electronics

MDPI

Article

JR-TFViT: A Lightweight Efficient Radar Jamming Recognition Network Based on Global Representation of the Time–Frequency Domain

Bin Lang and Jian Gong *

Air Defense and Antimissile School, Air Force Engineering University, Xi'an 710051, China
* Correspondence: drgong@aliyun.com; Tel.: +86-151-2982-2811

Abstract: Efficient jamming recognition capability is a prerequisite for radar anti-jamming and can enhance the survivability of radar in electronic warfare. Traditional recognition methods based on manually designed feature parameters have found it difficult to cope with the increasingly complex electromagnetic environment, and research combining deep learning to achieve jamming recognition is gradually increasing. However, existing research on radar jamming recognition based on deep learning can ignore the global representation in the jamming time–frequency domain data, while not paying enough attention to the problem of lightweighting the recognition network itself. Therefore, this paper proposes a lightweight jamming recognition network (JR-TFViT) that can fuse the global representation of jamming time–frequency domain data while combining the advantages of the Vision Transformer and a convolutional neural network (CNN). The global representation and local information in the jamming time–frequency data are fused with the assistance of the multi-head self-attention (MSA) mechanism in the transformer to improve the feature extraction capabilities of the recognition network. The model's parameters are further decreased by modifying the standard convolutional operation mechanism and substituting the convolutional operation needed by the network with Ghost convolution, which has less parameters. The experimental results show that the JR-TFViT requires fewer model parameters while maintaining higher recognition performance than mainstream convolutional neural networks and lightweight CNNs. For 12 types of radar jamming, the JR-TFViT achieves 99.5% recognition accuracy at JNR = −6 dB with only 3.66 M model parameters. In addition, 98.9% recognition accuracy is maintained when the JR-TFViT parameter number is further compressed to 0.67 M.

Keywords: radar jamming recognition; vision transformer; convolutional neural network (CNN); global representation

Citation: 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. https://doi.org/10.3390/electronics11172794

Academic Editor: Massimiliano Pieraccini

Received: 18 July 2022
Accepted: 3 September 2022
Published: 5 September 2022

1. Introduction

Radar plays a crucial role in contemporary warfare as a real-time information acquisition device. To prevent hostile radars from detecting, tracking, or imaging targets, numerous radar jamming techniques have been developed [1,2]. Efficient radar jamming recognition capability not only provides guidance for radar anti-jamming strategy, but also is the premise for radar to survive in the increasingly complex electromagnetic environment. Therefore, there is vital military significance and practical value to research on radar jamming recognition technologies.

Radar jamming recognition is currently the subject of extensive research both domestically and overseas. The traditional jamming recognition method manually designs the features of radar jamming in the time, frequency, and time–frequency domains, and then implements jamming recognition using classification techniques such as threshold matching, support vector machines (SVM) [3], and decision trees [4]. For example, Ref. [5] obtained a radar jamming mode by extracting the time-domain kurtosis ratio, moment kurtosis coefficient, envelope undulation, and spectral similarity coefficient features and

comparing them according to a certain order of threshold values. The effectiveness of traditional jamming recognition methods relies on the researcher's subjective design of jamming signal features, and the deep abstract features of jamming signals are not easily extracted by human design; thus, the traditional jamming recognition methods have found it difficult to cope with increasingly complex jamming patterns and compound jamming scenarios.

In recent years, a number of new radar jamming recognition methods have emerged. Due to deep learning's strong capacity to automatically learn the features of input data, it has been extensively used in the field of digital image and signal recognition [6]. Radar jamming signals can be converted from a 1D time-domain signal to a 2D time–frequency domain with the help of time–frequency transformation to obtain an image of jamming signal time–frequency distribution [7]. Therefore, using jamming time–frequency images as input data [8–10], deep learning-based image recognition methods are being widely migrated to the application of radar jamming recognition to make up for the lack of feature extraction capability in traditional jamming recognition methods. The authors of [8] used the short-time Fourier transform (STFT) to obtain jamming signal time–frequency images, established the time–frequency image training dataset, and designed a simple convolutional neural network to achieve the jamming recognition of nine kinds of jamming under 0–8 dB JNR conditions. Ref. [9] also used STFT to obtain jamming time–frequency images and used two convolutional neural networks, AlexNet [11] and VGG-16 [12], for recognition experiments, with significantly higher recognition accuracy than the traditional model. The research mentioned above demonstrates that it is practical and efficient to implement jamming recognition using neural network models and time–frequency domain data related to jamming signals. However, the recognition networks of [8,9] choose mature models in the field of computer vision and are not designed according to the characteristics of time–frequency images, so they cannot make full use of the information in time–frequency images and require a large number of samples for training. Ref. [10] extracts the real part, imaginary part, mode, and phase of the jamming time–frequency map to construct multiple datasets and uses an integrated CNN with weighted voting and transfer learning to achieve jamming recognition with excellent performance, even under small sample training conditions. However, Ref. [10] does not consider how lightweight the recognition network should be, and its recognition network requires multi-dimensional data to determine the type of jamming, resulting in a complex recognition network structure and a large number of parameters. It is not conducive to the deployment of actual devices.

In addition, there are still some problems with the above deep learning-based radar jamming recognition methods. Firstly, the existing studies above on jamming recognition based on a CNN with jamming time–frequency images as input do not exploit the global representation of time–frequency images. The time–frequency distribution image is distinct from the natural image, which is an image reflecting the change in jamming signal frequency with time and which has significant global context information. Getting the global representation of the jamming time–frequency image can help enhance the feature extraction ability of the recognition network. However, the structural limitations of CNNs prevent them from fully utilizing the time–frequency images' global representation. Convolution is a straightforward and efficient method for obtaining local information from images, but it is difficult to capture global representations [13]. CNNs need to expand the receptive field by continuously stacking convolutional layers and using pooling operations in order to achieve global information extraction. This mechanism leads to a bloated recognition network, with a significant increase in computation and the number of parameters. The Vision Transformer (ViT) [14] is able to establish long-term dependencies in the input data using self-attention mechanisms and has the outstanding ability to capture global representations [15], and ViT is gradually becoming a new approach to replace CNNs. Unlike CNNs, ViT is heavy in terms of weight, and the performance improvement of ViT-based models comes at the cost of increased network parameters and delays [16]. Unfortunately, in contrast to CNNs, the self-attention module in ViT ignores local feature

details [13]. Apparently, if the different characteristics of CNNs and ViT can be combined for global representation and local information extraction, the recognition performance of jamming recognition networks can be better enhanced by fusing global representation and local information of time–frequency images. Therefore, this paper extracts a recognition network that fuses the global representation and local information in the jamming time–frequency domain, extracts the local information of the jamming time–frequency image using a convolutional operation, and captures the global representation using the self-attention mechanism of ViT.

In addition, the jamming recognition task has strict requirements for real-time performance [17], and a jamming recognition network with a large model and many parameters will struggle to deploy applications on devices with limited resources and power. Therefore, in this paper, the local information is obtained first with a lower number of convolutional module parameters by adjusting the operational mechanism of convolution. Secondly, ViT is fused between convolution modules, which can implicitly combine convolutional characteristics in the whole network and deal with global representation at the same time. This ViT application method can model local information and global representation in the input tensor with fewer parameters [16]. Based on the above improvements, this paper proposes a lightweight jamming recognition network with better performance compared to the large number of parameters in CNN networks, but with very a low number of parameters.

1. The research questions addressed in this paper are as follows. To address the problem that the existing CNN-based radar jamming recognition network cannot fully utilize the global representation of jamming signals in the time–frequency domain, the JR-TFViT is proposed that can fuse the global representation of jamming in the time–frequency domain with local information to improve jamming recognition performance.
2. For the lightweight requirement of the jamming recognition network, the traditional convolutional operation mechanism is adjusted and ViT is fused into the convolutional structure between, which significantly reduces the number of parameters in the jamming recognition network.

The rest of the paper is organized as follows. Section 2 describes the construction method of the radar jamming dataset required for the experiments. Section 3 presents the principle and details of the proposed JR-TFViT construction. Section 4 presents the details of the experiments, the experimental results, and the analysis of the results. Section 5 summarizes the work of this paper and discusses the future research outlook.

2. Radar Jamming Signal Data Preparation

2.1. Generating Jamming Signals

The radar transmit signal uses a line frequency modulation signal to generate the jamming signal dataset required for training and testing based on mathematical models of suppression and deception jamming. To verify the effectiveness of the JR-TFViT, 12 typical radar jamming techniques are used to build network training and test datasets. The main simulation parameters of jamming signals are set as shown in Table 1. Among them, there are three kinds of suppressive jamming: aiming jamming (AJ), blocking jamming (BJ), and sweeping jamming (SJ). There are four kinds of deceptive jamming: distance deception jamming (DDJ), velocity deception jamming (VDJ), interrupted sampling repeater jamming (ISRJ), and smeared spectrum jamming (SMSP). There are five types of composite jamming: DDJ + ISRJ, DDJ + SMSP, VDJ + ISRJ, VDJ + SMSP, and ISRJ + SMSP.

As for the training dataset, 50, 75, and 100 samples were generated for each class of jamming to construct a training set with three sample sizes and a validation set with 100 samples per class, for a total of $12 \times (50 + 75 + 100 + 100) = 3900$ sets of samples. Test data are generated at 2 dB intervals between -6 dB and 12 dB JNR, for a total of 10 JNR environments. Each jamming signal generates 100 sets of samples for a total of

$12 \times 10 \times 100 = 12{,}000$ sets of test samples. *Pjam* is the jamming signal power, *Pnoise* is the noise power added to the jamming signal, and JNR is defined as follows:

$$\mathrm{JNR} = 10\log_{10}\frac{P_{jam}}{P_{noise}} \tag{1}$$

Table 1. Main simulation parameter settings for the jamming signals.

Signal Parameter	Value Ranges
JNR	−6:1:12 dB
center frequency	10~15 MHz
pulse repetition period	100 μs
bandwidth	40 MHz
sampling frequency	200 MHz
number of false targets	4

2.2. Time Frequency Transformation

To obtain the time–frequency domain data of the jamming signal, the generated jamming signal is analyzed in time–frequency using the short-time Fourier transform (STFT) to obtain the time–frequency distribution image of the jamming signal, and the image size is scaled to 256×256 pixels in size. The time–frequency image is used as the input for the JR-TFViT, thus extracting the global representation and local information of the jamming signal in the time–frequency domain. For a jamming signal $j(t)$, $g*(t)$ is the conjugate of the window function $g(t)$. The STFT calculation expression is:

$$STFT(t,f) = \int j(\tau)g^*(\tau - t)e^{-j2\pi f\tau}d\tau \tag{2}$$

The time–frequency distribution images of the 12 randomly selected and generated types of jamming signals are shown in Figure 1.

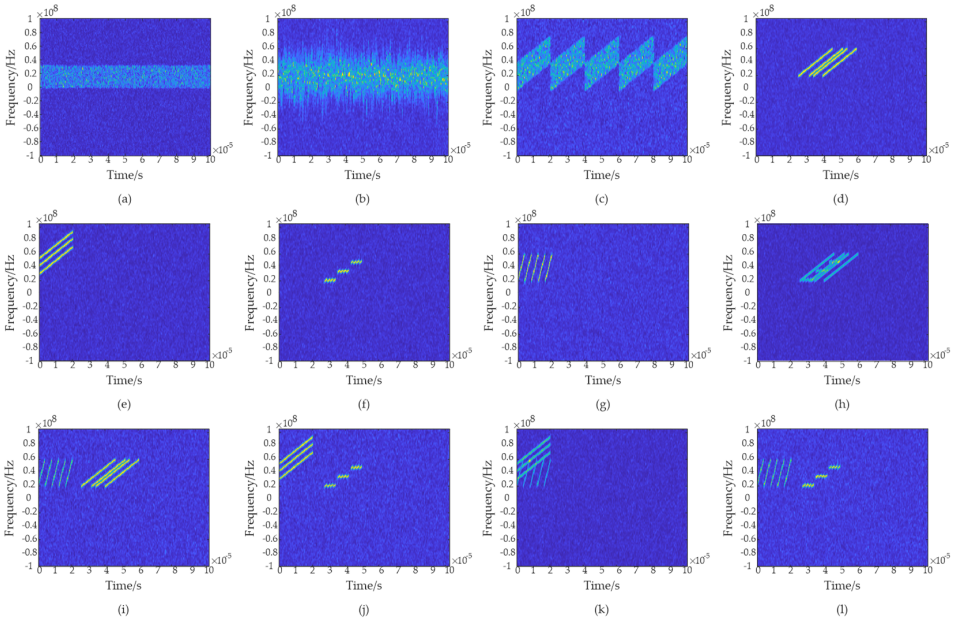

Figure 1. Time−frequency spectrograms of the above 12 jamming signals. (**a**) AJ; (**b**) BJ; (**c**)SJ; (**d**) DDJ; (**e**) VDJ; (**f**) ISRJ; (**g**) SMSP; (**h**) DDJ + ISRJ; (**i**) DDJ + SMSP; (**j**) VDJ + ISRJ; (**k**) VDJ + SMSP; (**l**) ISRJ + SMSP.

3. Methods

In this section, the global representations and local information present in the time–frequency images are first introduced, and the salient ability of the ViT's self-attentive mechanism to capture global representations is then presented. This is followed by a description of the structure and lightweight implementation details of the two basic modules that make up the JR-TFViT. Finally, the complete architecture of the JR-TFViT is described.

3.1. Global Representation and MSA

The spatial frequency model of vision [18] suggests that, in natural images, pictures are divided into a low-frequency part and a high-frequency part [19]. The low and high frequencies are the low- and high-frequency parts corresponding to the picture after Fourier transform, as shown in Figure 2. The low-frequency part corresponds to the global representation of the picture, specifically the part of the grayscale map that changes gently, such as the global shape and structure of a scene or object. The high-frequency part corresponds to the local information of the picture, specifically the part of the grayscale map that changes drastically, such as local edges and textures. Similarly, these global representations and local information are also contained in the jamming time–frequency images, and the features in both the global representations and local information should be used as information for the jamming recognition network. However, existing jamming recognition methods do not make good use of the exploitation of global representations in time–frequency images because the convolutional operation is equivalent to a high-pass filter [20], and the jamming recognition network built with the convolutional operation as the basic module is good at extracting local information while finding it difficult to extract global representations.

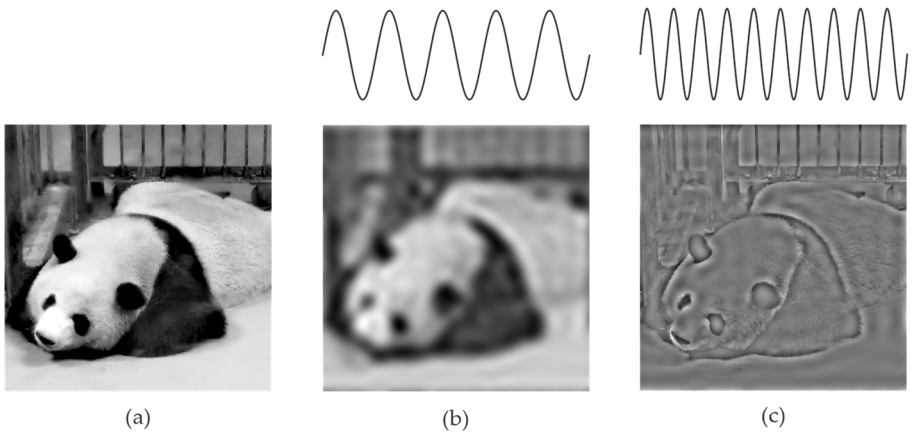

Figure 2. Natural images can be decomposed into a low and a high spatial frequency part. (**a**) Original picture; (**b**) low-frequency part; (**c**) High-frequency part.

Unlike the architecture of CNNs, ViT has the characteristic of a low-pass filter [15] due to the fact that MSA mainly captures low-frequency information and ignores high-frequency components. The basic module that makes up ViT is the transformer encoder, whose structure is shown in Figure 3. The transformer encoder consists of two main modules, the MSA module and the MLP Block [21], each of which is connected to each other using residuals [22]. LayerNorm and Dropout layers help the network to aggregate better and prevent network overfitting, GELU is an activation function, and Linear is a fully connected layer.

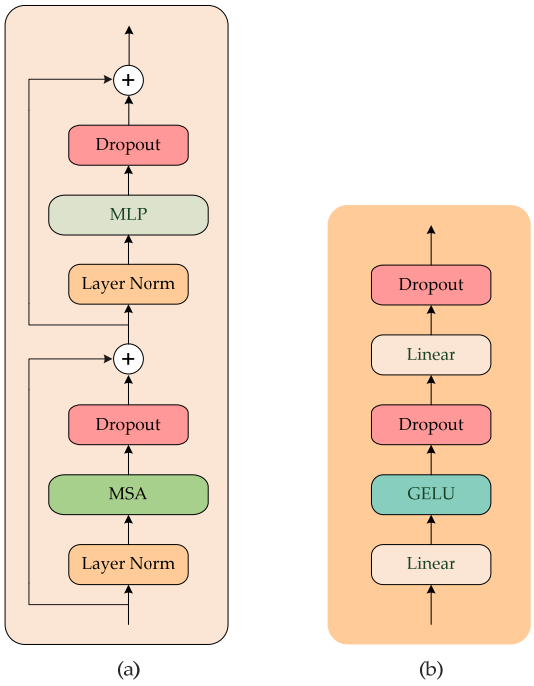

Figure 3. Transformer encoder and MLP Block. (**a**) Transformer encoder; (**b**) MLP Block.

Self-attention can model the long-term dependencies in the input sequence data, and self-attention is calculated as shown in Equations (3) and (4). Q, K, and V represent query, key, and value matrices, respectively, and value represents the extracted information. The query is matched with the key by calculating the correlation between them, and the calculation process is the softmax part in Equation (3) with dk as the length of vector k. The final multiplication with V indicates that greater correlation corresponds to the greater weight of the vector in V.

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

$$Q = \begin{pmatrix} q^1 \\ \dots \\ q^L \end{pmatrix}, K = \begin{pmatrix} k^1 \\ \dots \\ k^L \end{pmatrix}, V = \begin{pmatrix} v^1 \\ \dots \\ v^L \end{pmatrix} \tag{4}$$

For sequence data x of length L, the q, k, and v generation mechanisms are shown in Figure 4. The input is L nodes $x_i, i = 1, 2, \dots, L-1, L$, mapped to a_i by the input embedding operation. W^q, W^k, W^v are trainable parameter matrices, and a_i can obtain the corresponding q_i, k_i, v_i by multiplication with the three transformation matrices W^q, W^k, W^v.

MSA in Figure 3 developed from self-attention by using MSA to be able to combine information learned from different head parts [23]. MSA is calculated as shown in Equations (5) and (6) with W^o, W_i^Q, W_i^K, W_i^V as trainable weight matrices and $j = 1, \dots, h$, h as the number of heads in the MSA.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^o \tag{5}$$

$$\text{head}_j = \text{Attention}\left(QW_j^Q, KW_j^K, VW_j^V\right) \tag{6}$$

Based on the above inspiration of the transformer encoder and MSA in ViT, and with the help of MSA exhibiting a low-pass filter effect [19], the JR-TFViT proposed in this paper introduces a transformer structure to extract the global representation of jamming time–frequency images while retaining the convolutional operation to obtain local infor-

mation of time–frequency images and fuses the global representation of time–frequency images with local information to improve the feature extraction ability and recognition accuracy of jamming recognition networks.

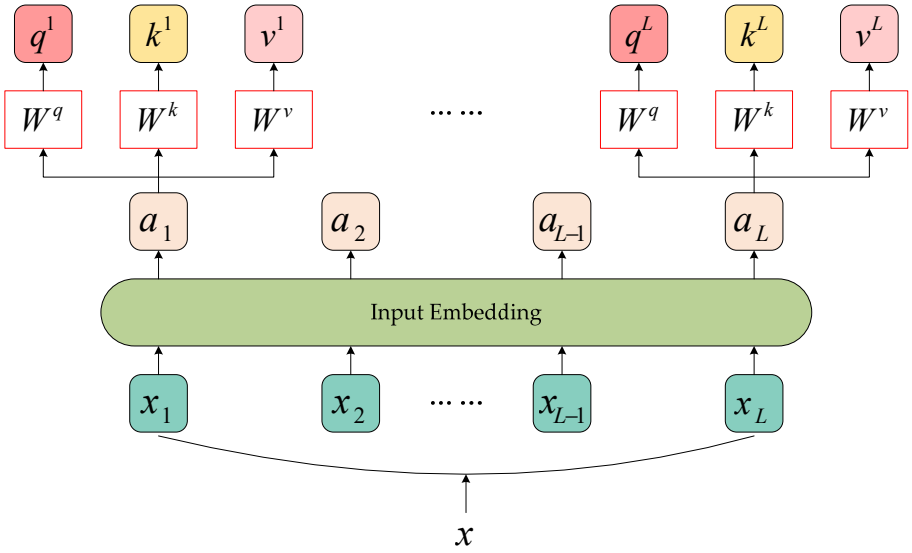

Figure 4. Generation mechanism of q, k, v.

3.2. Lightweight Improvements

3.2.1. Ghost Convolution

Conventional CNNs have a large number of redundant feature maps generated by convolutional operations in the spatial dimension, which are very similar to each other. This rich or even redundant information usually guarantees a comprehensive understanding of the input data [24] and is crucial to the accuracy of the model. However, the process of generating these redundant feature maps consumes a large amount of computational resources and contains a large number of network parameters. The authors of [24] investigated whether it was possible to generate these redundant feature maps in a cost-effective way without removing them directly. Therefore, a new GhostConv module is proposed that uses fewer parameters to generate more features and validates the recognition on the ImageNet ILSVRC2012 classification dataset.

In this paper, the standard convolutional operation is lightly improved based on Ghost convolution, which can significantly reduce the number of network parameters. Suppose a set of input feature map sizes is $C_{in} \times H_{in} \times W_{in}$, with C_{in}, H_{in}, and W_{in} as the number of input feature map channels, height, and width respectively, and the output feature map size is $C_{out} \times H_{out} \times W_{out}$. For a standard convolution with a convolutional kernel size of $K \times K$, the number of parameters for this operation is $(K \times K \times C_{in}) \times C_{out}$. For Ghost convolution, the operation's process is divided into two stages, as shown in Figure 5. Let the size of the convolution kernel remain as $K \times K$; thus, the input feature map is first obtained by a standard convolutional operation for C_{in}/s Channels and the number of parameters of the process is $(K \times K \times C_{in}) \times C_{out}/s$. These feature maps are then linearly transformed to generate redundant feature maps. ϕ refers to the linear transformation operation, and the figure is a depthwise convolution which can be considered as a grouped convolution of $G = C_{in}$. The number of parameters for the process is $K \times K \times C_{out} \times (s-1)/s$. The feature maps of these two parts are combined into the final output feature map, and the number of parameters of the whole process is $(K \times K \times C_{in}) \times C_{out}/s + (K \times K \times C_{in}) \times C_{out}/s$. Ghost convolution is about $1/s$ of the number of standard convolutional parameters, which means that when s is set to 2, the number of Ghost convolutional parameters can be reduced by half compared to the standard convolution.

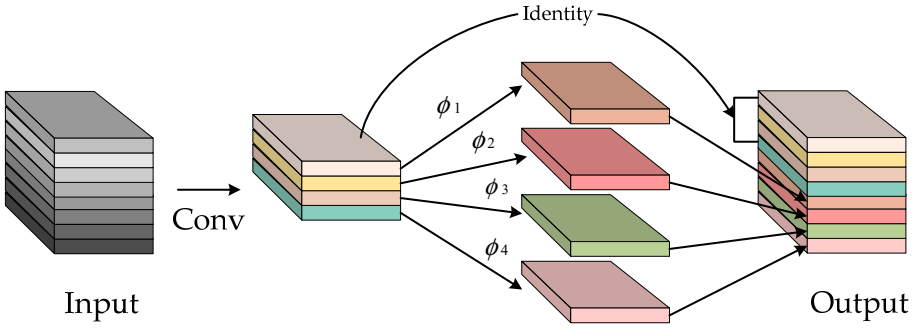

Figure 5. Ghost convolution operation process.

The Ghost Block required for the JR-TFViT is built according to Ghost convolution, and the structure is shown in Figure 6, where '1 × 1','3 × 3' indicates the size of the convolution kernel and 'SiLU' indicates that the activation function used is SiLU. When stride = 1, the input data in the Ghost Block after three levels of GhostConv operations are summed with the input as Ghost Block output according to the inverse residual mechanism [25]. When stride = 2, there is no inverse residual mechanism within the Ghost Block, but a downsampling operation is performed to compress the feature map size to half of the input. With the Ghost Block as the basic module for extracting local information in the JR-TFViT, the number of parameters and the amount of operations are reduced while ensuring the feature extraction capability.

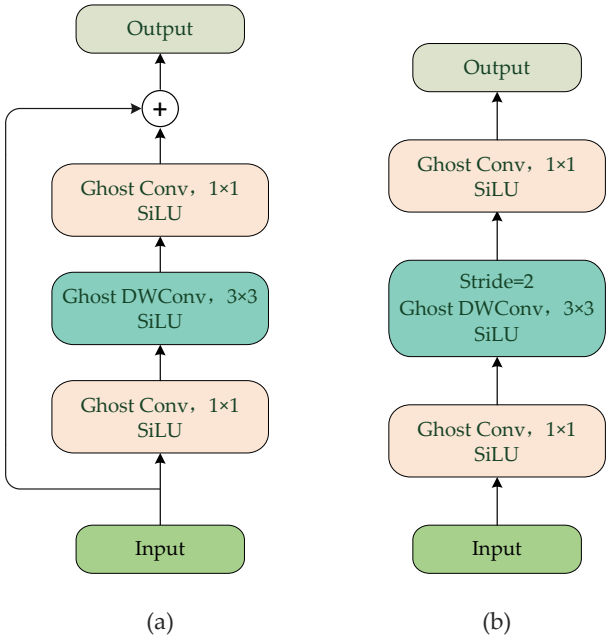

Figure 6. Ghost Block structure. (**a**) Stride = 1 Ghost Block. (**b**) Stride = 2 Ghost Blocks.

3.2.2. Ghost–MobileViT

The standard ViT is shown in Figure 7. For an image with input size $C \times H \times W$, it is first flattened into a set of patches with size $N \times PC$, where N is the number of patches and P is the number of pixels per patch; thus, $P = wh$ and w, h is the width and height of each patch. Each patch is then mapped to a one-dimensional vector by a linear mapping and N tokens meeting the transformer input requirements are obtained, each of length d. Positional encoding operations are then performed to superimpose location information on the tokens. Finally, the L-group stacked transformers are used to learn the inter-patch representations.

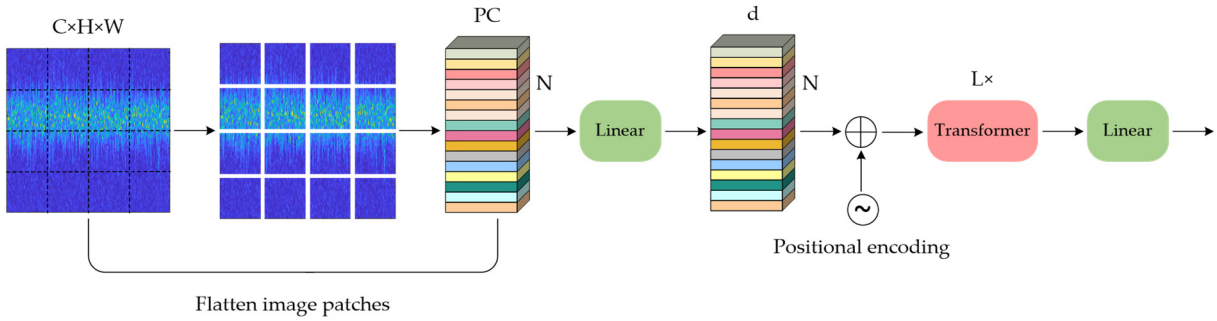

Figure 7. Standard Vision Transformer.

In terms of the number of model parameters, the standard ViT has a much larger number of parameters than the easily optimized and integrated lightweight CNN and requires a large number of data samples for network training. This is because the ViT encodes the global representation of the input data by learning inter-patch information using a transformer; however, the ViT loses the image specific sensing bias inherent to CNNs. As a result, the ViT requires more parameters to learn visual representations, resulting in ViT models that are often deep and wide [16].

In the JR-TFViT, a Ghost–MobileViT is used instead of the standard ViT. A MobileViT, proposed by Apple in 2021, is able to implicitly combine convolutional features in the network to model local information and global representations in the input tensor with fewer parameters. In the original MobileViT Block, both the standard convolution and transformer are used to learn the local and global representations, respectively, and the transformer is responsible for replacing the local processing in the convolution with global processing. In this paper, the MobileViT Block is further lightened and improved by using Ghost convolution to replace the standard convolution in the MobileViT Block, and the Ghost–MobileViT Block is proposed as shown in Figure 8.

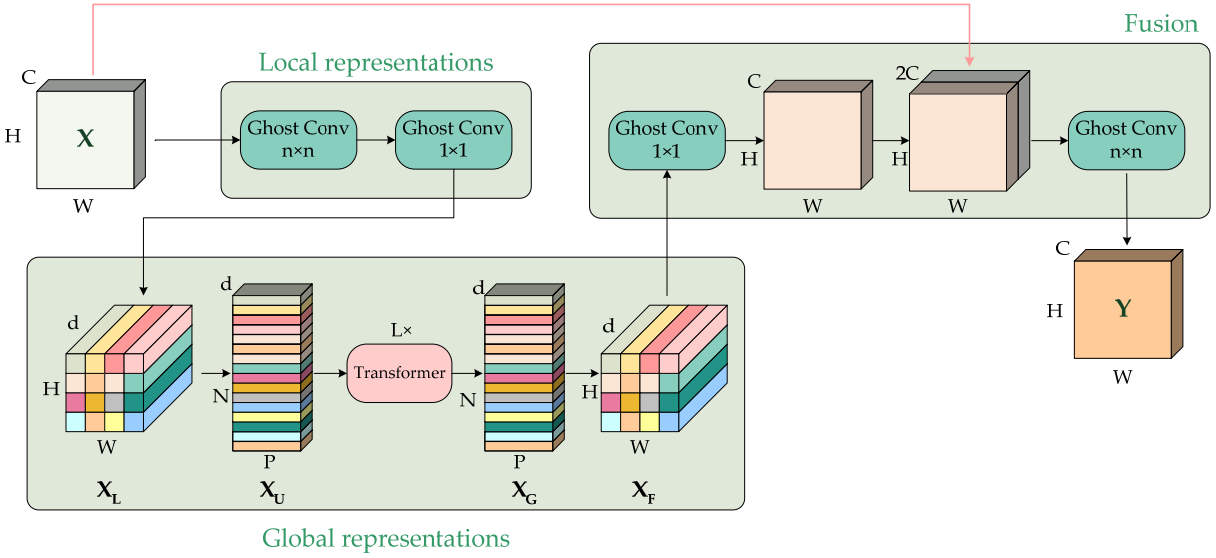

Figure 8. Ghost–MobileViT Block.

The operational process of the Ghost–MobileViT Block is divided into three stages: extraction of local representations, extraction of global representations, and feature fusion. For the input feature map X of size $C \times H \times W$, the Ghost–MobileViT Block first extracts the local information using the Ghost convolution layer of $n \times n$, and then maps X to the d-dimensional space using the Ghost convolution layer of 1×1 to get X_L, where $d > C$. After entering the global representation extraction phase, X_L is first expanded to contain N

non-overlapping flattened patches X_U. Where $P = wh, N = HW/P, w, h$ are the width and height of each patch and $w \leq n, h \leq n$. X_U obtains the global representation between the patches using the transformer to obtain X_G. The process can be expressed as follows:

$$X_G(p) = \text{Transformer}(X_U(p)), 1 \leq p \leq P \quad (7)$$

X_G is then collapsed to get X_F, which preserves the patch order and the spatial order of pixels within each patch and does not lose the pixel spatial order as ViT does. In the fusion phase, X_F is mapped to low C-dimensional space using 1×1 Ghost convolution and is combined with X by the concatenation operation. Finally, these concatenated features are fused using $n \times n$ Ghost convolution.

3.3. JR-TFViT

The JR-TFViT takes the time–frequency domain data of radar jamming signal as input and, after a series of Ghost Block, Ghost–MobileViT Block, and other computing modules, extracts the local information from the time–frequency domain data of radar jamming signals and fuses the global representation before finally outputting the jamming recognition results. Based on the Ghost Block and Ghost–MobileViT Block, the complete framework of the JR-TFViT for radar jamming identification proposed in this paper is shown in Figure 9, where C_i, $i = 0, \ldots, 9$ indicates the number of channels in the current module and the values are given in Table 2. The input JR-TFViT time–frequency image size is $3 \times 256 \times 256$. Firstly, the time–frequency image is downscaled by Ghost convolution with stride = 2 and 3×3 size to obtain a feature map of $C_0 \times 128 \times 128$. The feature map size is $C_3 \times 32 \times 32$ after three groups of Ghost Blocks. Next, the Ghost–MobileViT Block models local and global information in the input feature map. After two iterations of the Ghost Block, the Ghost–MobileViT Block is used to obtain a $C_8 \times 8 \times 8$ feature map. Finally, a set of global pooling and fully connected layers are used to output the jamming class probability of this time–frequency image.

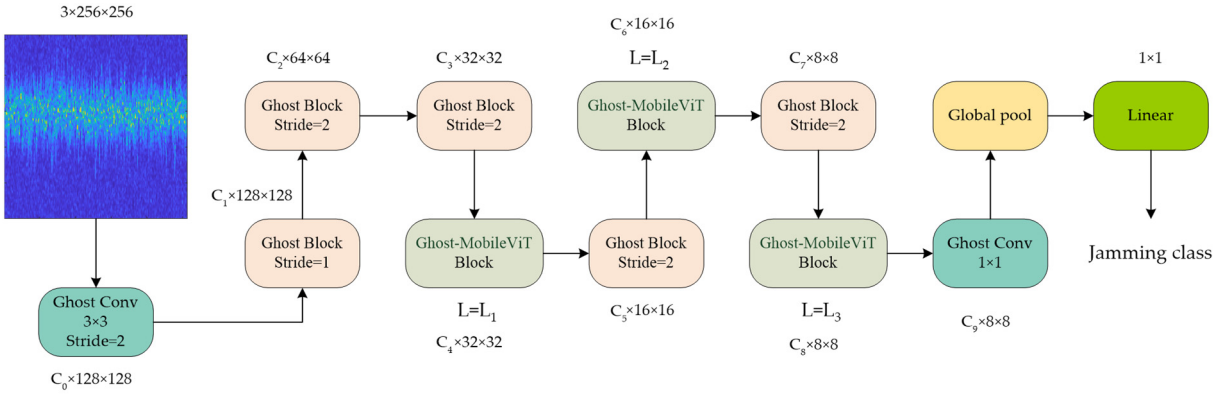

Figure 9. The JR-TFViT structure framework for radar jamming signal recognition.

Table 2. Parameter settings of JR-TFViT_S, JR-TFViT_M, and JR-TFViT_L.

	Params	C_i,i=0,…,9	L_i,i=1,2,3	**d**
JR-TFViT_S	0.67 M	[16,16,24,48,48,64,64,80,80,320]	[2,4,3]	[64,80,96]
JR-TFViT_M	1.5 M	[16,32,48,64,64,80,80,96,96,384]	[2,4,3]	[96,120,144]
JR-TFViT_L	3.66 M	[16,32,64,96,96,128,128,160,120,640]	[2,4,3]	[144,192,240]

4. Experiments and Results

4.1. Experiment Settings

In this paper, three different sizes of models were set up for training on the JR-TFViT, namely JR-TFViT_S, JR-TFViT_M, and JR-TFViT_L. The three network architectures remain

consistent with Figure 9 but different parameters are set. The number of parameters for the three networks, the number of output channels $C_i, i = 0, \ldots, 9$ for each module, the number of cascades $L_i, i = 1, 2, 3$ of transformers in the three Ghost MobileViT Blocks, and their internal spatial mapping dimension d are set as shown in Table 2. In addition, three mainstream CNNs (VGG16, ResNet50, and ResNet18) and three typical lightweight CNNs (Mobilenet v2, Mobilenet_v3_small, and Mobilenet_v3_large) were set up simultaneously for comparison experiments. The parametric quantities of the six compared networks are shown in Table 3.

Table 3. Comparison of network parameters.

	VGG16	ResNet50	ResNet18	Mobilenet v2	Mobilenet_v3_small	Mobilenet_v3_large
Params	134.91 M	23.53 M	11.18 M	2.24 M	1.53 M	4.22 M

The hardware environment built for the experiment includes an Intel Xeon Gold 6246 CPU, 256 GB RAM, and an NVIDIA Quadro GV100 graphics card; the software platform is the Windows 10 operating system, python 3.6.13, Pytorch 1.8.1 as the network model building framework, PyCharm 2019.3.5 (Community Edition) as the compiler, and CUDA 11.1.

The main hyperparameters of all networks in the training process are set as follows: training epochs, 128; batch size, 32; Stochastic Gradient Descent (SGD) selected as the optimizer; weight decay coefficient of 0.00005; initial learning rate of 0.001.

4.2. Evaluation Metrics

In order to evaluate the effectiveness of the JR-TFViT's jamming recognition, Overall Accuracy (OA), Kappa coefficient K, and F1 score were used as evaluation metrics to assess the effectiveness of recognition of different networks. *OA* is the ratio of the number of jamming samples correctly predicted by the recognition network on the test set to the total number of jamming samples on the test set, which can directly reflect the proportion of correct classifications. The expression of *OA* calculation is:

$$OA = \frac{n}{N} \times 100\% \tag{8}$$

The Kappa coefficient [26] performs an evaluation of bias for the recognition network, and the stronger the bias, the lower the Kappa value, defined as shown in Equation (9). The total number of test samples for all kinds of jamming is N. There are s classes of jamming samples, the number of test samples in each class is $t_i, i = 1, \ldots, s$, and the number of samples identified in each class is $p_i, i = 1, \ldots, s$.

$$kappa = \frac{OA - p_e}{1 - p_e},\ p_e = \frac{t_1 \times p_1 + t_2 \times p_2 + \ldots + t_s \times p_s}{N \times N} \tag{9}$$

The true category of jamming samples and the predicted category of the recognition network can be classified as true positive (TP), false positive (FP), true negative (TN), or false negative (FN). The definitions of Precision and Recall are shown in Equations (10) and (11). The F1 Score is used to take into account both the precision and recall of the model, also known as the balanced F-score, which is the summed average of recall and precision. The F1 Score is calculated as shown in Equation (12).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$F1 = 2 \cdot \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{12}$$

4.3. Results and Analysis

The OA of jamming recognition for the JR-TFViT and the other six comparison recognition networks under different JNR conditions are shown in Figure 10. Apparently, the nine recognition networks perform differently in their ability to cope with small sample training. The best performance of JR-TFViT_L is obtained when the number of training samples is 50, and OA can be maintained above 97% under different JNR conditions. ResNet50 performs close to JR-TFViT_L, but JR-TFViT_L has only 15% of the total number of parameters of ResNet50. ResNet18 and VGG16 perform preferably, but their number of parameters is still much higher than that of JR-TFViT_L. The performance of JR-TFViT_S and JR-TFViT_M lags behind that of JR-TFViT_L under the small sample training condition because of the small number of parameters, but the OA of JR-TFViT_S and JR-TFViT_M is still superior compared to that of Mobilenet series models with similar parameter quantities. This shows that the JR-TFViT structure not only has the advantage of being lightweight, but, with the ability to fuse global representations in the time–frequency domain of jamming, it can also obtain superior recognition performance compared to CNNs under small sample conditions.

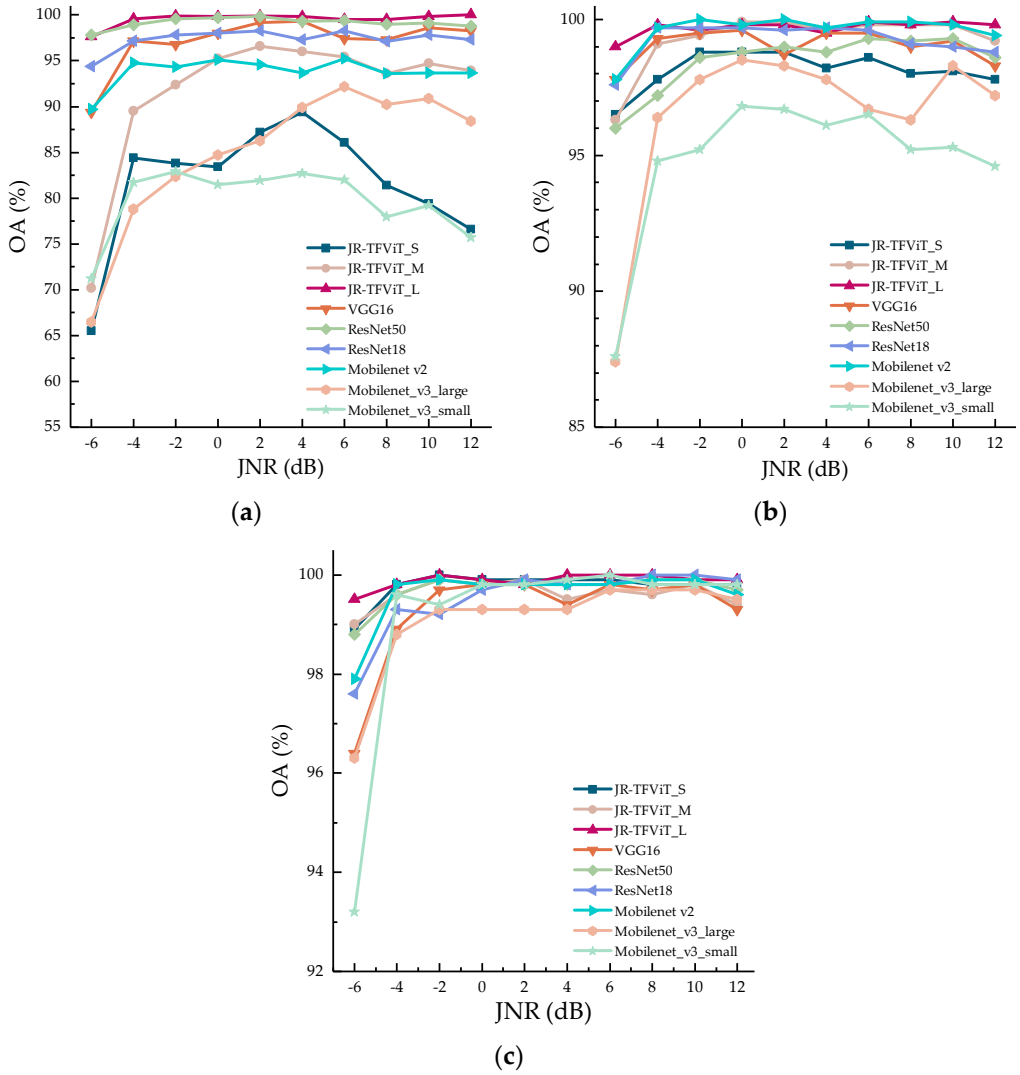

Figure 10. The OA of different recognition networks with different numbers of training samples: (**a**) 50 training samples; (**b**) 75 training samples; (**c**) 100 training samples.

The OA, Kappa, and F1 scores of the JR-TFViT and the six comparison recognition networks under different JNR conditions when the number of training samples is 100 are shown in Tables 4–6, respectively. According to the test results, when the training samples are sufficient, the OA of each recognition network's performance can reach more than 99% for the jamming scenario with JNR > −4 dB, and the OA of JR-TFViT_S, which has the lowest number of parameters at this time, reaches more than 99.8%. JR-TFViT_S has the same or even better recognition performance compared to various comparison networks, while the number of parameters for JR-TFViT_S is only 0.5% of VGG16 parameters and 2.8% of ResNet50 parameters. Compared to lightweight CNNs, the number of parameters for JR-TFViT_S is also only 43.8% of that of Mobilenet_v3_small. This indicates that the lightweight improvement method of the JR-TFViT is very effective in reducing the number of parameters for the model. Under the low JNR condition of JNR −4 dB, the OA of the CNN-based recognition network shows a significant decrease, while the JR-TFViT still maintains excellent recognition results. This is because the local information, such as image texture in the radar jamming time–frequency image, is affected by noise occlusion in the low JNR environment. In this case, the JR-TFViT extracts the global representation with the help of a transformer and, by fusing the local information of the time–frequency image with the global representation, a higher recognition accuracy can be achieved compared to recognition networks that rely solely on convolutional operations to obtain local information. The difference between the OA of JR-TFViT_S and JR-TFViT_L is only 0.6%, which indicates that adding global representation extraction capability can help identify the network to further reduce the parameters and make the JR-TFViT even lighter.

Table 4. The OA of the nine networks in different JNR conditions. The best accuracy is highlighted in bold.

Network	Params	JNR									
		−6	−4	−2	0	2	4	6	8	10	12
JR-TFViT_S	0.67 M	98.9	**99.8**	**100**	**99.9**	**99.9**	99.9	99.9	99.8	99.8	99.8
JR-TFViT_M	1.5 M	99	99.6	99.9	99.8	**99.9**	99.5	99.7	99.6	99.8	99.4
JR-TFViT_L	3.66 M	**99.5**	**99.8**	**100**	**99.9**	99.8	**100**	**100**	**100**	99.9	**99.9**
VGG16	134.91 M	96.4	98.9	99.7	99.8	99.8	99.4	99.8	99.7	99.8	99.3
ResNet50	23.53 M	98.8	99.6	99.9	99.8	99.8	99.8	99.8	99.9	99.9	99.7
ResNet18	11.18 M	97.6	99.3	99.2	99.7	**99.9**	99.8	99.8	**100**	**100**	**99.9**
Mobilenet v2	2.24 M	97.9	99.8	99.9	99.8	99.8	99.8	99.8	99.9	99.9	99.6
Mobilenet_v3_small	1.53 M	96.3	98.8	99.3	99.3	99.3	99.3	99.7	99.7	99.7	99.5
Mobilenet_v3_large	4.22 M	93.2	99.6	99.4	99.8	99.8	99.9	**100**	99.8	99.8	99.8

Table 5. The Kappa of the nine networks in different JNR conditions.

Network	Params	JNR									
		−6	−4	−2	0	2	4	6	8	10	12
JR-TFViT_S	0.67 M	98.8	**99.8**	**100**	**99.9**	**99.9**	99.9	99.9	99.8	99.8	99.8
JR-TFViT_M	1.5 M	98.9	99.5	99.9	99.8	**99.9**	99.5	99.6	99.5	99.7	99.4
JR-TFViT_L	3.66 M	**99.5**	**99.8**	**100**	**99.9**	99.8	**100**	**100**	**100**	99.9	**99.9**
VGG16	134.91 M	96.1	98.8	99.6	99.8	99.8	99.4	99.7	99.6	99.8	99.2
ResNet50	23.53 M	98.6	99.5	99.9	99.7	99.7	99.8	99.8	99.9	99.9	99.6
ResNet18	11.18 M	97.4	99.3	99.1	99.6	**99.9**	99.8	99.8	**100**	**100**	**99.9**
Mobilenet v2	2.24 M	97.7	99.7	99.9	99.8	99.7	99.7	99.8	99.9	99.9	99.5
Mobilenet_v3_small	1.53 M	96	98.6	99.2	99.2	99.2	99.2	99.6	99.6	99.6	99.5
Mobilenet_v3_large	4.22 M	92.6	99.5	99.4	99.7	99.8	99.9	**100**	99.8	99.8	99.8

Table 6. The F1 score of the nine networks in different JNR conditions.

Network	Params	JNR									
		−6	−4	−2	0	2	4	6	8	10	12
JR-TFViT_S	0.67 M	0.989	**0.998**	**1**	**0.999**	**0.999**	0.999	0.999	0.998	0.998	0.998
JR-TFViT_M	1.5 M	0.99	0.995	0.999	0.998	**0.999**	0.995	0.997	0.996	0.997	0.994
JR-TFViT_L	3.66 M	**0.995**	**0.998**	**1**	**0.999**	0.998	**1**	**1**	**1**	0.999	**0.999**
VGG16	134.91 M	0.962	0.989	0.997	0.998	0.997	0.995	0.997	0.997	0.998	0.993
ResNet50	23.53 M	0.987	0.996	0.999	0.997	0.998	0.998	0.998	0.999	0.999	0.997
ResNet18	11.18 M	0.975	0.992	0.992	0.995	**0.999**	0.998	0.998	**1**	**1**	**0.999**
Mobilenet v2	2.24 M	0.978	0.997	0.999	0.998	0.997	0.997	0.998	0.999	0.999	0.996
Mobilenet_v3_small	1.53 M	0.963	0.988	0.992	0.993	0.993	0.993	0.996	0.996	0.997	0.995
Mobilenet_v3_large	4.22 M	0.93	0.996	0.994	0.997	0.998	0.999	**1**	0.998	0.998	0.998

In addition to OA and Kappa under different JNR conditions, confusion matrices are also widely used for the analysis of recognition ability in multiclassification problems. The recognition accuracy of JR-TFViT_S at JNR = −6 dB is 98.9%, and the confusion matrix is shown in Figure 11, where the horizontal axis is the true type of jamming and the vertical axis is the type of jamming predicted by JR-TFViT_S. J1-J12 represent AJ, BJ, SJ, DDJ, VDJ, ISRJ, SMSP, DDJ + ISRJ, DDJ + SMSP, VDJ + ISRJ, VDJ + SMSP, and ISRJ + SMSP. From the confusion matrix, it can be seen that a part of DDJ + SMSP is incorrectly recognized as DDJ + ISRJ, leading to a decrease in the accuracy of JR-TFViT_S recognition. This is because the spoofing distance parameter of DDJ is randomly generated and DDJ will overlap with SMSP or ISRJ. In addition, the masking of noise when the JNR is too low threatens the effectiveness of the recognition network, making it easy to confuse these two types of composite jamming. However, under the condition of JNR = −6 dB, the accuracy of the three JR-TFViT networks still outperforms other CNNs.

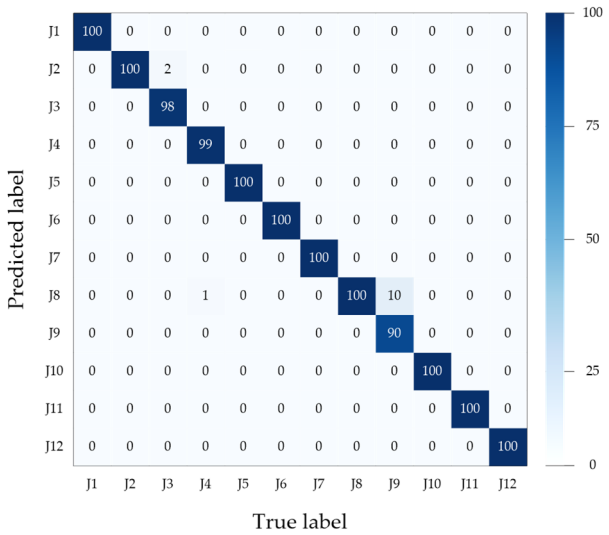

Figure 11. JR-TFViT_S confusion matrix at JNR = −6 dB.

5. Conclusions

In this paper, a lightweight radar jamming recognition network (JR-TFViT) is presented which consists of a transformer cascaded with Ghost convolution modules. Compared with CNN-based jamming recognition networks, it can focus on the global representation in the jamming time–frequency domain data. By fully fusing the global representation and local information of the jamming time–frequency domain data, an excellent feature extraction capability can be achieved using a small number of network parameters. At the same time, the Ghost convolution module is used in the JR-TFViT instead of a standard convolutional

operation to further reduce the number of parameters in the JR-TFViT; the number of parameters in the lightest JR-TFViT_S is only 0.67 M. The recognition experiments for 12 typical jamming techniques show that the recognition performance of the JR-TFViT is better than that of mainstream CNNs, and recognition accuracy under low JNR conditions is especially better than that of the six comparison networks.

Further study to achieve accurate recognition of more types of radar jamming and deployment testing of jamming recognition networks on actual equipment will be the key research directions for subsequent work. However, in real scenarios, the types and compound patterns of radar jamming are more complex and variable.

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

Funding: This research was funded by Natural Science Funding of Shaanxi Province, grant number 2021JM-222.

Data Availability Statement: Data are contained within the article. The data presented in this study are available in the article.

Acknowledgments: The authors thank all the reviewers and editors for their great help and useful suggestions.

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

References

1. Tan, M.; Wang, C.; Xue, B.; Xu, J. A Novel Deceptive Jamming Approach Against Frequency Diverse Array Radar. *IEEE Sens. J.* **2021**, *21*, 8323–8332. [CrossRef]
2. Li, N.J.; Zhang, Y.T. A survey of radar ECM and ECCM. *IEEE Trans. Aerosp. Electron. Syst.* **1995**, *31*, 1110–1120.
3. Futong, Q.; Jie, M.; Jing, D.; Fujiang, A.; Ying, Z. Radar jamming effect evaluation based on AdaBoost combined classification mode. In Proceedings of the 2013 IEEE 4th International Conference on Software Engineering and Service Science, Beijing, China, 23–25 May 2013.
4. Gao, M.; Li, H.; Jiao, B.; Hong, Y. Simulation research on classification and identification of typical active jamming against LFM radar. In Proceedings of the Eleventh International Conference on Signal Processing Systems, Chengdu, China, 15–17 December 2019.
5. Su, D.; Gao, M. Research on Jamming Recognition Technology Based on Characteristic Parameters. In Proceedings of the 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 23–25 October 2020.
6. Zhao, J.; Hu, T.; Zheng, R.; Ba, P.; Mei, C.; Zhang, Q. Defect Recognition in Concrete Ultrasonic Detection Based on Wavelet Packet Transform and Stochastic Configuration Networks. *IEEE Access.* **2021**, *9*, 9284–9295. [CrossRef]
7. Thayaparan, T.; Stankovic, L.; Amin, M.; Chen, V.; Cohen, L.; Boashash, B. Time-Frequency Approach to Radar Detection, Imaging, and Classification. *IET Signal Process.* **2010**, *4*, 325–328. [CrossRef]
8. Liu, Q.; Zhang, W. Deep learning and recognition of radar jamming based on CNN. In Proceedings of the 2019 12th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 14–15 December 2019.
9. Qian, J.; Teng, X.; Qiu, Z. Recognition of radar deception jamming based on convolutional neural network. In Proceedings of the IET International Radar Conference, Online, 4–6 November 2021.
10. Lv, Q.; Quan, Y.; Feng, W.; Sha, M.; Dong, S.; Xing, M. Radar Deception Jamming Recognition Based on Weighted Ensemble CNN With Transfer Learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5107511. [CrossRef]
11. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *60*, 1097–1105. [CrossRef]
12. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
13. Peng, Z.; Huang, W.; Gu, S.; Xie, L.; Wang, Y.; Jiao, J.; Ye, Q. Conformer: Local Features Coupling Global Representations for Visual Recognition. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021.
14. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Online, 3–7 May 2021.
15. Si, C.; Yu, W.; Zhou, P.; Zhou, Y.; Wang, X.; Yan, S. Inception Transformer. *arXiv* **2022**, arXiv:2205.12956.

16. Mehta, S.; Rastegari, M. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. *arXiv* **2021**, arXiv:2110.02178.
17. Zhang, H.; Yu, L.; Chen, Y.; Wei, Y. Fast Complex-Valued CNN for Radar Jamming Signal Recognition. *Remote Sens.* **2021**, *13*, 2867. [CrossRef]
18. Campbell, F.W.; Robson, J.G. Application of Fourier analysis to the visibility of gratings. *Physiology* **1968**, *197*, 551–556. [CrossRef] [PubMed]
19. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; Feng, J. Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
20. Park, N.; Kim, S. How Do Vision Transformers Work. *arXiv* **2022**, arXiv:2202.06709.
21. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. *arXiv* **2021**, arXiv:2108.11539.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
24. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
25. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
26. Sumaiya, M.N.; Kumari, R.S.S. Logarithmic Mean-Based Thresholding for SAR Image Change Detection. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1726–1728. [CrossRef]