Cooperative Multi-Node Jamming Recognition Method Based on Deep Residual Network

Junren Shen 1,2, Yusheng Li 1,*, Yonggang Zhu 1 and Liujin Wan 1,*

1 63rd Research Institute, National University of Defense Technology, Nanjing 210007, China
2 School of Electronic Science, National University of Defense Technology, Changsha 410000, China
* Correspondence: albert_atr@126.com (L.W.); lys63s@163.com (Y.L.)

Abstract: Anti-jamming is the core issue of wireless communication viability in complex electromagnetic environments, where jamming recognition is the precondition and foundation of cognitive anti-jamming. In the current jamming recognition methods, the existing convolutional networks are limited by the small number of layers and the extracted feature information. Simultaneously, simple stacking of layers will lead to the disappearance of gradients and the decrease in correct recognition rate. Meanwhile, most of the jamming recognition methods use single-node methods, which are easily affected by the channel and have a low recognition rate under the low jamming-to-signal ratio (JSR). To solve these problems, a multi-node cooperative jamming recognition method based on deep residual networks was proposed in this paper, and two data fusion algorithms based on hard fusion and soft fusion for jamming recognition were designed. Simulation results show that the use of deep residual networks to replace the original shallow CNN network structure can gain a 6–14% improvement in the correct recognition rate of jamming signals, and the hard and soft fusion-based methods can significantly improve the correct jamming recognition rate by about 3–7% and 5–12%, respectively, under low JSR conditions compared with the existing single-node method.

Keywords: jamming recognition; multi-node cooperation; residual network; data fusion

1. Introduction

Due to its openness and broadcast nature, the wireless communication signal transmitted by the wireless channel is susceptible to attack by malicious enemy jamming, which is a major factor threatening the viability of communication [1]. In order to solve this problem, various anti-jamming techniques have been proposed, where jamming recognition is a key part of cognitive anti-jamming techniques and also the basis and prerequisite for the implementation [2–5]. Jamming recognition methods can be divided into two categories based on the presence or absence of feature extraction. The features-based category is an evaluation by the performance parameters that are described to be the calculation of C (carrier factor coefficient), FSE (average frequency domain flatness coefficient) and other feature parameters of jamming [5], extraction of a signal’s time–frequency diagrams [6] and decision tree classification algorithms [7]—all such methods rely on setting up the threshold values that are greatly influenced by the channel environment and prone to lose useful information. Compared to threshold-based features analysis, the deep learning categories [8–10] can directly use the original information for deep feature mapping, which does not require pre-extraction of features and, thus, can obtain better recognition performance. The Convolutional Neural Network (CNN) has a multilayer network structure which is suitable for solving image feature extraction problems and has advantages in extracting features related to special (two-dimensional) features [11], and so CNNs are described as bearing the potential for jamming and modulation recognition. Consequently, researchers have introduced CNNs into these fields where the CNN also obtains leading performance. Moreover, one of the active research areas on jamming is related to jamming
of the classical and quantum radar [12]. Researchers used ResNet-AE [13] or Quantum-secured imaging [14] to recognize the jamming of radar. Additionally, the multi-node method was used to provide decisions for the anti-jamming of radar [15,16]. In the next step of our research, we can apply the multi-node method to the jamming recognition of classical and quantum radar to improve the overall performance compared with the single-node method.

For the concentration and unsatisfactory performance of single-node jamming recognition among most of the current research, authors used a multi-node method based on convolutional neural networks for jamming recognition in a previous work [17] to solve the problem of the single-node method being susceptible to channel influence, which effectively improved the correct recognition rate of jamming at a lower JSR. To the authors’ knowledge, this is the first paper to apply a multi-node approach to the field of jamming recognition. However, existing CNN jamming recognition methods still suffer from the problems of a small number of network layers and limited feature information extraction, which will lead to the gradient disappearance of CNNs and a decrease in the correct recognition rate of jamming as the number of network training layers increases.

The shortcut connection [18] was applied early to Hopfield networks by adding a cross-layer loop between two neurons of the neural network, enabling the gradient to skip the current neuron and pass directly to the next neuron, which is conducive to gradient transfer during neural network training and improves the efficiency of information propagation. Earlier multilayer perceptrons also added shortcut connections between network layers to address gradient disappearance, thus enabling the stacking of more network layers [19]. Deep network structure helps to improve the correct rate of signal classification and recognition. What is more, the introduction of residual connections [20] into the jamming recognition problem helps the gradient transfer to deeper networks, overcoming problems such as limited extraction of features by jamming recognition methods in existing shallow convolutional networks and gradient disappearance due to deeper layers.

The simple network structure of the shallow CNN jamming recognition method leads to limited extraction of feature information of the received signal and decrease in the correct recognition rate due to stacking layers directly. In order to solve this problem, this paper adds residual connections to CNN and proposes a multi-node cooperative jamming recognition method based on deep residual networks with reference to the cooperative spectrum sensing scheme [21]. This method uses multiple nodes to jointly perform jamming signal perception and deep residual networks for jamming signal recognition, while two schemes based on hard fusion as well as soft fusion are proposed when the central node performs data fusion.

2. System Model

Assume that the multi-node cooperative jamming recognition network consists of one central node and $M$ cooperative cognitive nodes [22,23], as shown in Figure 1. After receiving the signal from $M$ cognitive nodes, the central node will train a network for recognition. The signal $y_i(n)$ received by the $i$-th cognitive node can be expressed as:

$$y_i(n) = S_i(n) + J_i(n) + v_i(n) \quad n = 1, 2, \cdots N$$

Here $S_i(n)$ is the communication signal, $J_i(n)$ is the jamming signal, $v_i(n)$ is the noise signal, and this noise signal obeys a complex Gaussian distribution with mean 0 and variance $\sigma^2$, and $N$ is the number of signal sampling points.

The vector of sample points received by the $M$ cognitive nodes at moment $n$ can be expressed as: $y(n) = [y_1(n), y_2(n), \ldots, y_M(n)]^T$ [24,25]. The received information is aggregated to the central node and can be written as $y = [y_1, y_2, \ldots, y_N]$, which is composed of the
information of $M$ cognitive nodes in the cognitive time period. It can also be presented as the received signal matrix, i.e.,

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}^T = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(N) \\ y_2(1) & y_2(2) & \cdots & y_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ y_M(1) & y_M(2) & \cdots & y_M(N) \end{bmatrix}_{M \times N}$$  (2)

Figure 1. Framework of multi-node cooperative jamming recognition.

The communication signal in this paper is a QPSK signal, and the channel is an additive Gaussian white noise channel. After receiving the information from each cooperative cognitive node, the central node aggregates the information to the input of neural network. The input layer performs the FFT transform of the received signal matrix, i.e., $Y = \text{FFT}[y]$, then the real part as well as the imaginary part of the FFT transform of the received signal matrix can be expressed as:

$$Y_I = \begin{bmatrix} I_1(1) & I_1(2) & \cdots & I_1(N) \\ I_2(1) & I_2(2) & \cdots & I_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ I_M(1) & I_M(2) & \cdots & I_M(N) \end{bmatrix}_{M \times N}$$  (3)

$$Y_Q = \begin{bmatrix} Q_1(1) & Q_1(2) & \cdots & Q_1(N) \\ Q_2(1) & Q_2(2) & \cdots & Q_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ Q_M(1) & Q_M(2) & \cdots & Q_M(N) \end{bmatrix}_{M \times N}$$  (4)

$Y_I$ and $Y_Q$ are used as input of the neural network model in the form of pages, that is, the input to the neural network is a two-page matrix. Compared to inputting the I-way and Q-way information into the neural network training by rows, the input by pages can make better use of the interrelationship between the I-way and Q-way during network training and obtain more information, which can improve the performance of the final trained network [26].

Finally, the trained network is used to classify the six malicious jamming signals. In this paper, the six common malicious jamming styles [27] to be classified are single-tone jamming, multi-tone jamming, narrow-band jamming, broad-band jamming, comb jamming and swept jamming. The amplitude spectra of the six jamming signals at a Jammer-to-Noise Ratio (JNR) of 10 dB are given in Figure 2.
3. Cooperative Multi-Node Jamming Recognition Method Based on Deep Residual Network

The existing CNN jamming recognition methods generally use the LeNet-5 network structure [6]. Given that the more layers of the CNN, the richer the features that can be learned, as the common CNN models are relatively simple, directly stacking the number of network layers will lead to the disappearance of gradients, resulting in a decrease in the correct recognition rate of jamming signal. Hence, this paper adds residual connections to existing CNN networks and proposes a multi-node cooperative jamming recognition method based on deep residual neural networks with referring to a multi-node cooperative spectrum sensing scheme. The process of the method is shown in Figure 3. The jamming signals are perceived and transmitted back to the central node by multiple cooperative perception nodes for data fusion and training of the neural network. The central node then receives the current jamming signal information and uses the trained neural network for jamming recognition.

![Figure 3. Multi-node cooperative jamming recognition process.](image)

3.1. Residual Connections

Gradient disappearance occurs from time to time during the training of deeper CNNs while residual learning can solve the gradient vanishing problem, to some extent. Suppose that, in a deep network, we expect a non-linear unit (which can be one or more convolutional...
that, in a deep network, we expect a non-linear unit (which can be one or more convolutional layers) to approximate an objective function $h(x)$. At this point, the objective function is split into two parts: the identity function $x$ and the residue function $h(x) - x$:

$$h(x) = x + (h(x) - x) \quad (5)$$

According to the general approximation theorem, a nonlinear unit constituted by a neural network has sufficient capacity to approximate either the original objective function or the residual function, but, in practice, the latter is easier to learn [20]. Thus, the original optimisation problem can be transformed into a new case: let the non-linear unit $f(x; \theta)$ approximate the residual function $h(x) - x$ and use $f(x; \theta) + x$ to approximate the objective function $h(x)$.

An example of a typical residual module is given in Figure 4. The residual module consists of multiple cascaded convolutional layers and a straight-connected edge across the layers, followed by an activation function to obtain the output. The convolutional layers within the shortcut connection are omitted. A residual network is a deep network consisting of several residual modules connected in series, which is similar to a highway network. To facilitate analysis without loss of generality, a simplified structure of a residual network with five residual modules is given in Figure 5.

![Figure 4. Example of a residual module.](image_url)

![Figure 5. Simple residual network structure diagram with six residual modules.](image_url)

The residual network learns features from shallow $l$ to deep $L$ via shortcut connections, which can be expressed as:

$$x_L = x_l + \sum_{i=l}^{L} F(x_i; \{W_i\}) \quad (6)$$
Using the chain rule of derivation, the gradient in the backpropagation process can be expressed as:

\[
\frac{\partial \text{loss}}{\partial x_l} = \frac{\partial \text{loss}}{\partial x_L} \frac{\partial \text{loss}}{\partial x_I} = \frac{\partial \text{loss}}{\partial x_L} \left( 1 + \frac{\partial}{\partial x_L} \sum_{i=1}^{L-1} F(x_i, \{W_i\}) \right)
\]

(7)

where \(\frac{\partial \text{loss}}{\partial x_l}\) denotes the gradient of the loss function reaching the \(x_l\) layer, and the “1” in Equation (7) indicates that the shortcut connection can pass the gradient without loss, while the other residual gradient cannot be propagated directly and needs to pass through the \(l\) layer to the \((L - 1)\) layer with the convolutional layer with the right value \(W_l\). Even if the residual gradient is close to 0, the presence of an “1” does not make the gradient of the residual network disappear. This is the reason why adding residual connections to a deep network allows the gradient to be passed deeper and allows the network layers to be trained even if they are very deep.

In the forward propagation process of neural network training, there is an error between the output of the residual network and the actual output, and the loss function can be expressed as:

\[
\text{Loss} = \frac{1}{2} \sum_{i=1}^{c} \left| s^{(n)}_{\text{train}}(i) - \hat{s}^{(n)}_{\text{train}}(i) \right|^2
\]

(8)

where \(c\) denotes the number of jamming classifications, \(s^{(n)}_{\text{train}}\) denotes the true jamming classification label, and \(\hat{s}^{(n)}_{\text{train}}\) denotes the predicted jamming classification label.

In the backpropagation process, the gradient of the last layer \(k\) can be expressed as:

\[
\frac{\partial \text{loss}}{\partial x_k} = \frac{1}{2} \sum_{i=1}^{c} \left| s^{(n)}_{\text{train}}(i) - \hat{s}^{(n)}_{\text{train}}(i) \right|^2 \frac{\partial \hat{s}^{(n)}_{\text{train}}(i)}{\partial x_k} = e_j \frac{\partial f(x_k)}{\partial x_k}
\]

(9)

Bringing \(l = 0, 3, 6, \ldots, k - 3, k\) into Equation (7) in turn and associating it with Equation (9), we can deduce that the gradient of \(x_0\) is:

\[
\frac{\partial \text{loss}}{\partial x_0} = e_j \frac{\partial f(x_k)}{\partial x_k} \cdot \left( 1 + \frac{\partial}{\partial x_3} \sum_{i=0}^{2} F(x_i, \{W_i\}) \right) \cdot \left( 1 + \frac{\partial}{\partial x_6} \sum_{i=3}^{5} F(x_i, \{W_i\}) \right) \cdots \left( 1 + \frac{\partial}{\partial x_{L-3}} \sum_{i=L-3}^{L-1} F(x_i, \{W_i\}) \right)
\]

(10)

In addition, we can use a similar approach to derive the gradient of \(x_0\) in a CNN without residual connections as:

\[
\frac{\partial \text{loss}}{\partial x_0} = e_j \frac{\partial f(x_k)}{\partial x_k} \cdot \frac{\partial x_k}{\partial x_3} \cdot \frac{\partial x_k}{\partial x_6} \cdots \frac{\partial x_k}{\partial x_{L-3}}
\]

(11)

Comparing Equations (10) and (11), it can be concluded that as the number of layers of the network increases, the gradient values of each convolutional layer are cumulatively multiplied in the CNN without residual connections, causing the gradient reaching \(x_0\) to become smaller and smaller until it disappears. In contrast, the presence of the coefficient “1” in the network with residual connections does not lead to the disappearance of the gradient even if the value obtained by cumulative multiplication of the weights \(W\) of each convolutional layer is close to 0.

3.2. Residual Network Structure

The residual network designed in this paper is composed of an input layer, a convolutional layer (Conv), a batch normalization (BN) layer, a ReLU layer, a fully connected layer(fc), a classification layer and a residual connection. Figure 6 shows a schematic diagram of the structure of the two residual modules, where the residual module changes
the size of the convolutional layer by using a $1 \times 1$ convolutional layer as well as the BN when the size of the convolutional layer changes. In the two residual modules, “conv” denotes the convolutional layer, “$K$” denotes the number of convolutional kernels, “$S$” denotes equal-width convolution and “/2” denotes downsampling by factor 2. The residual module (a) indicates that it has been downsampled, while the residual module (b) indicates equal-width convolution.

![Diagram of residual modules](image)

**Figure 6.** Example of two residual modules. The residual module (a) indicates that it has been downsampled, while the residual module (b) indicates equal-width convolution.

The structural parameters of this residual network are given in Table 1, where block1 indicates the use of structure (a) in Figure 6, and block2 indicates the use of structure (b).

**Table 1.** Residual network structure parameters $r$.

<table>
<thead>
<tr>
<th>Index</th>
<th>Layers</th>
<th>Output Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input</td>
<td>$512 \times 1$</td>
</tr>
<tr>
<td>2</td>
<td>$15 \times 1, \text{conv}, 48$</td>
<td>$512 \times 1$</td>
</tr>
<tr>
<td>3</td>
<td>$7 \times 1, \text{conv}, 64$</td>
<td>$512 \times 1$</td>
</tr>
<tr>
<td>4</td>
<td>$3 \times 1, \text{maxpool}, /2$</td>
<td>$256 \times 1$</td>
</tr>
<tr>
<td>5</td>
<td>Residual-block1 (64)</td>
<td>$128 \times 1$</td>
</tr>
<tr>
<td>6</td>
<td>Residual-block2 (64)</td>
<td>$128 \times 1$</td>
</tr>
<tr>
<td>7</td>
<td>Residual-block1 (128)</td>
<td>$64 \times 1$</td>
</tr>
<tr>
<td>8</td>
<td>Residual-block2 (128)</td>
<td>$64 \times 1$</td>
</tr>
<tr>
<td>9</td>
<td>Residual-block2 (128)</td>
<td>$64 \times 1$</td>
</tr>
<tr>
<td>10</td>
<td>$32 \times 1, \text{average pooling layer}$</td>
<td>$1 \times 256$</td>
</tr>
<tr>
<td>11</td>
<td>$\text{fc} \times 32$</td>
<td>$1 \times 32$</td>
</tr>
<tr>
<td>12</td>
<td>Softmax, $\text{fc} \times 6$</td>
<td>$1 \times 6$</td>
</tr>
</tbody>
</table>

Some of the parameters for the training of the neural network in this paper are given in Table 2.

**Table 2.** Training parameters of CNNs.

<table>
<thead>
<tr>
<th>Training Parameters</th>
<th>Numerical Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Number of iterative rounds</td>
<td>6</td>
</tr>
<tr>
<td>Small batch size</td>
<td>128</td>
</tr>
<tr>
<td>Parameter optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning rate drop period</td>
<td>9</td>
</tr>
<tr>
<td>Convolution kernel size</td>
<td>$1 \times 1, 3 \times 1, 7 \times 1, 15 \times 1$</td>
</tr>
<tr>
<td>Desampling step size</td>
<td>2</td>
</tr>
</tbody>
</table>

3.3. Multi-Node Cooperative Jamming Recognition Method

In single node jamming recognition methods, the recognition results tend to deteriorate due to channel effects. Cooperative spectrum sensing uses multiple cooperative sensing nodes to overcome this problem. Referring to this scheme and its data fusion...
scheme [28–31], two multi-node jamming recognition schemes are proposed in this paper, using two different data fusion algorithms for jamming recognition, namely, the hard and soft fusion-based cooperative jamming recognition method.

In the hard-fusion based multi-node cooperative jamming recognition method, the recognition network consists of one central node and M cooperative cognitive nodes, where the algorithm can be divided into three phases. In the training phase, each of the M cognitive nodes perceives the jamming signal, and the central node uses this perception information to train a neural network and distribute the parameters of this neural network to the individual cognitive nodes. In the test phase, the M cognitive nodes use the received neural network to recognize the current jamming signal, generate the recognition result \( H_i \) and transmit the recognition result back to the central node. In the data fusion phase, the global jamming recognition result \( H_w \) is given by the central node based on the majority judgment criterion. The global recognition result \( H_w \) can be expressed as:

\[
H_w = \Psi(H_1, H_2 \cdots H_M)
\]

(12)

where the function \( \Psi \) indicates returning the first value with the most frequent occurrences. The hard-fusion based multi-node jamming recognition method is shown in Table 3.

<table>
<thead>
<tr>
<th>Table 3. Multi-node cooperative jamming recognition method based on hard decision.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> sensory information from M co-cognitive nodes</td>
</tr>
<tr>
<td><strong>Output:</strong> global jamming signal classification judgement</td>
</tr>
<tr>
<td><strong>Step 1:</strong> (Training phase) The central node trains a neural network “Trainednet” using the perceptual information of the M co-cognitive nodes and distributes the network parameters to each co-cognitive node.</td>
</tr>
<tr>
<td><strong>Step 2:</strong> (Testing phase) The M co-cognitive nodes use the trained neural network to perform classification judgments, independently obtain recognition results ( H_i ) and send the recognition results back to the central node.</td>
</tr>
<tr>
<td><strong>Step 3:</strong> (Data fusion phase) The central node performs data fusion based on the sensory information of each cooperative cognitive node and derives the global recognition result ( H_w ) based on the majority judgment criterion, which is used as the final recognition result of the multi-node cooperative jamming recognition network.</td>
</tr>
</tbody>
</table>

In the soft-fusion based multi-node cooperative jamming recognition method, the structure of the recognition network is the same as the hard one, and the algorithm is also divided into three phases. The difference lies in the test phase where each cooperative cognitive node makes an independent judgement based on the neural network and returns a recognition vector \( V_i \), whose length is equal to the amount of jamming to be recognized, which can be expressed as:

\[
V_i = [V_i(1) V_i(2) V_i(3) V_i(4) V_i(5) V_i(6)], 1 \leq i \leq M
\]

(13)

where \( V_i(j) \) denotes the probability that the co-cognitive node considers the current jamming signal to be the \( j \)th type of jamming, and each component satisfies the following relation:

\[
\sum_{j=1}^{6} V_i(j) = 1, 1 \leq i \leq M
\]

(14)

In the data fusion phase, all vectors are added to produce a global judgment vector \( V_w \) after the central node receives the information from each cooperative cognitive node, and the jamming with the highest probability in the vector \( V_w \) is used as the global recognition result \( H_w \), i.e.,

\[
H_w = \text{MAX}(V_w)
\]

(15)

The soft-fusion based multi-node jamming recognition method is shown in Table 4:
**Table 4.** Multi-node cooperative jamming recognition method based on soft decision.

<table>
<thead>
<tr>
<th>Input: sensory information from M co-cognitive nodes</th>
<th>Output: global jamming signal classification judgement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: (Training phase) The central node trains a neural network “Trainednet” using the sensory information of the M co-cognitive nodes and distributes the network parameters to each co-cognitive node.</td>
<td></td>
</tr>
<tr>
<td>Step 2: (Testing phase) The M co-cognitive nodes use the trained neural network to perform classification judgments, obtain recognition vectors ( V_i ) independently and transmit the recognition results back to the central node.</td>
<td></td>
</tr>
<tr>
<td>Step 3: (Data fusion phase) The central node performs data fusion based on the sensory information of each cooperative cognitive node, all vectors are added to obtain the global judgment vector ( V_w ), and the jamming with the highest probability is used as the global recognition result ( H_w ).</td>
<td></td>
</tr>
</tbody>
</table>

**4. Simulation Result and Analysis**

**4.1. Parameter Settings**

Matlab simulations were used to generate the corresponding signals for QPSK modulated communication signals and six common jamming signals as the training sets and test sets for the neural network. In the simulation, the sampling rate \( f_s \) is 10 MHz, the number of sampling points is 512, the QPSK signal carrier frequency \( f_c \) is 2.5 MHz, the channel model is additive Gaussian white noise, the SNR is \(-5\) dB, and the frequency bands of the jamming signal are all located near the QPSK signal. The JSR of the training set ranges from \(-16\) dB to 10 dB with an interval of 2 dB, and the number of samples for each type of jamming signal under each JSR is 100. The JSR of test set ranges from \(-16\) dB to 10 dB with an interval of 2 dB, and the number of samples for each type of jamming signal under each JSR is 500. The parameters of the jamming signal are shown in Table 5.

**Table 5.** Simulation parameters of jamming signal.

<table>
<thead>
<tr>
<th>Jamming Pattern</th>
<th>Jamming Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-tone Jamming</td>
<td>The frequency point is randomly located near the carrier frequency ( f_c ).</td>
</tr>
<tr>
<td>Multi-tone Jamming</td>
<td>The frequency point is randomly located near the carrier frequency ( f_c ). The number of tones is random.</td>
</tr>
<tr>
<td>Narrow-band Jamming</td>
<td>The central frequency is located at ( f_c ) and the band width is randomly generated.</td>
</tr>
<tr>
<td>Broad-band Jamming</td>
<td>The central frequency is located at ( f_c ) and the band width is randomly generated.</td>
</tr>
<tr>
<td>Comb Jamming</td>
<td>Initial frequency is random around ( f_c ) and the number and intervals of frequency points are random.</td>
</tr>
<tr>
<td>Sweep Jamming</td>
<td>Initial frequency is random around ( f_c ).</td>
</tr>
</tbody>
</table>

**4.2. Performance Analysis**

In this paper, four sets of simulation experiments are designed to verify the performance of the algorithm proposed in this paper. In the experiments, the average correct recognition rate of six types of jamming signals is used as the main performance index.

Experiment 1: Advantages of residual networks with different layers over shallow CNNs. Figure 7 provides a comparison of the performance of the residual network with a different number of layers and the CNN with the single node recognition method (the number of co-cognitive nodes is 6). The figure shows that all residual networks with a different number of layers outperform CNNs. Additionally, the average correct recognition rate of residual networks reaches its highest when the number of network layers is 18. The correct recognition rate of the network first increases with the number of layers and then starts to decrease after reaching about 20 layers. This is because, when the number of network layers is small, increasing the number of layers helps to extract more features, resulting in an increase in the correct recognition rate. However, when the number of layers is deeper, although increasing the number of layers can enable the network to extract more features, it also leads to an increase in the training parameters and training time, and the complexity of the recognition model will also increase; too deep a network leads to
overfitting during training, resulting in a decrease in the correct recognition rate. In this paper, the correct recognition rate peaked at 18 layers, so the residual network with an 18-layer network structure was chosen as the jamming recognition model.

![Performance comparison of ResNet and CNN under single-node method.](image1)

**Figure 7.** Performance comparison of ResNet under different levels.

To further compare the performance differences between the two networks, Table 6 shows the average correct recognition rate of the 2 networks when using the single node solution when the number of layers is 18 for both networks. It can be seen in Figure 8 that the residual network effectively improves the correct recognition rate, and the improvement is even more pronounced at lower JSR, with the average correct recognition rate improving by approximately 6–12%. This is due to the fact that the additional residual connections allow for more efficient training and gradient transfer, resulting in better performance of the trained network.

<table>
<thead>
<tr>
<th>Net</th>
<th>Average Correct Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet</td>
<td>88.08%</td>
</tr>
<tr>
<td>CNN (Lenet)</td>
<td>82.41%</td>
</tr>
</tbody>
</table>

**Table 6.** Performance comparison of ResNet and CNN under single-node method.

![Performance comparison of ResNet and CNN under single-node method.](image2)

**Figure 8.** Performance comparison of ResNet and CNN under single-node method.

Experiment 2: Performance comparison of jamming recognition under different number of nodes in the hard-fusion based method.

Figure 9 gives the performance curves of the hard-fusion based methods with different numbers of co-cognitive nodes. It can be seen from the figure that the correct recognition
rates of the hard fusion-based methods are all higher. At JSR = −5 dB, all curves achieve a correct recognition rate of over 90%. Looking at the correct recognition rate curves for different numbers of nodes, it can be seen that the correct recognition rate gradually increases as the number of nodes increases, and that the performance improvement from increasing the number of nodes is more obvious at low JSR. This is because, as the number of nodes increases, the cognitive nodes receive more samples of jamming signals and more information back to the central node, enabling a more comprehensive judgement, which ultimately leads to an increase in network performance. As the JSR increases, the correct recognition rate of the network increases. This is because, as the JSR increases, the energy of the jamming signal also increases, making it easier for the network to extract the jamming signal information and, thus, make a correct recognition.

Experiment 3: Performance comparison of jamming recognition for a different number of nodes in the soft-fusion based method.

The performance curves of the soft-fusion based method with a different number of nodes are given in Figure 10. Observing the performance curves of the soft-fusion based method in Figure 10, it can be seen that similar to the performance curves of the hard fusion method, the correct recognition rate for all curves reaches over 90% at JSR = −5 dB. In addition, as the number of nodes increases, the correct recognition rate based on the soft-fusion based method gradually increases.

Figure 9. Performance comparison of hard-fusion based methods under a different number of nodes.

Figure 10. Performance comparison of soft-fusion based methods under a different number of nodes.
Experiment 4: Performance comparison of the of hard-fusion based and soft-fusion based multi-node cooperative jamming recognition.

Figure 11 gives the performance curves of the residual network as well as the existing CNN for the soft-fusion based method and the hard-fusion based method when the number of co-cognitive nodes is six. Looking at Figure 11, it can be observed that the residual network outperforms the CNN method in both fusion methods, and the soft-fusion based method outperforms the hard-fusion based method overall. This is because the residual connections incorporated in the residual network allow for better gradient and training information transfer in the deeper network, resulting in better performance of the trained network. Compared to the hard-fusion based method, the soft-fusion based method fuses more information and takes more advantage of multi-node jamming perception. However, the soft-fusion based method also results in increased transmission information, placing higher demands on the data processing capabilities of the central node and adding additional system complexity.

![Figure 11. Performance comparison between ResNet and CNN under two fusion methods.](image)

Under the conditions of Experiment 4, in order to see further how well the two networks and data fusion methods correctly identified each type of jamming, confusion matrices were made for the CNN method under the hard-fusion based method (Figure 12) and for the residual network under both fusion methods when the number of co-cognitive nodes was 6 and JSR = −6 dB (Figures 13 and 14). Generally speaking, the overall performance of the confusion matrix is consistent with Figure 11, with the recognition performance of the residual network outperforming the CNN network and the soft-fusion based method outperforming the hard one. Among the misclassified labels, it is mainly the confusion between narrow-band jamming and broad-band jamming. This is due to the fact that the spectra of the two types of jamming are more similar in the low JSR case, which makes it difficult for the neural network to distinguish between the two types of jamming signal characteristics, and, therefore, it is easy to misclassify the two types of jamming signals. Comparing Figure 12 with Figure 13, it can be seen that the residual network significantly reduces the misclassification probability and mitigates the confusion between different jamming, mainly generating incorrect judgements for broad-band jamming, with the rest of the jamming being correctly recognized at close to 100%. Comparing Figure 13 with Figure 14, it can be seen that the soft-fusion based method further improves the network performance, reducing the misclassification probability of broad-band jamming from 68.2% to 56% compared with the hard-fusion based method.
Figure 12. Confusion matrix of the hard-fusion based method (CNN, the number of co-cognitive nodes is 6 under JSR = −6 dB).

Figure 13. Confusion matrix of the hard-fusion based method (ResNet, the number of co-cognitive nodes is 6 under JSR = −6 dB).

Figure 14. Confusion matrix of the soft-fusion based method (ResNet, the number of co-cognitive nodes is 6 under JSR = −6 dB).

5. Conclusions

In this paper, we investigated a cooperative multi-node jamming recognition method based on deep residual networks and proposed two judgement algorithms based on hard and soft fusion. Through the multi-node co-cognitive jamming signal information and
training the neural network, the jamming signal is recognized by both hard and soft fusion methods. Simulation results show that the use of deep residual networks to replace the original shallow CNN network structure can gain a 6–14% improvement in the correct recognition rate of jamming signals, and the hard-fusion based method can significantly improve the correct recognition rate by about 3–7% under low JSR conditions compared with the single-node method. At the same time, the soft-fusion based method further improves the performance by about 2–5% over the hard one, but the system complexity and processing time increase relatively.

**Author Contributions:** Conceptualization, J.S. and Y.L.; methodology, J.S. and Y.L.; software, J.S.; validation, J.S., L.W. and Y.Z.; formal analysis, J.S. and Y.L.; investigation, J.S. and L.W.; resources, Y.L.; data curation, J.S. and L.W.; writing—original draft preparation, J.S.; writing—review and editing, J.S., Y.L., L.W. and Y.Z.; visualization, J.S.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Science Foundation of China (NSFC grants: U19B214, 61901502), the Foundation Strengthening Plan Area Fund (grants: 2019-JCQ-JJ-212, 2019-JCQ-JJ-226) and the Research Project of the National University of Defense Technology (grants: 18-QNCXJ-029).

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

2. Lang, B.; Gong, J. JR-TFViT: A Lightweight Efficient Radar Jamming Recognition Network Based on Global Representation of the Time–Frequency Domain. *Electronics* 2022, 11, 2794. [CrossRef]
16. Zhao, S.; Yi, M.; Liu, Z. Cooperative Anti-Deception Jamming in a Distributed Multiple-Radar System under Registration Errors. *Sensors* 2022, 22, 7216. [CrossRef]


