dhar Roy, and S. Kundu, "Throughput of cognitive radio networks with improved energy detector under security threats," *International Journal of Communication Systems*, vol. 31, no. 6, article e3512, 2018.

[8] Y. Lu, "A sensing contribution-based two-layer game for channel selection and spectrum access in cognitive radio ad-hoc networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3631–3640, 2018.

[9] E. Ollila and E. Raninen, "Optimal shrinkage covariance matrix estimation under random sampling from elliptical distributions," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2707–2719, 2019.

[10] X. Hao, G. Zhang, and S. Ma, "Deep learning," *International Journal of Semantic Computing*, vol. 10, no. 3, pp. 417–439, 2016.

[11] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning based semantic communications: an initial investigation," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan, December 2020.

[12] S. Zheng, S. Chen, and X. Yang, "DeepReceiver: a deep learning-based intelligent receiver for wireless communications in the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 5–20, 2021.

[13] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: a deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2020.

[14] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Non-cooperative OFDM spectrum sensing using deep learning," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, Big Island, HI, USA, February 2020.

[15] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1727–1730, 2019.

[16] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Communications*, vol. 17, no. 2, pp. 138–148, 2020.

[17] J. Xie, J. Fang, C. Liu, and X. Li, "Deep learning-based spectrum sensing in cognitive radio: a CNN-LSTM approach," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2196–2200, 2020.

[18] S. Lin, L. Cai, X. Lin, and R. Ji, "Masked face detection via a modified LeNet," *Neurocomputing*, vol. 218, pp. 197–202, 2016.

[19] M. Alencastre-Miranda, R. R. Johnson, and H. I. Krebs, "Convolutional neural networks and transfer learning for quality

inspection of different sugarcane varieties," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 787–794, 2021.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2014.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Computer Science*, 2015.

[22] J. Zhang, L. Chen, L. Zhuo, X. Liang, and J. Li, "An efficient hyperspectral image retrieval method: deep spectral-spatial feature extraction with DCGAN and dimensionality reduction using t-SNE-based NM hashing," *Remote Sensing*, vol. 10, no. 2, p. 271, 2018.

[23] Y. Liu, J. Yi, X. Wan, X. Zhang, and H. Ke, "Time varying clutter suppression in CP OFDM based passive radar for slowly moving targets detection," *IEEE Sensors Journal*, vol. 10, no. 2, p. 271, 2020.

[24] M. Ranjbar, N. H. Tran, M. Vu, T. V. Nguyen, and M. Cenk Gursoy, "Capacity region and capacity-achieving signaling schemes for 1-bit ADC multiple access channels in Rayleigh fading," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 6162–6178, 2020.

[25] R. Umar, A. U. H. Sheikh, and M. Deriche, "Unveiling the hidden assumptions of energy detector based spectrum sensing for cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 713–728, 2014.

[26] J. Mu, X. Jing, H. Huang, and N. Gao, "Subspace based method for spectrum sensing with multiple users over fading channel," *IEEE Communications Letters*, vol. 22, no. 4, pp. 848–851, 2018.