IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 13, 2018, accepted December 24, 2018, date of publication January 10, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891594

Radar HRRP Target Recognition Based on Deep One-Dimensional Residual-Inception Network

CHEN GUO, YOU HE, HAIPENG WANG, TAO JIAN, AND SHUN SUN
Institute of Information Fusion, Naval Aviation University, Yantai 264001, China

Corresponding author: Haipeng Wang (whp5691@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61471379, Grant 61790551, Grant 61102166, and Grant 61671463.

ABSTRACT A novel radar target recognition method based on the deep one-dimensional residual-inception network is proposed for a high-resolution range profile (HRRP). The traditional methods based on shallow models can hardly extract the complete information of the targets HRRP from different angles. The deep models, such as sparse autoencoder, have been adopted to solve this problem. However, these deep models with a huge amount of parameters require more training samples to guarantee the generalization performance. To solve the above-mentioned problem, a model based on the one-dimensional convolutional kernel and a pooling layer is proposed. It is worth noting that the one-dimensional convolutional kernel and pooling operation have the potential to overcome the translation sensitivity and target aspect sensitivity of the HRRP, and both of them can greatly reduce the parameters and improve the generalization performance of the model. In addition, a new loss function is proposed to further enhance the separability of features. The experimental results show that compared with other four deep models, the proposed model can achieve a good performance in recognition accuracy and robustness.

INDEX TERMS Radar target recognition, high-resolution range profile, one-dimensional residual-inception network, cosine center loss.

I. INTRODUCTION

The high-resolution range profile (HRRP) is one-dimensional projection of the target in the radar observation direction obtained by wideband radar (the resolution of which is much smaller than the target size), which reflects abundant information of the scatterers contained in the target, such as the distribution of radar cross section (RCS) along the observation direction of the radar, the structure and intensity of the scatterers etc [1]. Compared with the two-dimensional range profile, HRRP is much easier to obtain, process and store, therefore it has become a hotspot of radar automatic target recognition.

How to extract the separable features of the targets has become a key point in radar target recognition. There are a large number of feature extraction methods for HRRP, such as radar HRRP target recognition method based on scattering centers matching [2]–[5], information extraction method of target scatterers by super-resolution algorithm [6], feature dimension reduction method with manifold learning (ML) [7], noise robust feature extraction methods with dictionary learning(DL) [8]–[12], feature extraction kernel methods [13], [14], feature learning method using privilege information [15], etc. Features extracted by these shallow linear structures cannot effectively represent the complete information of the targets, which are designed artificially and require sufficient prior knowledge to support. Therefore, feature extraction methods based on deep non-linear networks have been adopted for radar automatic target recognition.

The essence of target recognition based on deep learning is feature extraction using deep networks. At present, some HRRP target recognition methods based on deep learning have been proposed, which will be introduced as below. A recognition framework based on t-distributed Stochastic Neighbor Embedding (t-SNE) and discriminant deep belief network (DDBN) is proposed in [16]. First, the t-SNE algorithm is used to segment the data, then, adopt the DDBE model for target recognition. By this method, problem of target recognition with unbalanced data sets is solved. In [17], a discriminative deep auto-encoder (DDAE) is proposed to improve the recognition accuracy with limited data samples.

In [18], a deep network called Stacked Corrective Autoencoder (SCAE) model is proposed to solve the problem of HRRP target recognition. According to the angular adjacent similarity characteristics of HRRP, the average profile of the HRRP in the adjacent angular domain is used as a corrective output for a stacked autoencoder. The covariance matrix of each HRRP is employed to form a loss function based on the Mahalanobis distance. The importance of the model depth has been proved by experiment. In [19], a HRRP recognition model combining sparse denoising autoencoder (SDAE) and multi-layer perceptron (MLP) is proposed. SDAE is used to automatically extract features without the label information of the training samples. The MLP is used for the final classification. The robustness of the model is very good due to the noise added during the training. The robust variational autoencoder proposed in [20] can better extract the robustness features of the targets. In [21], a model combining stacked autoencoder(SAE) and extreme learning machine is proposed. The extreme learning machine is adopted to replace the traditional back propagation network which greatly reduces the amount of the training parameters and improves the learning speed and generalization performance of the model. In [22], deep convolutional neural network (CNN) is employed to extract features from HRRP. The output of the convolutional neural network is regarded as probability of each class, and then average the outputs obtained from different monostatic and bistatic radar pairs to get the global target probabilities. Finally, compare the average probability with a selected threshold for final classification. A good recognition accuracy rate can be obtained by this method.

At present, the existing HRRP feature extraction methods using deep learning are mainly based on autoencoder model and its variants. The autoencoder is composed of two basic parts: encoder and decoder. The encoding is the process of mapping the input to the hidden layer (the output of the hidden layer can be regarded as the feature extracted by the autoencoder). When the number of neurons in the hidden layer is smaller than the input data dimension, the encoding can be regarded as a dimensionality reduction operation, similar to principal component analysis (PCA), but is a non-linear operation. The larger the number of neurons in the hidden layer is, the more information of HRRP will be extracted by the model, but the training difficulty also increases. As we all know, the HRRP has a characteristic of the target aspect sensitivity. That is, the HRRPs of the same target may be very different from each other, which lead to the complexity of the feature extraction models. The complexity of the autoencoder model can be increased by increasing the number of hidden layers or the depth of the model. But in the non-cooperative circumstance, the insufficient training data may lead to the overfitting of a complex autoencoder model. Therefore, the methods based on autoencoder models are not appropriate for full angle domain HRRP target recognition.

In order to solve the target recognition problem of HRRP in the full-angle domain, a method based on deep one-dimensional residual-inception network is proposed. The main contributions are as follow:

1 A deep one-dimensional residual-inception network based on one-dimensional convolutional kernel is proposed. The convolutional kernel has the two main characteristics: weight sharing, convolutional kernels in different scales can extract features with different fineness. In this paper, the model is designed to extract the full-angle complex features of HRRP with fewer parameters by applying these two features, which greatly improves the training efficiency.

2 A new loss function named cosine center loss is proposed to improve the separability of the features and the recognition accuracy of the model.

It's worth mentioning that the proposed model can recognize the target without angle domain division and template matching.

The rest of this paper is organized as follows: in Section 2, the HRRP data used in this paper and the one-dimensional convolutional network are described. In Section 3, the structure of the proposed model and the corresponding new loss function are presented in detail. In Section 4, several experiments are performed on the simulated HRRP dataset. Finally, we conclude our work in Section 5.

II. DESCRIPTION OF THE HRRP DATA AND THE ONE-DIMENSIONAL CONVOLUTIONAL NETWORK

This section gives a brief description of the HRRP data and the basic convolutional neural network.

A. DESCRIPTION OF THE HRRP DATA

HRRP is the amplitude of the coherent summations of the complex time returns from target scattering points in each range cell [1]. When the radar resolution is much smaller than the target size, the target occupies multiple radar range cells. The echo of each radar range cell is the superposition of radar echoes of all the scattering points contained in the corresponding cell. Usually, we use the scattering point model to describe the HRRP of the target, the radar echoes of the mth range cell can be described approximately by

$$x_n(m) = \sum_{i=1}^{I_n} \sigma_{n,i} \exp\left[-j\left(\frac{4\pi R_{n,i}(m)}{\lambda} + \theta_{n,i}\right)\right] \quad (1)$$

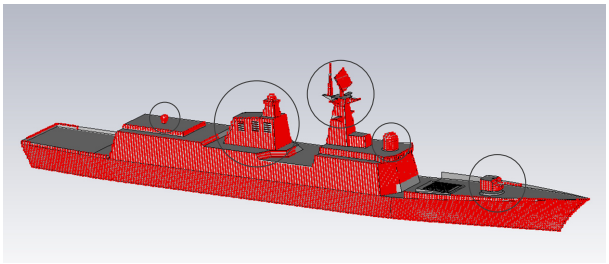
where, I_n denotes the number of target scattering points in the range cell, $R_{n,i}(m)$ denotes the distance between radar and the ith scattering point in the mth sampled echo, $\sigma_{n,i}$ and $\theta_{n,i}$ denote the amplitude and initial phase of the ith scattering echo, respectively. HRRP data can be expressed as

$$x_n = [|x_n(1)|, |x_n(2)|, \cdots, |x_n(D)|] \quad (2)$$

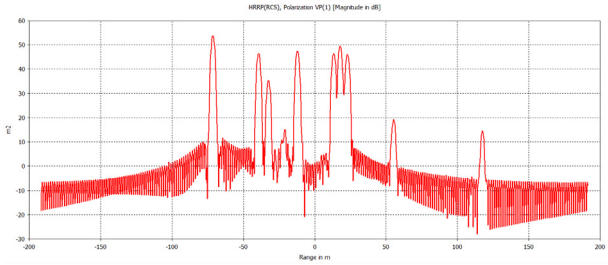
where, D is the dimension of the HRRP data.

The HRRP of the ship target is simulated by the CST software. The ship model and simulation result are shown in Fig. 1.

As shown in Fig. 1 (a), the red part of the ship represents the scattering surface of the ship in the direction of radar



(a)



(b)

FIGURE 1. Scattering surface of the ship and its corresponding simulated RCS at incident angle 45ř (a) The red part of the ship model represents the scattering surface of the ship (b) The equivalent RCS of each range cell of HRRP data(unit dB).

incident angle. For ship targets, the strong scattering points marked with circles in the Fig. 1 (a) are mainly concentrated in the hull and the bulges on the deck and the RCS shown in Fig. 1 (b) of the corresponding echoes are stronger than other parts, which proves that the echoes generated by the CST software is valid.

B. DESCRIPTION OF ONE-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORKS

Deep convolutional neural networks (CNN) have achieved great success in the field of computer vision [23]–[27]. To our best knowledge, CNNs and their variants have achieved the state of the art performance in all aspects in target recognition. The features extracted by the CNN model have the characteristic of translation invariance and rotation invariance, which is suitable to solve problem of translation sensitivity and aspect sensitivity of the HRRP. The basic structure of CNN mainly consists of five parts, namely input layer, convolution layer, pooling layer, fully connected layer and output layer. The functions of these five parts will be explained as below.

1) INPUT LAYER

The length of input data should be fixed. The data preprocessing at the input layer is divided into two steps as below:

1 Normalize the amplitude of the HRRP. Set $x(n)$ as the nth HRRP sample and its amplitude is normalized as $\bar{x}(n) = \frac{x(n)}{\max(|x(n)|)}$.

2. Calculate the mean value $\bar{x}_m = \frac{1}{N}\sum_{n=1}^{N}\bar{x}(n)$ of $\bar{x}(n)$, perform the subtraction operation for all $\bar{x}(n)$, that is $\bar{x}(n)-\bar{x}_m$, $n = 1, 2, \ldots, N$, where N is the number of all training data.

Data preprocessing at the input layer can reduce impact of the amplitude disturbance on the target recognition performance and improve the robustness of the model.

2) CONVOLUTIONAL LAYER

The function of the convolutional layer is to automatically extract the features of the target. The traditional methods recognize the target on the basis of effective segmentation in angle domain according to the adjacent angle similarity characteristic of the HRRP, and extract appropriate template of each angle domain to match with. These methods can have a good recognition performance but has little adaptability to overcome the translation sensitivity and target aspect sensitivity of the HRRP. The convolutional layer extracts feature by using convolution kernel as a sliding window which is correlated with corresponding portion of the input feature vector with a certain stride. The convolution kernels have no certainty which part of the HRRP to match with before training, but will automatically extract the interested features of the HRRP for target recognition after training.

The specific operation process in the convolutional layer is as follows. First, perform correlation operation between the input feature vector and the convolution kernel of each channel (The length of convolution kernel is usually set as $(2n + 1, 1), n = 0, 1, \ldots, N$). Then, obtain the output feature vector through an activation function. The mathematical expression of the convolution operation is as follows:

$$\begin{aligned} x_j^l &= f(u_j^l) \\ u_j^l &= \sum_{i \in M_j} x_i^{l-1} \otimes k_{ij}^l + b_j^l \end{aligned} \tag{3}$$

where, u_j^l is raw activation of the jth channel of the convolutional layer l, x_j^l is the output of the jth channel of the convolutional layer l. $f(\cdot)$ is the activation function, which employs ReLU function. k_{ij}^l is convolution kernel vector of the jth channel of the convolutional layer l corresponding to the ith input vector. b_j^l is the bias of the jth channel of the convolutional layer l, $\otimes$ represents the convolution operation. Assuming the length of vector is $n \times 1$, the length of the convolution kernel is $c \times 1$, and the stride is t, the length of the output vector is $(n - c + 1)/t$ with no zero-padding.

Each convolutional layer contains a plurality of convolution kernels, and the number of output feature vectors is the same as the number of convolution kernel channels in each layer, that is, each output feature vector is corresponding to one convolution kernel channel.

3) POOLING LAYER

The function of the pooling layer employed in the two-dimensional convolutional neural network is to remove the redundant information of the features extracted by the convolutional layers. For HRRP, when the resolution of radar is high enough, the adjacent points of the feature vector will be redundant and require pooling operation to ensure its sparsity. The translation sensitivity of HRRP can also be solved by

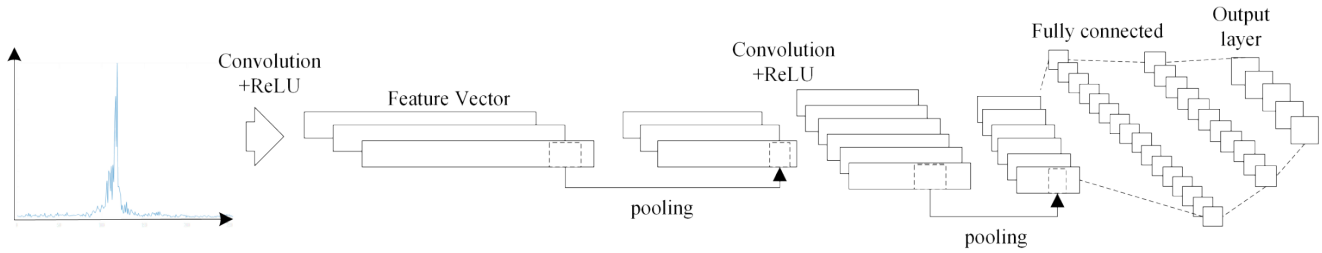


FIGURE 2. A simple one-dimensional CNN model.

the pooling layer. Pooling operation can be expressed by the following formula:

$$x_j^l = \beta_j^l \text{down}(x_j^{l-1}) \tag{4}$$

where, u_j^l is the raw activation of the jth channel of pooling layer l, x_j^l is the output of the jth channel of the pooling layer l. β_j^l is the weight of the jth channel of pooling layer l and is usually set to 1, down$(\cdot)$ represents the pooling function. To better understand the pooling operation, we can also regard it as a window sliding on the input vector with a certain stride. The output vector can be obtained by calculating the sum, average or maximum value in the window. Assuming the dimension of the input vector is $n \times 1$, the length of the window is $p \times 1$, the stride is t, the length of the output vector is $(n-p+1)/t$ with no zero-padding of the feature vector. p usually equals to t.

4) FULLY CONNECTED LAYER

The fully connected layer is an ordinary neural network located between the pooling layer and the output layer in a convolutional neural network. The input of the first fully connected layer is the expansion and splicing of all the feature vectors from the upper layer. Fully connected operation can be expressed by the following formula:

$$x^l = f(u^l) = f(W^l x^{l-1} + b^l) \tag{5}$$

where, u^l is the raw activation of the fully connected layer l, x^l is the output of the fully connected layer l. $f(\cdot)$ is the activation function, usually using the ReLU function, W^l is the weight matrix of the fully connected layer, b^l is the bias term. The function of the fully connected layer is to map the feature vector extracted by the convolution layer to the output layer. A convolutional neural network may contain one or more fully connected layers.

5) OUTPUT LAYER

The number of neurons in the output layer is the same as the number of class. The output of each neurons is usually normalized by the softmax function to ensure the sum be 1. Therefore, the output of each neuron can be regarded as the probability of the corresponding class. In theory, the ith neuron corresponding to the ith class is activated to be 1, and the other neurons output are 0. In practice, the class corresponding to the largest output value of the neuron is selected as the recognition result.

The structure of a simple one dimensional CNN model is shown in Fig. 2.

As shown in Fig. 2, the model contains two convolutional layers, two pooling layers and two fully connected layers, the dimension of the feature vector is halved by the pooling layer, the number of the feature vector is determined by the number of the kernel channels. In Fig. 2, the number of the kernel channels in first and second convolutional layer is 3 and 6 respectively.

III. DEEP RESIDUAL INCEPTION NETWORK

This section gives a detailed description of the proposed model and contains four parts: 1. structure and function of residual-inception block, 2. structure and function of inception-pooling layer, 3. detailed description of the proposed loss function. 4. the structure of the proposed model.

A. DESCRIPTION OF RESIDUAL-INCEPTION BLOCK

In this section, we give a detailed description of the residual block and the inception layer, then combine them to form the residual inception block.

1) DESCRIPTION OF RESIDUAL BLOCK

Deep convolutional neural networks can extract features from different levels. The deeper the layers are, the richer the extracted features and the higher the recognition accuracy are [23]. However, for the traditional convolutional neural network, as the depth of the network increases, non-convergence of the loss function may emerge because of the disappearance or explosion of the gradient. We also call this phenomenon network degradation. In order to prevent the degradation of the deep convolutional neural network, we employed the residual network to extract features [26]. The residual block is an important part of the residual network, which schematic diagram is shown in the Fig. 3.

The residual block is composed of convolutional layer (Conv represents convolutional layer in all figures). In order to avoid the degeneration of the network, the number of convolutional layers in the residual block should be limited and usually set to two. The output of the residual block is the sum of the input vector and the output vector from the last

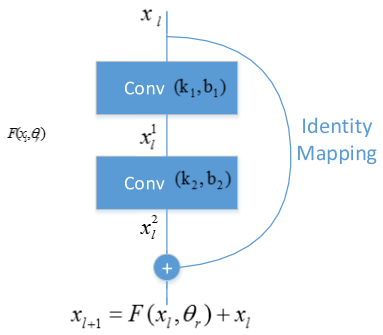


FIGURE 3. Schematic diagram of residual block.

convolutional layer. The plus sign in the Fig. 3 represents that the elements of the input and output feature vector are added one by one. The expression of the residual block is

$$x^{l+1} = F(x^l, \theta_r) + W_r x^l \tag{6}$$

where, $x^l \in \mathrm{R}^{k\times m}$ represents the input feature vector, $x^{l+1} \in \mathrm{R}^{h\times m}$ represents the output feature vector of the residual block, k and h represent the number of input and output feature vectors respectively. Generally, the number of output feature vectors is twice the number of input feature vectors, that is $h = 2k$ [28]. m is the size of min-batch. θ_r represents the parameter set of the residual block, $\theta_r = \{k_1, k_2, b_1, b_2\}$, $W_r \in \mathrm{R}^{h\times k}$ is used to adjust dimension of the input vector so as to make it match the output vector in size, here we set $W_r = \begin{bmatrix} 1^{k\times k} \\ 0^{(h-k)\times k} \end{bmatrix}$.

2) CONSTRUCTION OF THE INCEPTION LAYER

Convolution kernels with different scales can extract features with different precisions. The smaller the scale of the convolution kernel is, the finer the extracted features are. In conventional CNN models, the kernel scales of different channels in the same layer are the same, which lead to the incompleteness of feature expression [23], [23], [29], [30]. Therefore, we designed two kinds of inception layers to solve this problem, and their structures are shown in Fig. 4.

We can see from Fig.4 that both the inception layer I and II contains four branches, and each branch contains one or more convolutional layers. The block named Conv with color yellow, orange and purple represent convolutional layer with kernel scale 1∗1, 3∗1 and 5∗1 respectively. The number in the Conv block represents the channel number of the convolutional kernel. The block named Pooling represents the pooling layer. (3∗1) represents the length of pooling window mentioned in section II.B.(3).

We number the branches from left to right No.1 to No.4. The kernel scale of branch No.2 and No.3 (from top to bottom) are increasing in inception layer I while decreasing in inception layer II. In common convolutional neural networks, the shallow layer with large scale convolution kernel is mainly responsible for extracting low-level features such as the outline of the target. Otherwise, the deep layer with

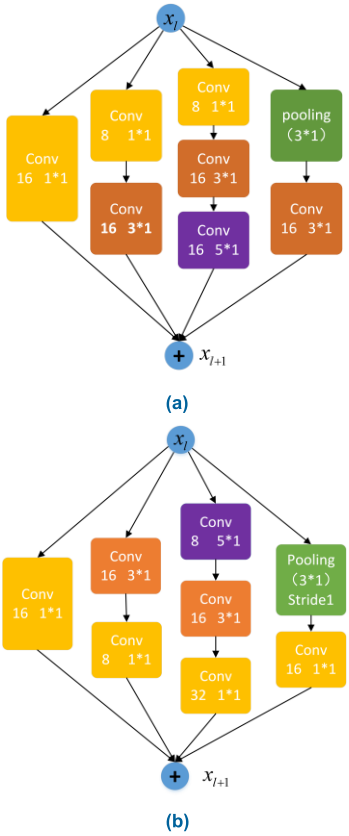


FIGURE 4. Two kinds of inception layer (a) inception layer I (b) inception layer II.

small scale convolution kernel is responsible for extracting the high-level features such as semantic information. Inception I and II are adopted as the shallow layer and deep layer respectively.

The final addition operation is not to sum or average the outputs of each branches, but to directly put together the features of each branch. Taking the inception layer I for example, the number of channels of the output feature extracted by inception layer I is $16 + 16 + 16 + 16 = 64$. Zero-padding is performed in every convolutional layer and pooling layer, that is, the dimension of the output vectors is the same with the input vectors.

The residual-inception block I and II can be obtained by replacing the convolutional layer with inception layer I and II respectively. The residual-inception block is shown in Fig. 5.

B. CONSTRUCTION OF INCEPTION-POOLING LAYER

Unlike the traditional pooling layer, the convolutional layer is adopted to down-sample the features by the inception-pooling layer. Two kinds of inception-pooling layers are designed in this section and shown in Fig. 6 (a), (b).

Both inception-pooling layer I and II contain three branches. Branch 1 and 2 only consist of convolutional layer. Branch 3 consists of pooling and convolutional layer. Each branch has only one layer with stride 2, therefore

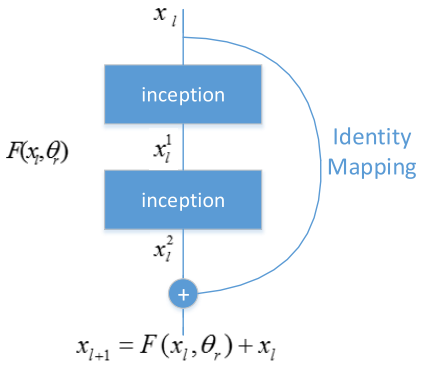


FIGURE 5. Residual-inception block.

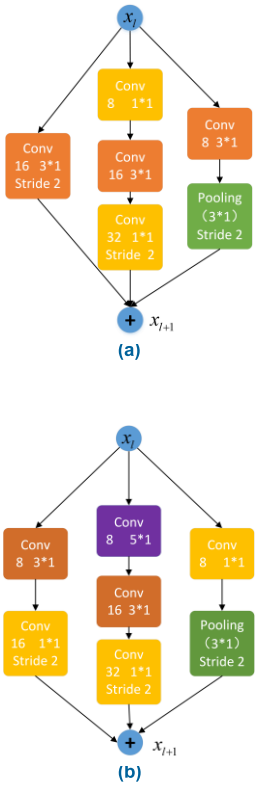


FIGURE 6. Two kinds of inception-pooling layer (a) inception-pooling layer I (b) inception-pooling layer II.

the dimension of the output vector is one half of the input vector. The channel number of the output vector is obtained by the same rule of inception layer. The function of inception-pooling layer is not only reducing the dimension of the feature vector but also extracting deeper features.

C. THE DESIGN OF THE LOSS FUNCTION BASED ON COSINE CENTER

Generally, the targets recognized by CNN are coarsely classified, such as cat, dog, flower, and bird, of which the features are very different from each other and easy to distinguish. However, features extracted from different ship targets are similar to each other and hard to distinguish because they belong to the same class. The traditional softmax-loss function is prone to cause the phenomenon that the intra-class distance even larger than the inter-class distance between features which lead to the unsatisfactory recognition result. Therefore, we design a new loss function which draws on the large margin cosine loss of the face recognition method [31]–[37] to meet the demand for ship target recognition. Considering the constraint of the intra-class distance of the feature, a loss function named cosine center loss (Referred to as CC) is proposed based on the feature center and cosine distance. During the training process, the loss function will punish the features with a large angle to the center of its corresponding class, so as to reduce the intra-class distance of the features and improve the separability of features. The expression of the loss function mentioned in this paper is as follows:

$$\begin{aligned} L &= L_{LMCL} + \lambda L_{center} \\ &= -\left(\begin{array}{l} \frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{s(\cos(\theta_{W_{y_i},x_i})-a)}}{e^{s(\cos(\theta_{W_{y_i},x_i})-a)} + \sum_{j\neq y_i} e^{s\cos(\theta_{W_j,x_i})}} \\ +\lambda \log \sum_{i=1}^{m} - \exp\alpha\left(\cos(\theta_{x_i,c_{y_i}})\right) \end{array} \right) \end{aligned} \tag{7}$$

where, m is the number of training samples. $x_i \in \mathrm{R}^d$ represents the feature of sample i extracted by the fully connected layer. y_i is the label of the sample i. $W_j \in \mathrm{R}^d$ denotes the jth column of the weight matrix $W \in \mathrm{R}^{d\times n}$ in the fully connected layer, which is corresponding to the class j, n is the number of classes, $\cos(\theta_{W_j,x_i})$ represents the cosine angle between the W_j and x_i. s and α are the hyperparameters which control the magnitude of the normalized features in L_{LMCL} and L_{center} respectively, so as to avoid the non-convergence of the loss function after feature normalization. a controls the magnitude of the cosine margin, $c_{y_i} \in \mathrm{R}^d$ is the feature center of the class corresponding to the label y_i of sample i, λ is the weight of the central loss term and is set to 2.

The loss function is divided into two parts, L_{LMCL} and L_{center}. L_{LMCL} is called Large Margin Cosine Loss (LMCL). As is known to all, the expression of the traditional softmax-loss is

$$L_S = \frac{1}{m}\sum_{i=1}^{N} -\log \frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{n} e^{W_j^T x_i}} \tag{8}$$

where $W_j^T x = \|W_j\| \|x\| \cos\theta_j$, $\cos\theta_j$ can be regarded as a measure of similarity between weight W_j and feature vector x_i. In order to make full use of the angle information, W and x are normalized. $W = W^*/\|W^*\|$, $x = x^*/\|x^*\|$, then $\cos(\theta_{W_j,x_i}) = W_j^T x_i$. As $\cos\theta_{W_j,x_i}$ is monotonically decreasing when $\theta_{W_j,x_i} \in [0,\pi]$. The smaller angle between the feature and the weight is, the larger $\cos\theta_{W_j,x_i}$ and similarity between feature and weight are. When $\cos(\theta_{W_{y_i},x_i}) > \cos(\theta_{W_j,x_i})(\forall, j = 1,2,\ldots n, j \neq y_i)$ is satisfied, the sample i belongs to the class y_i. To further enhance the constraint

between the inter-class distance, the condition changes to when $\cos(\theta_{W_{y_i},x_i}) - a > \cos(\theta_{W_j,x_i})$, $(\forall, j = 1, 2, \ldots n, j \neq y_i)$, the sample i belongs to the class y_i.

Ideally, the value range of a is $0 \leq a \leq 1 - \cos(\frac{2\pi}{n})$, where, n is the number of classes. If and only if the angle between the adjacent feature centers is the same and the angle between the feature of each sample and their corresponding feature center is 0, the equal sign is established. But in reality, the angle between the adjacent feature centers is not exactly the same, and the angle between the feature of the samples and their corresponding feature center is mostly nonzero. $a = 0.13$ is found to be the best value for the proposed model by simulation. $a > 0.13$ will lead to the overlap of features between different classes.

The parameter s solves the problem of non-convergence of the loss function. It satisfies the inequation $\log\left(1 + (m-1)e^{-\frac{m}{m-1}s}\right) \leq \varepsilon$, where, m is the number of samples in the min-batch. ε is a very small constant close to 0. We set s to be 30.

L_{center} is a loss function based on angle center. c is the normalized feature center whose initial expression is $c_j^* = \frac{1}{m_j}\sum_{i=1}^{m}\delta(y_i = j)x_i$, $c = c^*/\|c^*\|$. Where, $\delta(y_i = j)$ represents that when y_i belongs to class j, $\delta(\cdot)$ equal to 1, otherwise equal to 0, m_j denotes the number of samples contained in the class j in a min-batch. $\cos(\theta_{x_i,c_{y_i}}) = x_i^T c_{y_i}$ represents the cosine value of the angle between the feature x_i of the sample i and its corresponding feature center c_{y_i}. This function penalizes the sample with a large angle between the feature vector and the feature center. During the training process, we use the following formula to update the feature centers of each class:

$$\Delta c_j = \frac{\sum_{i=1}^{m}\delta(y_i = j)\cdot x_i e^{x_i^T c_j}}{1 + \sum_{i=1}^{m}\delta(y_i = j)} \tag{9}$$

The combined loss function with L_{LMCL} and L_{center} increases the inter-class angle distance while reduces the intra-class angle distance, which greatly improves the separability of the features.

D. CONSTRUCTION OF THE PROPOSED MODEL

The proposed model consists of common convolutional layer and pooling layer, the residual-inception block layer, the inception-pooling layer, and fully connected (FC) layer. The specific structure of proposed model is shown in Fig.7.

The number of residual inception blocks I and II are both set to three. The fully connected layer reduces the feature dimension to two, and its output is used as the input of L_{center}. Since there are seven types of ship targets to be recognized, the number of neurons in the output layer is set to be seven.

To better compare the difference of the proposed model and the comparison models, the detailed structure and parameter descriptions of the proposed model, CNN and SDAE are listed in Table 1, 2, 3.

We can see from the Table 1, 2, 3 that the parameters in model based on autoencoder far outnumber models based on convolutional kernel. The weight sharing property of the convolutional kernel can greatly reduce the number of parameters and avoid the overfitting.

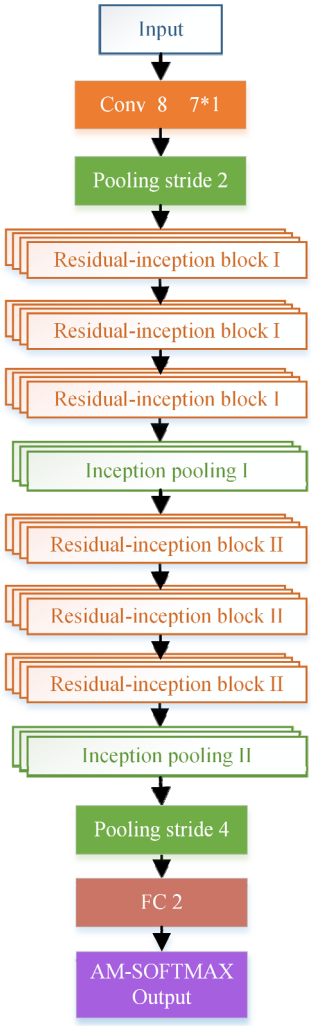


FIGURE 7. The structure of the proposed model.

TABLE 1. The main structure of proposed model.

NETWORK STRUCTURE	NUMBER OF PARAMETERS	OUTPUT SIZE
CONV,7*1 , 8	56	512*1*8
POOLING, STRIDE 2	0	256*1*8
RESIDUAL-INCEPTION BLOCK I *3	17042	256*1*64
INCEPTION-POOLING I	6016	128*1*56
RESIDUAL-INCEPTION BLOCK II*3	21120	128*1*80
INCEPTION-POOLING II	6748	64*1*56
POOLING, STRIDE 4	0	16*1*56
FULLY CONNECTED LAYER	1792	2
AM-SOFTMAX LAYER	14	7
TOTAL NUMBER	29674	

IV. SIMULATIONS AND ANALYSIS

A. CONSTRUCTION OF THE DATA SET

Most of the Ship targets are non-cooperative. It is difficult to establish the target HRRP dataset through measure data.

TABLE 2. The main structure of CNN.

NETWORK STRUCTURE	NUMBER OF PARAMETERS	OUTPUT SIZE
CONV,5*1 , 32	160	508*1*32
CONV,5*1 , 32	5120	504*1*32
POOLING, STRIDE 2	0	257*1*32
CONV,3*1 , 64	6144	128*1*64
CONV,3*1 , 64	12288	128*1*64
POOLING, STRIDE 2	0	64*1*64
CONV,1*1 , 128	8192	16*1*128
CONV,1*1 , 128	16384	16*1*128
POOLING, STRIDE 4	0	16*1*128
FULLY CONNECTED LAYER	2048	2
OUTPUT LAYER	14	7
TOTAL NUMBER	50350	

TABLE 3. The main structure of model based on autoencoder.

NETWORK STRUCTURE	NUMBER OF PARAMETERS	OUTPUT SIZE
HIDDEN LAYER I	307200	600
HIDDEN LAYER II	120000	200
HIDDEN LAYER III	10000	50
TOTAL NUMBER	437200	

A 1:1 target model is established by 3-dimensional software Solidworks, then import the model to CST electromagnetic simulation software to simulate the HRRP data. The parameters of the ships are shown in Table 4. There are several factors should be considered before simulation:

1) THE SHIP IS A KIND OF SUPER LARGE ELECTRIC TARGET

In the simulation, the CST makes extremely fine meshing of the ship which calls high performance for computer. The HRRP data has characteristics of similarities between adjacent angles (the HRRP data within a small angle range has small fluctuations, so they can be considered as the same data). According to the two factors, a larger angle step will be selected during the simulation process.

2) THE SHIP MODEL USED IN THIS PAPER IS SYMMETRICAL, SO ONLY THE HALF ANGLE RANGE OF THE TARGET IS SELECTED FOR SIMULATION.

Combining the above factors, the CST simulation parameters are set as follows: azimuth angle is $-90\sim90$ degrees, pitch angle is 0 degree, angle step is 1 degree. The parameters of the radar are shown in Table 2, the center frequency is 10GHz, the bandwidth is 100MHz, the polarization method is v, the number of frequency sampling points is 256. At last, the default optimal mesh size of the software is adopted. The ray tracing algorithm is used to conduct the simulation. Finally, HRRP data of the target at 181 azimuth angles are simulated by CST.

The proposed model has a high complexity and requires a large amount of data for training. Insufficient data may lead to over-fitting of the model. We use matlab2016b to process the data obtained by CST. The processing is as follows:

Step 1: Convert the CST simulated RCS data into HRRP of ship targets with 181 angles by the radar equation. Calculate the noise power according to the signal-to-noise ratio (SNR) set by the simulation, and add random Gaussian noise 15 times to each original noiseless HRRP data to expand

TABLE 4. Structure parameters of 7 kinds of ship targets.

NO.	LENGTH （M）	WIDTH （M）	DRAUGHT DEPTH （M）
SHIP 1	182.8	24.1	8.1
SHIP 2	172.8	16.8	6.5
SHIP 3	153.8	20.4	6.3
SHIP 4	135	16.8	4.5
SHIP 5	121	17.6	4.3
SHIP 6	102.2	16.5	4.2
SHIP 7	89.3	12.1	4.0

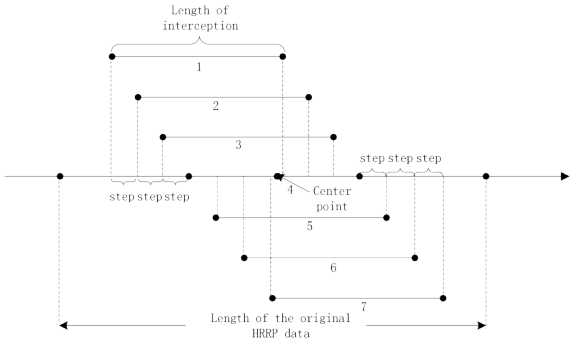


FIGURE 8. Interception of the HRRP.

the data. At this point, the data is still insufficient and needs to be further expanded.

Step 2: Intercept the original HRRP data. The length of the data from the above step is 1024, and the points containing more target information are symmetrically distributed around the center point which is the 512th point in this simulation. One point of the HRRP data corresponds to a radar distance unit of 0.3 meters. As shown in Table4, the maximum length of the target is 182 meters, equivalent to 606 distance units, that is, only 606 points out of 1024 points contain target information. At the same time, considering the pooling operation, the data will be halved by many times, the length of the intercepted data is preferably a power of 2. In summary, the length of the intercepted HRRP data is set to be 512. As shown in the Fig. 8, 7 HRRP fragments numbered from 1 to 7 are intercepted from the original one. The center point of the 4^{th} HRRP fragment and the original HRRP data are the same. The step is set to be 30. The 1^{st}, 2^{nd}, 3^{rd} HRRP fragment are obtained by left shifting the interception, whereas, 5^{th}, 6^{th}, 7^{th} HRRP fragment by right shifting the interception. By this method, seven fragments of HRRP data are intercepted from the original HRRP data which are partially coincident but not the same and the data obtained in step 1 is expanded by 7 times.

Through the above process, the total number of the dataset achieves 133035.

The experiments are mainly divided into two aspects. 1. Verify the recognition performance of the proposed model under different noise conditions. 2. Verify the ability of the proposed model to overcome the target aspect sensitivity problem of HRRP. In order to verify these two aspects, the above data set is divided into training set and test set according to different criteria.

In order to verify the recognition performance of the model under different SNR conditions, the dataset is generated according to the required SNR, and the same type of data is randomly disrupted within the class. Take 70% of each class as the training set, and the rest 30% as testing set. The dataset divided in this way is called the dataset A.

In order to verify the ability of the model to overcome the target aspect sensitivity problem of HRRP, it is necessary to ensure the angle range of the training set and the testing set do not completely coincide and the angle range of the training set needs to be smaller than that of the testing set. Therefore, we randomly select 70% of HRRP data corresponding to 150 random angles of each target class as the training set, and the remaining part as the testing set. The dataset divided in this way is called the dataset B.

After obtaining the training set, we preprocess the training data according to the method in the literature [18], assuming the number of HRRP data of the same target at the same angle is q, q group can be formed by selecting non repeating $q-1$ data with permutation and combination rule C_q^{q-1}. Calculate the average profile of each group to form the final training data set.

B. EXPERIMENTAL ENVIRONMENT AND PROCEDURE

Experiments are carried out in the 64-bit win7 system. The software is mainly based on deep learning architecture of Keras and python development environment Sublime Text 3. The hardware is based on Intel (R) Core (TM) i7-7700K @ 3.60GHz CPU and one NVIDIA GTX 1070 GPU, with CUDA8.0 accelerating computation. The training process of the proposed model is performed according to Table 5.

The well trained model can extract the features of the testing samples and output the corresponding recognition results automatically.

TABLE 5. The training process of the proposed model.

Input	Training samples
Output	The model for recognition
Step 1	Model construction. Construct the model according to Fig. 7. Initialize the parameters, such as the parameters $\theta_c = \{k_c, b_c\}$ of the convolution layer, the weight matrix W of the fully connected layer, and the feature center $\{c_j, j = 1, 2, ..., k\}$ of each target class. The epoch is set to 300, the batch size is 181, the initial learning rate is 0.01, and the learning rate is halved for every 50 epochs.
Step 2	Forward propagation. Calculate the loss function of the data during iteration i, $L^i = L^i_{LMCL} + \lambda L^i_{center}$
Step 3	Parameter update. Stochastic gradient descent method (SGD) is employed to carry out back-propagation. The gradient is calculated by chain rule. Assuming the learning rate is μ, the gradient update formula for each parameter is as follows: $\frac{\partial L^i}{\partial x_i} = \frac{\partial L^i_{LMCL}}{\partial x_i} + \lambda \frac{\partial L^i_{center}}{\partial x_i}$ (10) $\theta_c^{i+1} \leftarrow \theta_c^i - \mu \sum_{i=1}^{m} \frac{\partial L^i}{\partial x_i} \frac{\partial x_i}{\partial \theta_c^i}$ (11) $W^{i+1} \leftarrow W^i - \mu \sum_{i=1}^{m} \frac{\partial L^i_{LMCL}}{\partial x_i} \frac{\partial x_i}{\partial W^i}$ (12) $c_j^{i+1} \leftarrow c_j^i - \mu \Delta c_j^i$ (13)
Step 4	Repeat Step 2, 3 until the loss function converges, end the training process, and use the testing data to verify the validity of the model.

C. SIMULATION RESULTS AND ANALYSIS

In this section, we present the experimental results from two aspects. The first part mainly analyzes the effectiveness of the simulation by the recognition accuracy. The second part mainly shows the effectiveness of the proposed method by visualizing the features extracted by the models.

1) RECOGNITION ACCURACY

The recognition accuracy rate is the ratio of correctly classified samples to the total testing samples. It can be used as an indicator to evaluate the validity of the model. The recognition performance of proposed model is illustrated from the following two aspects in this section.

a: THE RECOGNITION ACCURACY OF PROPOSED MODEL AND FOUR COMPARISON MODELS UNDER DIFFERENT SNR CONDITIONS

In this section, we compare the recognition performance of the proposed model with two shallow models and four deep models, namely K-SVD, LSVM, CNN, SDAE, SCAE and SAE. The SNR is defined as the ratio of the power between the signal and the noise. The expression is as follows:

$$SNR = 10\log(\frac{P_{signal}}{P_{noise}}) \tag{14}$$

where, P_{signal} and P_{noise} represent the average power of the signal and noise respectively, and the unit of SNR is dB. We choose 3 different SNR which are 5, 10 and 15 respectively. The simulation results are shown in Table 6.

TABLE 6. Recognition accuracy of five models under different SNR conditions.

MODEL	ACCURACY(%)		
	SNR=5	SNR=10	SNR=15
K-SVD	81.58	86.19	88.72
LSVM	72.11	78.45	84.82
CNN	93.01	94.36	96.22
SDAE	91.58	94.39	96.05
SCAE	91.27	94.47	96.12
SAE	90.94	93.72	95.36
RIN	**94.65**	**96.98**	**98.32**

It can be seen from Table 6 that the whole performance of the deep models are much better than the shallow ones. Among the deep models, both the proposed model and CNN have better recognition performance than the methods based

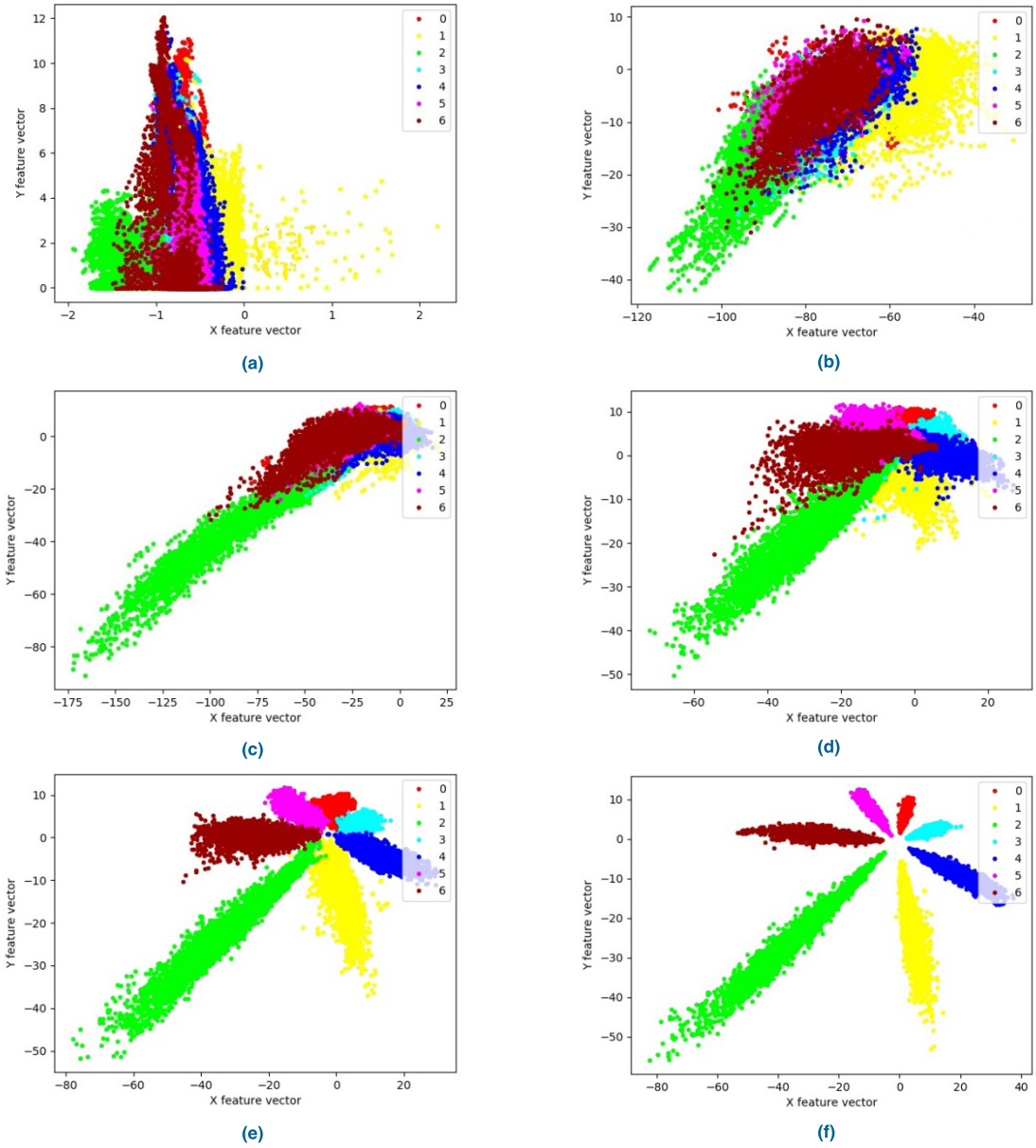


FIGURE 9. Visualization of the feature extracted by the proposed model during training (a)epoch = 1 (b) epoch = 50 (c)epoch = 100(d)epoch = 150(e) epoch = 200 (f) epoch = 250.

on the autoencoders under different SNR conditions. The models based on convolutional kernels have better robustness. The proposed model is even better than CNN.

b: THE ABILITY OF PROPOSED MODEL AND 4 COMPARISON MODELS TO OVERCOME THE TARGET ASPECT SENSITIVITY PROBLEM OF HRRP

In this section, the recognition accuracy of different models is simulated using data set B under the condition SNR = 10. The simulation results are shown in Table 7.

As shown in Table 7, the recognition performance of the proposed model and the convolutional neural network is better than the other three.

TABLE 7. Recognition accuracy of five models.

MODEL	RIN	CNN	SDAE	SCAE	SAE
ACCURACY (%)	**94.83**	92.51	91.39	92.85	90.37

The reasons why recognition performance of the proposed model is better than models based on autoencoder can be summarized as following:

First, the parameters of the proposed model are far less than models based on autoencoder which make the model less prone to overfit and have a better generalization performance.

Second, the loss function of the proposed model takes both the inter-class distance and intra-class distance into consideration which make the features more separable.

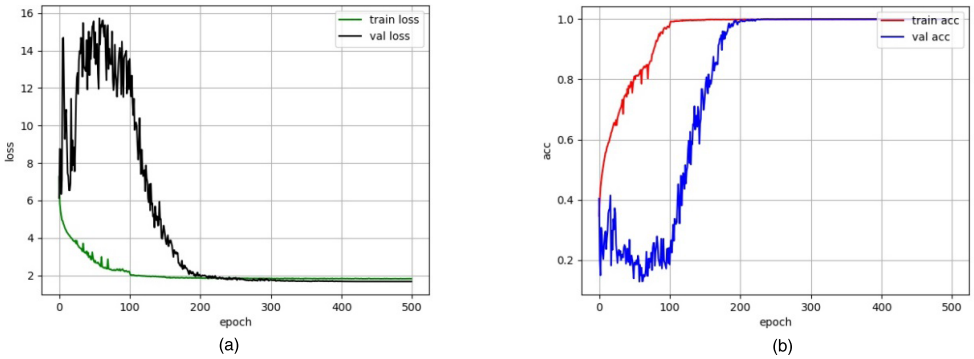


FIGURE 10. Loss curve and recognition accuracy curve during training (a) loss curve of the training and testing set (b) recognition accuracy curve of training and testing set.

Thirdly, the pooling layer reduces the dimension of each feature vector and remains the largest element within the window which makes the proposed model more invariant to the translation and target aspect sensitivity of HRRP data.

Lastly, the feature extracted by convolutional kernel preserves the topology structure of the scattering centers and the ReLU function can remove information of the weak points and make the features more sparsity.

2) FEATURE VISUALIZATION

All the simulations in this section is done by using data set A under condition SNR $= 15$.

a: FEATURE VISUALIZATION DURING TRAINING

In this section, we visualize the features extracted by the proposed model during training, as shown in Fig.9. The initial learning rate is very important. The loss function will fluctuate greatly and not converge with a high learning rate, otherwise, will converge too slow. After experimenting, a stable and relatively quick convergence of the loss function can be obtained by setting the initial learning rate to 0.01 and halving the learning rate for every 50 epochs. In order to better understand the feature evolution process during training, the loss function curve and the recognition accuracy curve during the training is shown in Fig.10.

As can be seen from Fig.9 (a), the initial features of each class are inseparable, and the initial recognition accuracy is about 0.2. Fig.9 (b) to (f) show the evolution of features extracted by the proposed model during training. As the number of epochs increases, the features of the various samples are gradually dissipated and aggregated toward various feature centers. As shown in Fig.9 (a), (b), (c), and the corresponding part of the loss curve and the recognition accuracy curve in Fig.10, in the first 100 epochs, the features are in a mess and inseparable. The loss curve and the accuracy curve of the training set descend and increase gradually with a small fluctuation, respectively. While, both the loss curve and accuracy curve of the testing set have a drastic fluctuation instead of a convergence. This indicates that the model is in the searching process for the global optimal value, and at this time, the model is at the over-fitting state and has a poor generalization performance. As shown in Fig.9 (d), (e) and Fig.10, in the 100 to 200 epochs, the features extracted from the samples of each class are gradually separated. The loss curve and accuracy curve of the validation data converge to the corresponding value of the training data with a small fluctuation. This indicates that during this period the model is approaching to the global optimal value and non-overfitting state. As shown in Fig.9 (e), when the epoch number is 200, the features of each class are separable, but with the fuzzy boundaries. Reduce the learning rate, continue training. As shown in Fig.9 (f), when the epoch number is 250, the intra-class and inter-class angular distance of each feature are small and large respectively, and the feature separability is excellent. At this time, the loss curve and the accuracy curve both converge to a certain and steady value.

b: FEATURE VISUALIZATION OF THE CNN AND PROPOSED METHOD UNDER DIFFERENT CLASS NUMBER

Since the recognition accuracy of the normal convolutional neural network (CNN) is better than that of the other methods in the full-angle domain. Therefore, we select the CNN to be the comparison model and visualize the features of CNN and the proposed model. The features extracted by the two models are visualized under different class number conditions and is shown in Fig.11.

It can be seen from the Fig.11 (a) that when the class number is 3, the features extracted by the CNN are separable, the inter-class boundaries are more distinct and have no overlapping.

As shown in Fig.11 (c), (e) that as the class number increases, the features of different classes extracted by CNN are more likely to overlap and have a poor separability.

It can be seen from Fig.11 (b), (d), (f) that both inter-class separability and intra-class aggregation of the features extracted by the proposed model is better than that of CNN. In particular, the advantages of the proposed model become more obvious with an increase of the class number. Look at Fig.11 (b) in detail, when the class number is small, the features of different classes appear to be misrecognized. The main reason is that the proposed model with relatively large

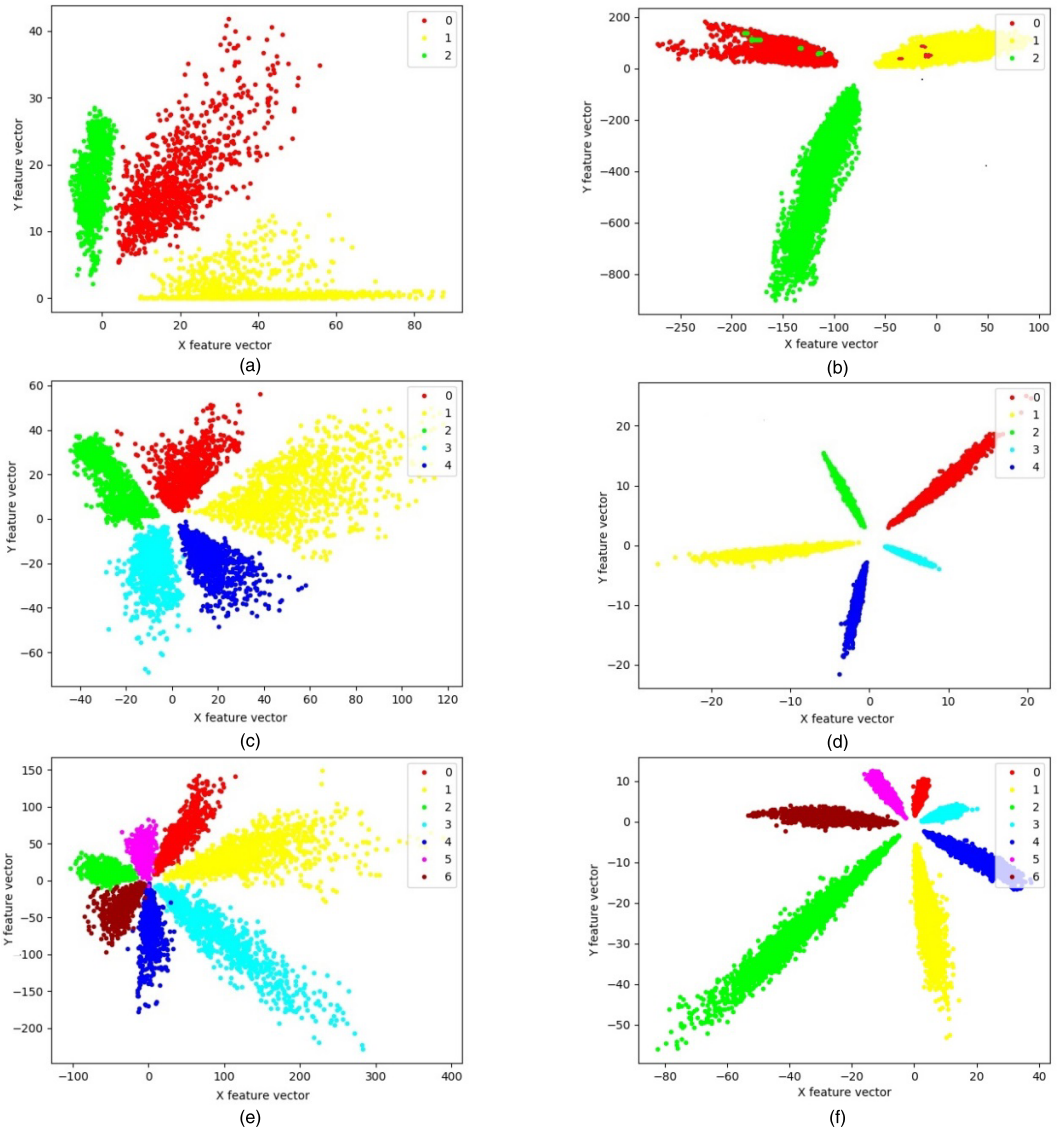


FIGURE 11. Feature visualization of CNN and proposed method under different class number conditions (a) feature visualization of CNN when class number is 3 (b) feature visualization of proposed method when class number is 3 (c) feature visualization of CNN when class number is 5 (d) feature visualization of proposed method when class number is 5 (e) feature visualization of CNN when class number is 7 (f) feature visualization of proposed method when class number is 7.

parameters is prone to over-fit because of the insufficient training data.

Because of the computation complexity of proposed model, the training time is relatively long compared with CNN. The proposed model takes 5404.2s for 300 epochs, whereas the common CNN takes 400.4s. Therefore, when the number of training samples is small, the CNN is the better choice for recognition task.

As the number of target class increases, the amount of training samples increases, and the target features become more complex. The convolutional kernel with a single scale in the same layer of the CNN has a relatively weak ability to represent complex features. Therefore, when the number of target class increases, the separability of the features extracted by the CNN decreases. However, the residual-inception layer and inception pooling layer contain convolution kernels with different scales, which makes the model have the ability to represent complex features. In summary, when the number of target class is large, the proposed model becomes a better choice.

V. CONCLUSIONS

A radar target recognition method based on deep one-dimensional residual-inception network is proposed in this paper. The weight sharing characteristic of the convolutional kernel is adopted to reduce the model parameters and improve the expression ability of the model, which greatly improves the training efficiency. In this paper, the residual inception layer and inception-pooling layer are used to extract different fine-grained features of the targets. A new loss function – cosine center loss function is proposed to constrain the features and make features more separable. From the simulation

results, the recognition accuracy of the proposed method is really good. The features extracted by the model can take into account both the inter-class separability and intra-class aggregation. The proposed model has a good generalization performance and robustness as well.

REFERENCES

[1] L. Du, P. Wang, H. Liu, M. Pan, F. Chen, and Z. Bao, "Bayesian spatiotemporal multitask learning for radar HRRP target recognition," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3182–3196, Jul. 2011.

[2] Y. Jiang, Y. Li, J. Cai, Y. Wang, and J. Xu, "Robust automatic target recognition via HRRP sequence based on scatterer matching," *Sensors*, vol. 18, no. 2, p. 593, 2018.

[3] L. Du, H. Liu, Z. Bao, and J. Zhang, "A two-distribution compounded statistical model for radar HRRP target recognition," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2226–2238, Jun. 2006.

[4] Y. Wang, L. Zhang, S. Wang, T. Zhao, Y. Wang, and Y. Li, "Radar HRRP target recognition using scattering centers fuzzy matching," in *Proc. CIE Int. Conf. Radar*, 2016, pp. 1–5.

[5] L. Du, H. He, L. Zhao, and P. Wang, "Noise robust radar HRRP target recognition based on scatterer matching algorithm," *IEEE Sensors J.*, vol. 16, no. 6, pp. 1743–1753, Mar. 2016.

[6] B. Pei and Z. Bao, "Multi-aspect radar target recognition method based on scattering centers and HMMs classifiers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 1067–1074, Jul. 2005.

[7] Y. Jiang, Y. Han, and W. Sheng, "Target recognition of radar HRRP using manifold learning with feature weighting," in *Proc. IEEE Int. Workshop Electromagn., Appl. Student Innov. Competition*, May 2016, pp. 1–3.

[8] D. Zhou, "Radar target HRRP recognition based on reconstructive and discriminative dictionary learning," *Signal Process.*, vol. 126, pp. 52–64, Sep. 2016.

[9] B. Feng, L. Du, C. Shao, P. Wang, and H. Liu, "Radar HRRP target recognition based on robust dictionary learning with small training data size," in *Proc. IEEE Radar Conf.*, Apr./May 2013, pp. 1–4.

[10] B. Feng, L. Du, H.-W. Liu, and F. Li, "Radar HRRP target recognition based on K-SVD algorithm," in *Proc. IEEE CIE Int. Conf. Radar*, Oct. 2012, pp. 642–645.

[11] K. Chen, Y. Li, and Y. Ma, "An efficient K-SVD algorithm of dictionary learning for HRRP targets recognition kun CHEN 1," in *Proc. ICMIA*, 2016, pp. 513–518.

[12] P. López-Rodríguez, D. Escot-Bocanegra, R. Fernández-Recio, and I. Bravo, "Non-cooperative target recognition by means of singular value decomposition applied to radar high resolution range profiles.," *Sensors*, vol. 15, no. 1, pp. 422–439, 2014.

[13] Y. Guo, H. Xiao, H. Fan, and Y. Zhu, "Multiclass multiple kernel learning for HRRP-based radar target recognition," *Proc. SPIE*, vol. 10443, p. 1044306, Jun. 2017.

[14] W. Xiong, G. Zhang, S. Liu, and J. Yin, "Multiscale kernel sparse coding-based classifier for HRRP radar target recognition," *IET Radar, Sonar Navigat.*, vol. 10, no. 9, pp. 1594–1602, 2017.

[15] Y. Guo, H. Xiao, Y. Kan, and Q. Fu, "Learning using privileged information for HRRP-based radar target recognition," *IET Signal Process.*, vol. 12, no. 2, pp. 188–197, 2018.

[16] M. Pan, J. Jiang, Q. Kong, J. Shi, Q. Sheng, and T. Zhou, "Radar HRRP target recognition based on t-SNE segmentation and discriminant deep belief network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1609–1613, Sep. 2017.

[17] P. Mian, J. Jie, L. Zhu, C. Jing, and Z. Tao, "Radar HRRP recognition based on discriminant deep autoencoders with small training data size," *Electron. Lett.*, vol. 52, no. 20, pp. 1725–1727, 2016.

[18] B. Feng, B. Chen, and H. Liu, "Radar HRRP target recognition with deep networks," *Pattern Recognit.*, vol. 61, pp. 379–393, Jan. 2017.

[19] H. Yan, Z. Zhang, G. Xiong, and W. Yu, "Radar HRRP recognition based on sparse denoising autoencoder and multi-layer perceptron deep model," in *Proc. 4th Int. Conf. Ubiquitous Positioning, Indoor Navigat. Location Based Services*, 2016, pp. 283–288.

[20] Y. Zhai, B. Chen, H. Zhang, and Z. Wang, "Robust variational autoencoder for radar HRRP target recognition," in *Proc. Int. Conf. Intell. Sci. Big Data Eng.*, 2017, pp. 356–367.

[21] F. Zhao, Y. Liu, K. Huo, S. Zhang, and Z. Zhang, "Radar HRRP target recognition based on stacked autoencoder and extreme learning machine," *Sensors*, vol. 18, no. 1, p. 173, 2018.

[22] J. Lundén and V. Koivunen, "Deep learning for HRRP-based target recognition in multistatic radar systems," in *Proc. IEEE Radar Conf.*, May 2016, pp. 1–6.

[23] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[25] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 770–778.

[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. (2016). "Inception-v4, inception-ResNet and the impact of residual connections on learning." [Online]. Available: https://arxiv.org/abs/1602.07261

[30] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 1800–1807.

[31] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[32] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

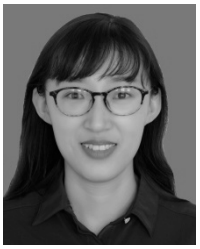
[33] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[34] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 507–516.

[35] Y. Liu, H. Li, and X. Wang. (2017). "Learning deep features via congenerous cosine loss for person recognition." [Online]. Available: https://arxiv.org/abs/1702.06890

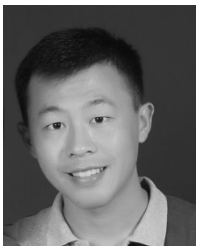
[36] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification,"in *Proc. ACM Conf. Multimedia*, 2017, pp. 1041–1049.

[37] Y. Liu, H. Li, and X. Wang. (2017). "Rethinking feature discrimination and polymerization for large-scale recognition." [Online]. Available: https://arxiv.org/abs/1710.00870



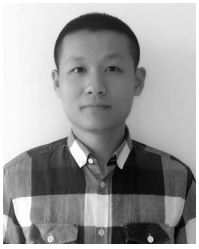
CHEN GUO received the M.S. degree from the National University of Defense Technology, China, in 2015. She is currently pursuing the Ph.D. degree with Naval Aviation University, China. Her research interests focus on radar signal processing, machine learning, and deep learning.

YOU HE received the Ph.D. degree from Tsinghua University, in 1997. He was a Visiting Scholar with the Brunswick University of Technology, Germany. He is currently a Full Professor and also a Doctor Tutor with Naval Aviation University. His research interests include the general area of information fusion theory and technology, equipment simulation, and big data technology and application. He is an IET Fellow.



HAIPENG WANG received the Ph.D. degree from Naval Aviation University, in 2012, where he is currently an Associate Professor. His research interests include the general area of intelligent perception and fusion, and big data technology and application. He also serves as a Reviewer for several distinguished journals, including IET RSN and IEEE AES.

TAO JIAN received the Ph.D. degree from Naval Aviation University, in 2010, where he is currently an Associate Professor. His research interests include the general area of radar signal processing and target recognition.



SHUN SUN is currently pursuing the Ph.D. degree with Naval Aviation University, China, in 2017. His research interests focus on information fusion theory and technology, passive localization, coordinative control, and deep reinforcement learning.