

RESEARCH

Open Access



Convolutional neural networks for radar HRRP target recognition and rejection

Jinwei Wan^{1,2}, Bo Chen^{1,2*}, Bin Xu^{1,2}, Hongwei Liu^{1,2} and Lin Jin^{1,2}

Abstract

Robust and efficient feature extraction is critical for high-resolution range profile (HRRP)-based radar automatic target recognition (RATR). In order to explore the correlation between range cells and extract the structured discriminative features in HRRP, in this paper, we take advantage of the attractive properties of convolutional neural networks (CNNs) to address HRRP RATR and rejection problem. Compared with the time domain representations, the spectrogram of HRRP records the amplitude feature and characterizes the phase information among the range cells. Thus, besides using one-dimensional CNN to handle HRRP in time domain, we also devise a two-dimensional CNN model for the spectrogram feature. Furthermore, by adding a deconvolutional decoder, we integrate the target recognition with outlier rejection task together. Experimental results on measured HRRP data show that our CNN model outperforms many state-of-the-art methods for both recognition and rejection tasks.

Keywords: Radar automatic target recognition (RATR), High-resolution range profile (HRRP), Spectrogram feature, Outlier rejection, Convolutional neural networks (CNNs)

1 Introduction

Radar target detection and radar automatic target recognition (RATR) are two active research fields of modern radar technology. In a typical modern radar system, targets are first located at the detection stage, which aims to estimate the target position/Doppler with high quality [1, 2]. The high-resolution radar (HRR) is then activated by specific targets for further identification or classification. The echo of HRR, high-resolution range profile (HRRP), is the amplitude of the coherent summations of the complex time returns from target scatterers in each range cell, which represents the projection of the complex returned echoes from the target scattering centers onto the radar line-of-sight (LOS), as shown in Fig. 1. Since HRRP contains abundant target structure signatures, such as target size and scatterer distribution, HRRP-based target recognition has received intensive attention from the RATR community [3–25].

Feature learning is critical for RATR. Many researchers [5–19] explore various feature (representation) learning

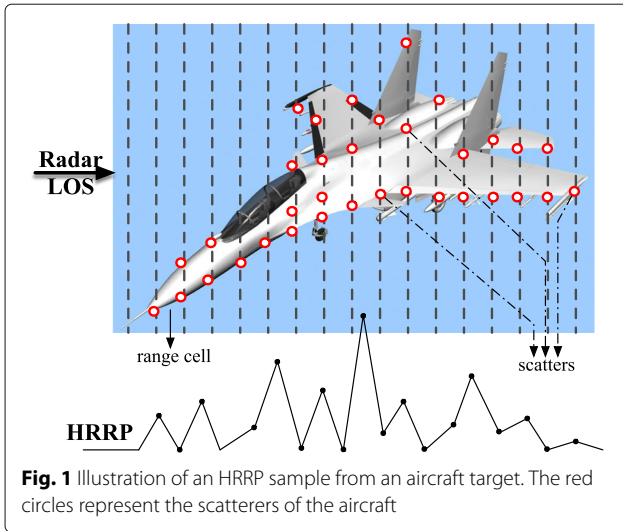
methods for HRRP-based RATR. In [5, 6], RELAX-based algorithms are employed to extract the location information of predominant scatterers from HRRP data as features for recognition tasks. In [7–9], the researchers investigate the recognition methods based on spectra features, which perform the recognition task in the learned feature subspace via specific physical meaning. Generally, such sophisticated features perform well in practice but they heavily rely on personal experiences and prior knowledge.

Besides those hand-crafted feature methods, data-driven RATR approaches have attracted increasing attention in past years due to their ability to learn useful features from the dataset automatically. In [10], the principal components analysis (PCA)-based feature subspace is constructed to minimize reconstruction error for RATR. In [11–15], the researchers employ factor analysis (FA) model to project and recognize HRRPs in a low-dimensional latent feature space. Considering the sparsity within the HRRP signals, Feng et al. [16] and Zhou [17] apply sparse constraint on the feature vectors and solve the problems via l_0 -minimization. However, all those methods build linear and shallow architectures that limit their capability to represent the complicated HRRP data.

*Correspondence: bchen@mail.xidian.edu.cn

¹National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, Shaanxi, China

²Collaborative Innovation Center of Information Sensing and Understanding, Xi'an 710071, Shaanxi, China



Recently, non-linear, deep networks have been successfully employed in various real-world tasks thanks to their powerful expressive ability [26–34]. In fact, several deep learning models have been developed for HRRP-based RATR. Yan et al. [18] employ stacked denoising autoencoder (SDAE [27]) to learn the robust representation of the original HRRP. Feng et al. [19] employ the average profile as the correction terms and stack a series of Corrective Autoencoders to extract features from HRRP. However, these two models are based on fully connected nets and may not capture the structural information among the range cells of HRRP layer by layer, since HRRP reflects the distribution of scatterers in target along the range dimension. Differently, convolutional neural networks (CNNs [29]) explicitly exploit structural locality in the feature space and can extract more descriptive features that exist in signals. Furthermore, the convolutional layer alternates with the pooling layer to learn the features, which leads to the local invariance of translation, scaling, and shift. They have been successfully applied to kinds of recognition tasks and exhibited better performance than fully connected networks [29–32, 35, 36].

In this paper, we first develop a one-dimensional CNN recognition procedure for the time domain HRRP, which represents the amplitude of projection of the target scattering centers onto the radar LOS and is widely used in RATR. In the 1-D CNN, the convolution operation only takes place at range dimension. Nevertheless, the time domain only characterizes the amplitude of the target signal which provides limited information for feature learning. By contrast, the spectrogram feature is a two-dimensional (time and frequency) representation of a HRRP signal, which embeds the frequency domain property of the target and reflects more phase information [20]. Hence, targeting the properties of spectrogram, a

novel CNN model is proposed for HRRP ATR, which, different from the conventional 2-D CNN for images, considers the spectral vector at each discrete time (range cell) as a whole and realizes the convolution operation along time dimension.

Since most of the targets are uncooperative or even hostile in real-world RATR task, it is not practical to acquire complete training database. Moreover, it is unreasonable to recognize a target that does not belong to any class of the existing template database. Therefore, we plug a decoder into the model to realize the outlier rejection. Specifically, the proposed approach uses a CNN as encoder and a deconvolutional neural network as decoder. By measuring the reconstruction error between the input and the output of the decoder, the model can identify whether a sample is an outlier or not.

The remainder of this article is organized as follows. Brief descriptions of time domain and spectrogram feature of HRRP are introduced in Section 2. In the following section, we will present our CNN model in detail. The experimental results of our model with time domain and spectrogram feature on measured HRRP data are shown in Section 4. Finally, conclusions are addressed in Section 5.

2 Preliminaries

In this section, we briefly review the concepts of HRRP and CNN, following which the corresponding descriptions of time domain and the spectrogram feature of HRRP are provided.

2.1 Time domain HRRP

Generally, high-resolution radar (HRR) operates in microwave frequency band, and for the wide bandwidth of the signal, the wavelength of radar is much smaller than the target size. Hence, for complex targets such as an aircraft, HRR can effectively divide the object into many range “cells.” According to the literatures [10, 14], if the radar-transmitted signal is $s(t)e^{jw_c t}$ ($s(\cdot)$ is the complex envelope and w_c is the carrier angular frequency) and the returned complex HRRP is denoted as a discrete complex vector, the n -th complex returned echo from the l -th range cell ($l = 1, 2, \dots, L$) in baseband can be approximated as:

$$x_l(t, n) \approx x_l(n) = e^{j\theta(n)} \sum_{i=1}^{V_l} \sigma_{li} e^{j\phi_{li}(n)} \quad (1)$$

where $\theta(n) = -\frac{4\pi}{\lambda}R(n)$ stands for the initial phase of the n -th returned echo related to the target distance and the radar wavelength and V_l denotes the number of target scatterers in the l -th range cell, σ_{li} the strength of the i -th scatterer in the l -th range cell, and $\phi_{li}(n)$ the remained

echo phase of the i -th scatterer in the l -th range cell of the n -th returned echo.

Then, the n -th complex HRRP can be written as $\mathbf{x}(n) = e^{j\theta(n)} [\tilde{x}_1(n), \tilde{x}_2(n), \dots, \tilde{x}_L(n)]^T$, where $\tilde{x}_l(n) = \sum_{i=1}^{V_l} \sigma_{li} e^{j\phi_{li}(n)}$ with $l = 1, 2, \dots, L$, and T denotes the transpose operation. The n -th time domain HRRP $\mathbf{x}_T(n)$ is obtained by taking absolute value of $\mathbf{x}(n)$, which can be represented as:

$$\begin{aligned} \mathbf{x}_T(n) &= [|\tilde{x}_1(n)|, |\tilde{x}_2(n)|, \dots, |\tilde{x}_L(n)|]^T \\ &= \left[\left| \sum_{i=1}^{V_1} \sigma_{1i} e^{j\phi_{1i}(n)} \right|, \left| \sum_{i=1}^{V_2} \sigma_{2i} e^{j\phi_{2i}(n)} \right|, \dots, \right. \\ &\quad \left. \left| \sum_{i=1}^{V_L} \sigma_{Li} e^{j\phi_{Li}(n)} \right| \right]^T \end{aligned} \quad (2)$$

where $|\cdot|$ means taking absolute value. The time domain HRRP \mathbf{x}_T represents the reflected signal intensity versus range along the radar LOS. Therefore, it is worth of note that we only consider the corresponding amplitude profile of the time domain HRRP and spectrogram of HRRP in this paper, which do not vary with the initial phase [10].

2.2 Spectrogram feature of HRRP

Spectrogram analysis is a common signal processing procedure adopted in spectral analysis and other fields, which can be readily created by calculating the short-time Fourier transform (STFT) of the time signal. The discrete-time STFT transform can be represented as:

$$\text{STFT}(m, \omega) = \sum_{n=-\infty}^{\infty} x(n) w(n-m) e^{-j\omega n} \quad (3)$$

where $x(n)$ is the signal to be transformed and $w(\cdot)$ is the window function. The magnitude squared of the STFT yields the spectrogram of $x(n)$:

$$\text{spectrogram}\{x(n)\}(m, \omega) = |\text{STFT}(m, \omega)|^2 \quad (4)$$

In this paper, the spectrogram feature of n -th HRRP, $\mathbf{x}_S(n)$, is obtained by applying STFT and square operation on the n -th complex HRRP $\mathbf{x}(n)$. Specifically, we first sequentially break up the complex HRRP $\mathbf{x}(n)$ into fragments of length M . To reduce artifacts at the boundary and increase the temporal dependence between fragments, each fragment overlaps with its neighbors of length $M-1$. For notational convenience, we replace the notation $\mathbf{x}(n)$ with \mathbf{x} , and a fragment starting from the k -th range cell can be written as $\mathbf{x}^M = [\tilde{x}_k, \tilde{x}_{k+1}, \dots, \tilde{x}_{k+M-1}]$, where $\tilde{x}_l = \sum_{i=1}^{V_l} \sigma_{li} e^{j\phi_{li}}$ with $l = k, k+1, \dots, k+M-1$. Then, performing the discrete Fourier transformation (DFT) on \mathbf{x}^M yields its frequency domain expression, $\mathbf{x}_{\mathcal{F}}^M$, which can be expressed as:

$$\begin{aligned} \mathbf{x}_{\mathcal{F}}^M(m) &= \sum_{n=0}^{M-1} \tilde{x}_{n+k} W_M^{mn} \\ &= \sum_{n=0}^{M-1} \left[\sum_{i=1}^{V_{k+n}} \sigma_{(k+n)i} e^{j\phi_{(k+n)i}} \right] e^{-j\frac{2\pi}{M} mn}, \end{aligned} \quad (5)$$

$m = 0, 1, \dots, M-1$

Taking the absolute value of $\mathbf{x}_{\mathcal{F}}^M$ s and concatenating them in sequence provides the spectrogram feature of n -th HRRP, \mathbf{x}_S . As a result, a complex vector $\mathbf{x} \in \mathbb{R}^{1 \times L}$ is transformed into a matrix $\mathbf{x}_S \in \mathbb{R}^{H \times L}$, where the axes of \mathbf{x}_S are frequency and time, respectively. Figure 2 presents the time domain HRRP samples and their corresponding spectrogram features from three airplane targets, i.e., An-26, Cessna Citation S/II, and Yark-42, where the spectrogram features are obtained by employing a length of 16, windowed with a Hamming window and 32 points FFT to STFT.

The spectrogram feature of HRRP has some distinct advantages over time domain HRRP. First, by taking the absolute value (2), the time domain HRRP loses its phase information. On the contrary, the spectrogram exploits the phase information to characterize the frequency domain correlation among range cells since the DFT operation will accumulate the same frequency component in the fragment (5). In other words, by making use of the phase information in the complex HRRP signal, the spectrogram feature of HRRP provides more (frequency domain) information than HRRP in time domain. Second, the scatterers living in several continuous range cells are relatively more robust to the target-aspect change than those in a single range cell.

3 HRRP target recognition and outlier rejection based on CNN

In order to employ CNN model to HRRP recognition task, we adopt a one-dimensional CNN model to classify the time domain HRRPs directly, while for spectrogram feature, the designed CNN model realizes the convolution operation going along the time dimension. Furthermore, by adding a deconvolutional network to the CNN model, we integrate recognition with rejection together. Figure 3 illustrates the structure of our model.

3.1 CNN for time domain

A deep convolutional neural network [29] consists of multiple layers based on convolution operation and can be viewed as a transformation from the input map to the output map. Suppose $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ is a multi-channel input, where each dimension of \mathbf{X} represents

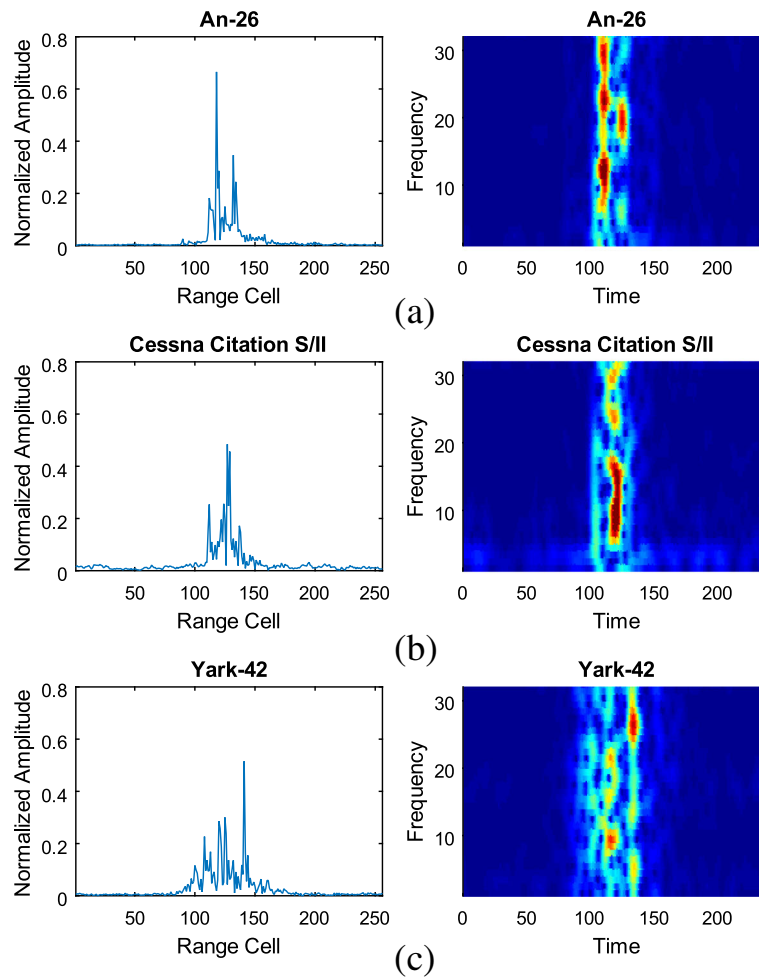


Fig. 2 Example representations of HRRP in time domain (left) and corresponding spectrogram features (right) of HRRP. **a** An-26. **b** Cessna Citation S/II. **c** Yark-42

the height, width, and the number of channels, respectively. Assume the CNN consists of L convolutional layers, and layer $l \in \{1, \dots, L\}$ K_l filters. For the k -th filter at layer 1, a convolution operation with stride length $r^{(1)}$ applies filter $\mathbf{W}^{(k,1)} \in \mathbb{R}^{h^{(1)} \times w^{(1)} \times C}$ to \mathbf{X} .

This yields feature map, $\mathbf{y}^{(k,1)} = f(\mathbf{X} * \mathbf{W}^{(k,1)} + \mathbf{b}^{(k,1)})$, where $*$ denotes the convolutional operator, $\mathbf{b}^{(k,1)}$ is the bias for the k -th feature map, and $f(\cdot)$ a nonlinear activation function. More specifically, the value of a unit at position (i', j') , denoted as $y_{i'j'}^{(k,1)}$, is given by:

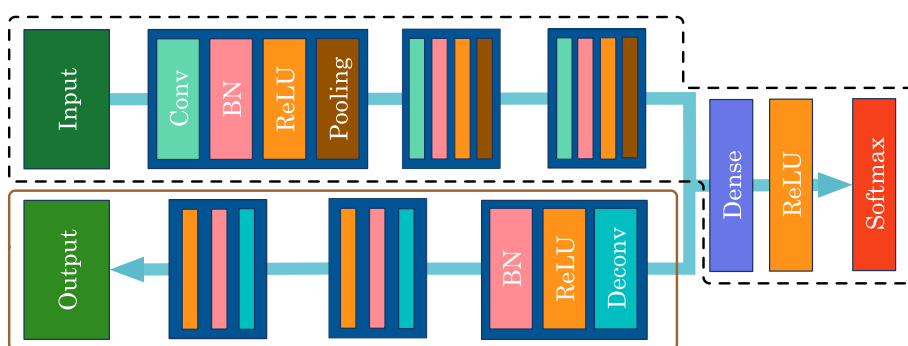


Fig. 3 Model structure. We take three convolutional (deconvolutional) layer as an example. The black dotted functions as the encoder part, and the brown solid block the decoder part

$$y_{i'j'}^{(k,1)} = \text{ReLU} \left(\sum_{i=1}^h \sum_{j=1}^w \sum_{c=1}^C X_{i+i',j+j',c} \odot W_{ijc}^{(k,1)} + b^{(k,1)} \right) \quad (6)$$

where \odot means element-wise product or Hadamard product and ReLU [37] is a nonlinear activate function, $\text{ReLU}(z) = \max(0, z)$. Figure 4 illustrates a convolutional layer in CNN. The computational complexity in the convolution layer 1 is in the order of $O(H \cdot W \cdot C \cdot h^{(1)} \cdot w^{(1)} \cdot K_1)$, similarly for the following convolution layers.

To increase the content covered by a convolutional kernel and the sparsity of the hidden units, a convolutional layer is usually followed by a subsampling layer. The resolution of the feature map is reduced by pooling over the local neighborhood on the feature map. After the convolution and pooling, the k -th feature map $\mathbf{h}^{(k,1)} \in \mathbb{R}^{H^{(h1)} \times W^{(h1)} \times 1}$ is yielded, where $H^{(h1)}$ and $W^{(h1)}$ are the height and width of the feature map, respectively. After that, we have the feature maps of layer 1, $\mathbf{H}^{(1)} \in \mathbb{R}^{H^{(h1)} \times W^{(h1)} \times K_1}$, and the computational complexity in this layer is in the order of $O(H \cdot W \cdot K_1)$, which is much smaller than that in the convolution layer and similar for the following subsampling layers.

With this convolutional-subsampling operation repeated in sequence for L layers, the last feature map $\mathbf{H}^{(L)}$ is obtained, which is then fed into some fully connected nets, to produce the final feature \mathbf{z} . In the end, the feature \mathbf{z} goes into a softmax classifier for the recognition task. The softmax function $p_c(\mathbf{z}) = \frac{\exp(\theta^{(c)\top} \mathbf{z})}{\sum_{j=1}^C \exp(\theta^{(j)\top} \mathbf{z})}$ is used to perform multi-class logistic regression, where C represents the number of categories, $p_c(\mathbf{z})$ the probability of \mathbf{z} belonging to the c -th category, and θ the parameter of the softmax classifier. Given the ground-truth class label t of input \mathbf{X} , we formulate the recognition loss function E_{recogn} as:

$$E_{\text{recogn}} = -\frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C t^{(n)} \ln p_c^{(n)}(\mathbf{z}) \quad (7)$$

where N is the sample number of each mini-batch.

The time domain HRRP, $\mathbf{x}_T \in \mathbb{R}^{1 \times W \times 1}$, is a real-valued vector (W is the length of range dimension); thus, a 1-D convolution CNN model, called TCNN, can be employed, where the convolution operation only takes place at the range dimension. For the first layer in TCNN, as shown in Fig. 5a, convolution operations with stride length $r^{(1)}$ apply K_1 filters, $\mathbf{W}_T^{(1)} \in \mathbb{R}^{1 \times w^{(1)} \times 1}$, to \mathbf{x}_T , resulting in feature maps of layer 1, $\mathbf{H}_T^{(1)} \in \mathbb{R}^{1 \times W^{(h1)} \times K_1}$. For the following layers, convolution-subsampling operation repeatedly apply K_l filters, $\mathbf{W}_T^{(l)} \in \mathbb{R}^{1 \times w^{(l)} \times K_{l-1}}$, to $\mathbf{H}_T^{(l-1)}$, as illustrated in Fig. 5b, gaining feature maps $\mathbf{H}_T^{(l)}$. For the final convolutional layer L , the feature maps $\mathbf{H}_T^{(L)}$ are obtained and fed into fully connected nets, to produce the feature vector \mathbf{z}_T . In the end, the recognition task is achieved by feeding \mathbf{z}_T to a softmax classifier, and the loss function is the same as (7).

3.2 CNN for spectrogram domain

As discussed in Section 2.2, the spectrogram representation (Fig. 2 right column) actually has two merits over time domain (Fig. 2 left column): (1) by bringing in the phase information, the spectrogram feature contains more information than time domain HRRP. (2) It is more insensitive to the variation of target aspect than time domain HRRP. To exploit the above advantages, we devise a CNN recognition model for HRRP spectrogram, denoted SCNN. Different from the conventional 2-D CNN for images, SCNN treats the spectral vector at each discrete time (range cell) as a whole, and the convolution operation goes along time (range) dimension. Similar with the TCNN model, SCNN model consists of L convolutional layers and several fully connected layers.

Let $\mathbf{x}_S \in \mathbb{R}^{H \times W \times 1}$ denote the spectrogram feature of HRRP, where H and W represent the dimensions of frequency and time (range) domain of spectrogram,

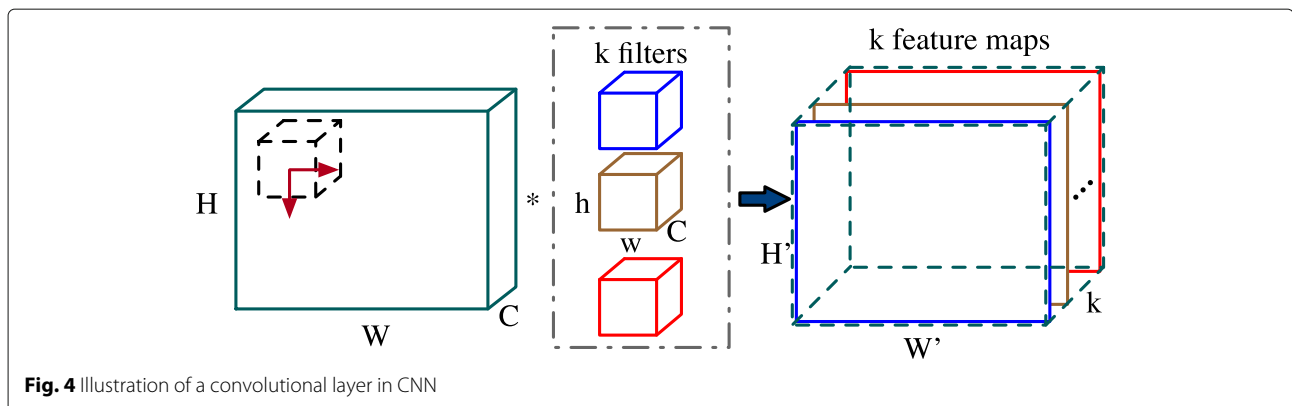
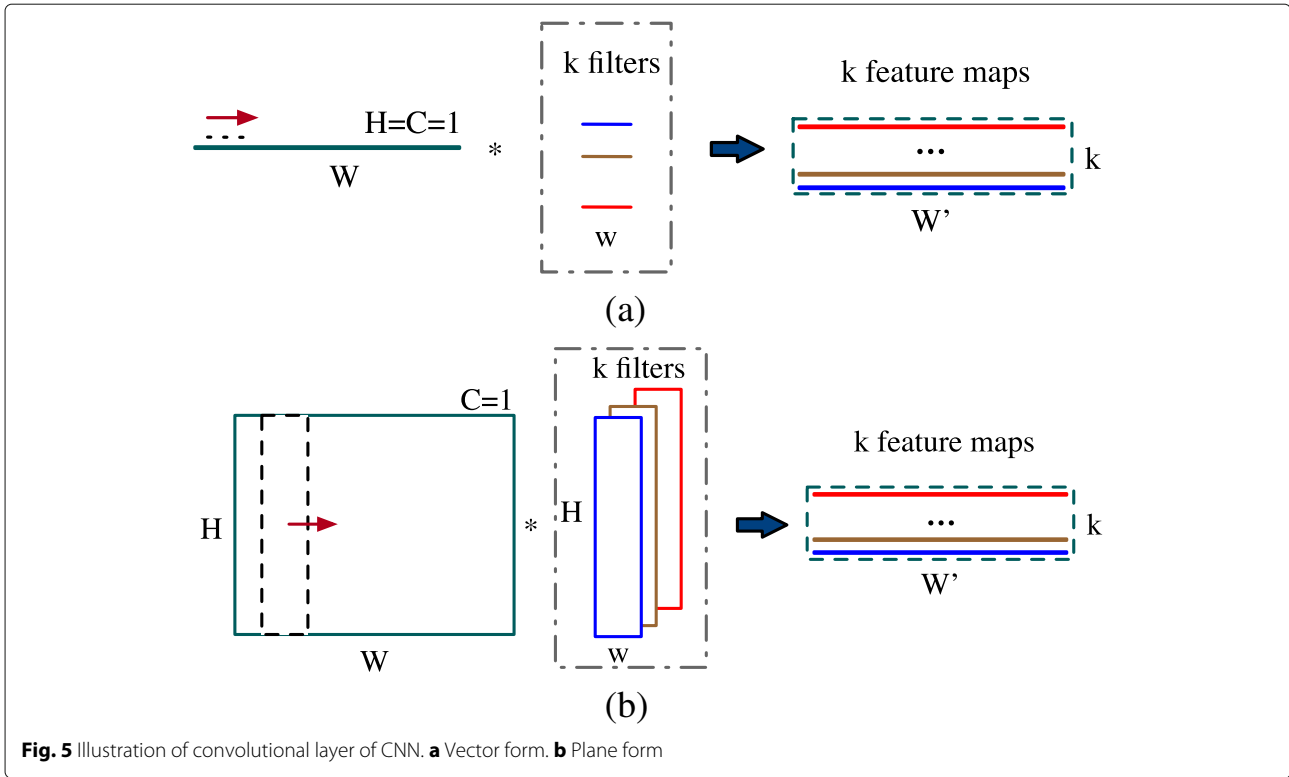


Fig. 4 Illustration of a convolutional layer in CNN



respectively. For the i -th filter at layer 1, a convolution operation with stride length $r^{(1)}$ applies filter $\mathbf{W}_S^{(i,1)} \in \mathbb{R}^{H \times w^{(1)} \times 1}$ to \mathbf{x}_S . Just as the TCNN model, feature map $\mathbf{h}_S^{(i,1)} = f(\mathbf{H}_S^{(1)} * \mathbf{W}_T^{(i,1)}) \in \mathbb{R}^{1 \times W^{(h1)} \times 1}$ is gained after a convolution-subsampling operation, where $W^{(h1)}$ is the width of feature map. We get feature maps $\mathbf{H}_S^{(1)} \in \mathbb{R}^{1 \times W^{(h1)} \times K_1}$ at layer 1. With this repeated in sequence for the following convolutional layers, we obtain the last convolution layer's feature maps, $\mathbf{H}_S^{(L)}$. After reshaping and feeding $\mathbf{H}_S^{(L)}$ to the fully connected nets, we acquire feature vector \mathbf{z}_S and a softmax function is applied on \mathbf{z}_S as the final layer.

3.3 Batch normalization

Batch normalization (BN) [33] is a technique to accelerate training and improve generalization, which is widely used in kinds of deep neural networks. For these reasons, we employ batch normalization in each convolution and fully connected layer in our model. For a layer with d -dimensional input $\mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(d)})$, the BN operation firstly normalizes each dimension with $\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}] + \epsilon}}$, where the expectation and variance are computed over the mini-batch from the training dataset, and ϵ is a constant added to the mini-batch variance for numerical stability. After normalizing, a pair of parameters $\gamma^{(k)}$, $\beta^{(k)}$ is introduced to scale and shift the normalized value:

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}. \quad (8)$$

The parameters $\gamma^{(k)}$ and $\beta^{(k)}$ are learned based on the original model parameters. By normalizing each input unit to have zero mean and unit variance, the batch normalization layer helps deal with poor initialization problems at the training stage and contributes to gradient flow in deeper models.

3.4 Combine recognition with rejection

Besides the recognition, outlier rejection is another function we have to concern in the RATR system. Thus, to integrate the outlier rejection task into our model, we introduce a decoder network, which focuses on capturing the task-general knowledge through the experience of representing the existing targets. Specifically, the input \mathbf{x} is transformed into the top feature maps, \mathbf{H} , through the convolutional encoder net, then besides feeding \mathbf{H} to the classification net, we also apply the deconvolution with stride (i.e., convolutional transpose), as the conjugate operation of convolution, to decode the feature maps \mathbf{H} back to the source. As the deconvolution operation proceeds, the spatial resolution gradually increases, where the output of the final deconvolution layer is aimed at reconstructing the input \mathbf{x} . The reconstruction error E_{recon} can be formulated as:

$$E_{\text{recon}} = \frac{1}{N} \sum_{n=1}^N \left\| \mathbf{x}^{(n)} - g_{\theta}(\mathbf{H}^{(n)}) \right\|^2, \quad (9)$$

where N is the size of each mini-batch and g_{θ} is the decoder parametrized by θ . By measuring the reconstruction error in (9), the model can identify whether a target is an outlier or not. Specifically, we expect that the in-class samples will get smaller reconstruction errors while the outlier samples' reconstruction errors are relatively larger.

Our combination model simultaneously trains a convolutional autoencoder and a supervised model to the HRRP rejection and recognition problem, which is similar to semi-supervised learning models [38, 39] except that we do not use unlabeled data at the training phase. Formally, via the combination of the recognition loss (7) and the reconstruction loss (9), the final objective can be defined as:

$$E = E_{\text{recon}} + \lambda E_{\text{recon}} \quad (10)$$

where parameter λ is a nonnegative regularization parameter used to balance the recognition loss E_{recon} and the reconstruction loss E_{recon} . A larger λ means the model cares more about the rejection performance; on the opposite, a smaller λ indicates the model pays more attention to recognition performance. Particularly, our model will degenerate into conventional CNN model if we set $\lambda = 0$. According to (10), it is worth to notice that by using this joint training strategy, the learned feature \mathbf{H} will be affected by both of data representation and classification tasks. In the following experimenting section, the detailed analysis of the influence of λ will be provided when the outlier rejection is functioned.

4 Results and discussion

In this section, measured HRRP data used in our experiments will be introduced firstly. Then, several detailed analysis and discussion about the recognition and rejection performance of our model are presented and studied. Meanwhile, the influence of some model parameters are analyzed and discussed during the experiments.

4.1 Measured data

We examine the recognition and rejection performance of our CNN model on the three-class measured data, Yark-42, Cessna Citation S/II, and An-26, among which Yark-42 is a large- and medium-sized jet aircraft, Cessna Citation S/II a small-sized jet aircraft, and An-26 a medium-sized propeller aircraft. The radar works on C-band with a bandwidth of 400 MHz, and the range resolution is about 0.375 m. The parameters of the radar and airplane targets are shown in Table 1, and the projections of target trajectories onto the ground plane are shown in Fig. 6. With large SNR, the effect of noise in the measured data on the recognition can be ignored.

Table 1 Parameters of radar and planes

Radar parameters	Center freq.	5520 MHz		
	Bandwidth	400 MHz		
Planes	Length (m)	Width (m)	Height (m)	
	Yark-42	36.38	34.88	9.83
An-26	23.80	26.20	9.83	
Cessna Citation S/II	14.40	15.90	4.57	

4.1.1 Training and test dataset selection

According to the literatures [11, 21, 22], the training and test datasets are selected following two principles: (a) The training dataset should cover almost all of the target-aspect angles. (b) The elevation angles of the training and the test dataset are different. Therefore, we select the fifth and the sixth segments of An-26, the sixth and the seventh segments of Cessna Citation S/II, and the second and the fifth segments of Yak-42 as training samples, and the rest of the segments are taken as test samples. More concretely, there are totally 140,000 training samples and 5,200 test samples involved in our experiments. Furthermore, to measure the rejection performance of our model, 18,000 HRRP samples generated by the electromagnetic simulator software, XPATCH, are used as an outlier target.

4.1.2 Preprocessing

As discussed in literature [23], it is a prerequisite for radar target recognition to deal with the target-aspect, time-shift, and amplitude-scale sensitivity. Similar to the previous study [11, 23], HRRP training samples should be aligned by the time-shift compensation techniques in ISAR imaging [40] to avoid the influence of time-shift sensitivity. Each HRRP sample is normalized by L2 normalization algorithm to avoid the amplitude-scale sensitivity. In the following experiments, all of the HRRPs are assumed to have been aligned and normalized.

4.2 Recognition

4.2.1 Experimental setup

To evaluate the recognition performance of our model, in this subsection, the parameter λ in (10) is set to 0. As shown in Fig. 2, the time domain HRRP is a 1×256 real-valued vector while its corresponding spectrogram feature is a 32×256 (32 represents frequency domain while 256 is the length of time domain) matrix. Each of the TCNN and SCNN model consists of several convolutional layers and a single fully connected layer with 300 units, each layer combined with BN and ReLU non-linearity. For notational convenience, we denote TCNN_L or SCNN_L to represent a TCNN or SCNN model with L convolutional layers. Models are optimized using RMSProp with a learning rate of 0.00001 and a mini-batch size of 100. The recognition results shown in this paper are obtained by averaging classification rates from each category. All experiments

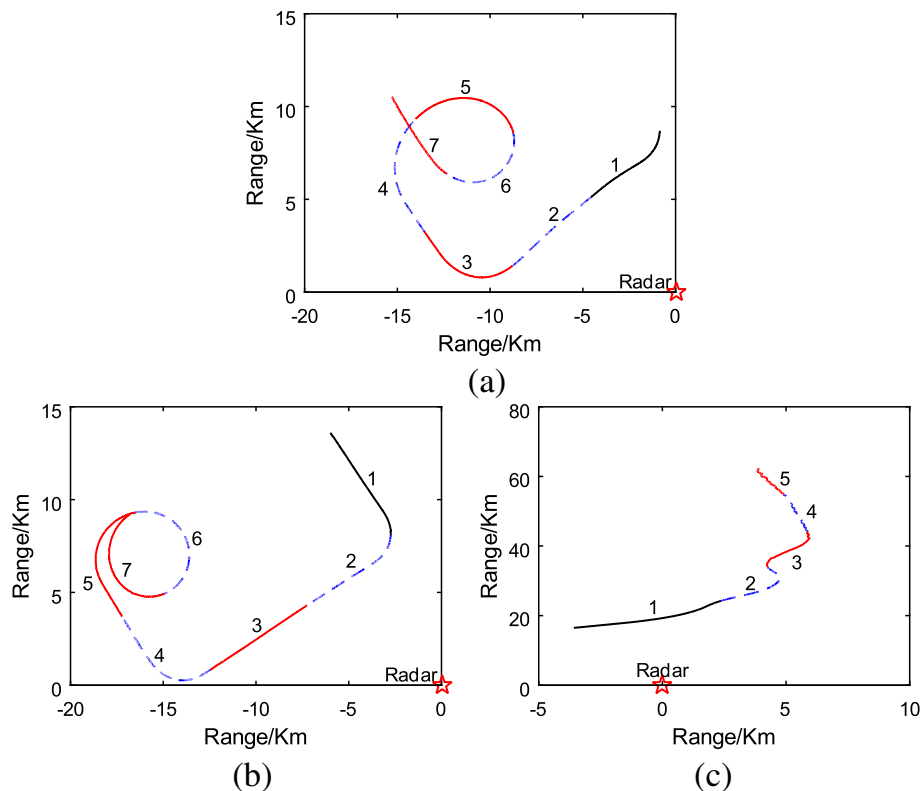


Fig. 6 Projections of target trajectories onto the ground plane. **a** An-26. **b** Cessna Citation S/II. **c** Yark-42

are performed on a desktop computer equipped with an Intel Core i7 4770 CPU, 32 GB RAM, and a graphics card NVIDIA GeForce GTX 1070.

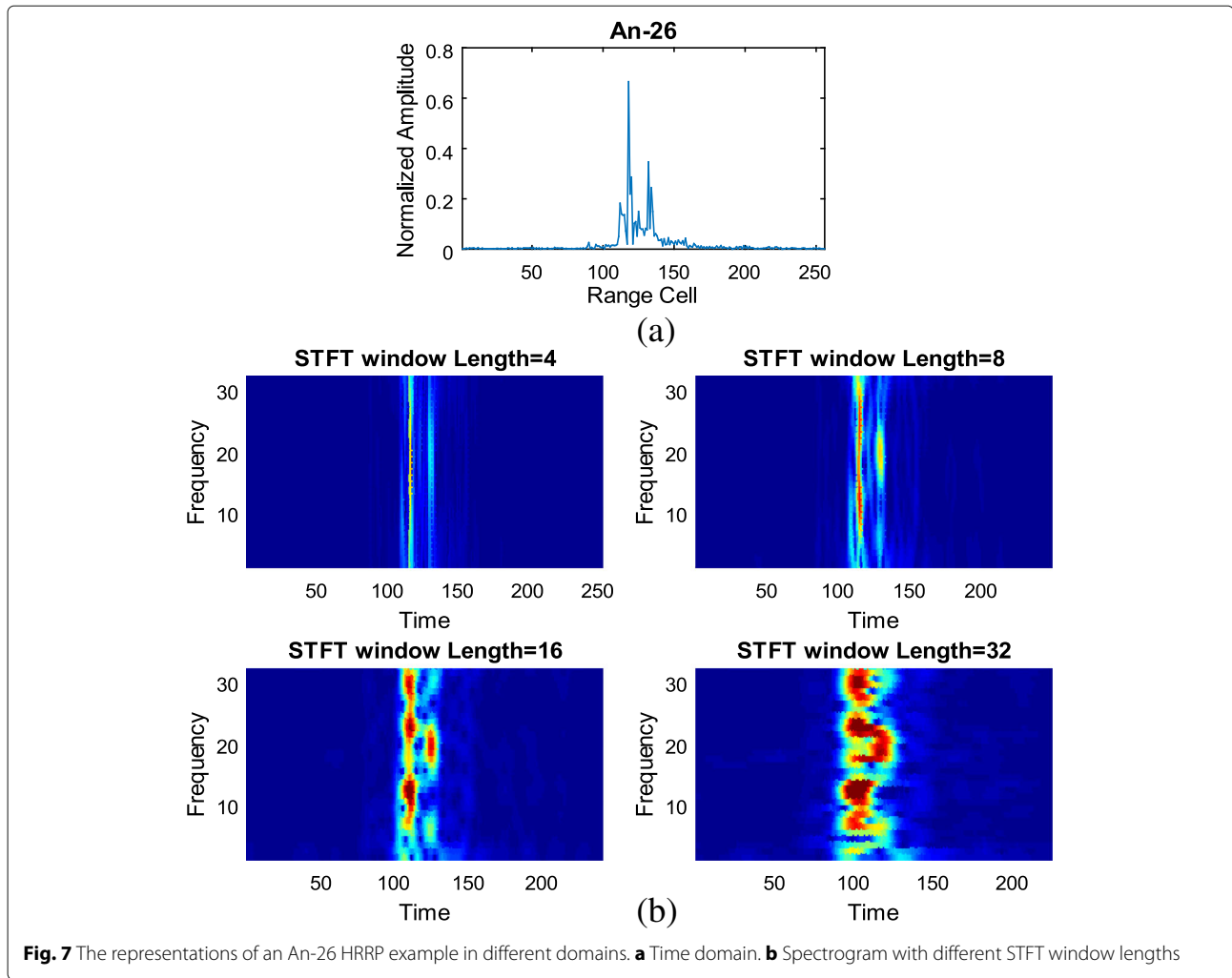
4.2.2 Impact of model parameters

In this subsection, we analyze the effect of some model parameters on the recognition performance, including STFT window length, the kernel size, and the number of convolutional layers of CNN.

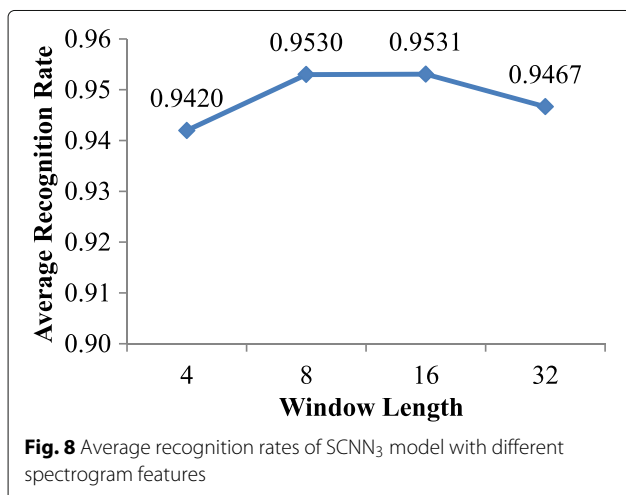
STFT window length The spectrogram feature of HRRP is obtained by calculating STFT of the aligned and normalized time domain signal, where the window length of STFT is a critical parameter which could be selected according to the radar range resolution and target sizes, since it determines the frequency resolution, which can separate the close frequency components and the time resolution, which can describe the frequency changing. As shown in Fig. 7, different STFT window lengths apparently produce different spectrogram features. Detailedly, in our measured data, since the radar range resolution is about 0.375 m, a window with length of 32 corresponds to the actual size of 12 m, which, compared with the lengths/widths of the targets falling in (14, 40) meters, makes the correlation between neighboring vectors too

high and leads to the target spectrogram representations ambiguous, while that of 4 (1.5 m) is too short to depict the effective structures of the targets. By contrast, from the spectrograms with the window length of 8 and 16, we can find the relatively clear and descriptive textures of the target. To further investigate its effect on the recognition performance, we also list the average classification accuracies based on those spectrograms with four different window lengths, where the representations with window lengths of 8 and 16 outperform those with 4 and 32 as shown in Fig. 8. Therefore, in the following experiments, we use a length of 16, windowed with a Hamming window and 32 points DFT to STFT to calculate the spectrogram feature.

Kernel size Since the HRRP echo embodies the physical composition of the target, the kernel size of CNNs actually reflects how much content a convolutional kernel is able to capture. Several first layer kernel sizes are chosen for the TCNN₃ and SCNN₃ (for time saving, we only take the first layer as an example), and the recognition results are shown in Fig. 9. In both TCNN₃ and SCNN₃, we find slightly longer kernel sizes, i.e., 1×9 , 1×15 in TCNN₃ and 32×6 , 32×10 in SCNN₃, perform better. We attribute this to a shorter kernel not being able to grasp



the effective structural features that exist in signals. Without loss of generality, in the following experiments, we fix the first layer kernel size of SCNN as 32×6 , and the higher layers of SCNN and all layers of TCNN use kernel sizes of 1×9 .



The number of CNN layer We conduct several experiments with different numbers of convolutional layers as shown in Fig. 10. The two-layer models contain two convolutional layers, both of which have 32 output channels, termed as 32-32. Similarly, the three-layer models and four-layer models have the structure of 32-32-64 and 32-32-64-128, respectively. According to the recognition results, for both SCNN and TCNN models, the best results are achieved at the structure of three convolutional layers. The two-layer models have relatively poor performance, since they do not have enough capacity of arbitrary nonlinear expressiveness. However, the performance does not benefit from enlarging the depth of the model, as the recognition rates of the four-layer models are worse than the three-layer models. Because a deeper model brings more parameters to learn which makes the representations in the higher layers excessive and overfit the training set, after all, we test it on only three types of targets. Consequently, we choose SCNN₃ and TCNN₃ for further comparisons in the following experiments.

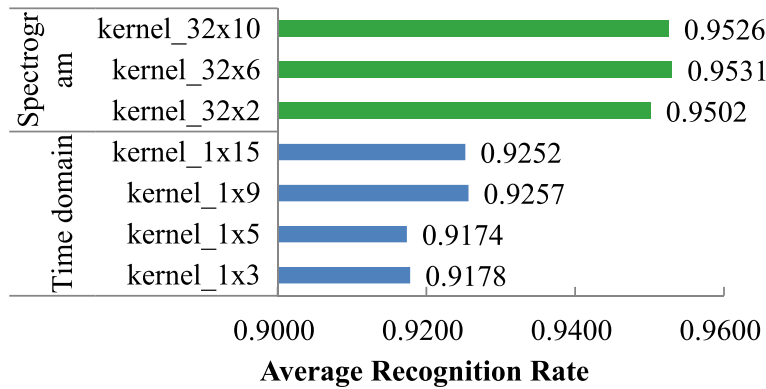


Fig. 9 Average recognition rates of TCNN₃ (blue bars) and SCNN₃ (green bars) models with different first layer kernel sizes

4.2.3 Recognition performance

The recognition accuracies and training time of SCNN₃ and TCNN₃ versus the size of training dataset are shown in Fig. 11. Here, we utilized six training datasets: 140,000, 70,000, 35,000, 17,500, 8750, and 4375, where 8750 presents 8750 samples randomly selected from a total of 140,000 training samples, similarly for other data sizes. From Fig. 11, we can see that SCNN₃ performs better than TCNN₃ by more than 2% but requires three times longer training time in each training dataset. The reason is that the spectrogram is a more informative and complex representation of HRRP, which provides more information, but also consumes more computing resources in feature learning. Furthermore, larger training dataset yields better recognition rates, but it also means a huger computational burden, as the training time increases linearly with the training data size. Therefore, in a real application, we need to consider the system resource and strike an appropriate balance between recognition performance and computation burden.

According to literature [13], the noises in the inphase and quadrature echoes of aircraft-like targets under a look-up scenario can be assumed as Gaussian white noises. To evaluate the effect of noises on the proposed method, we add simulated white noises to the inphase and quadrature component of the high signal-to-noise ratio (SNR) raw test data. The SNR is defined as:

$$\begin{aligned}
 \text{SNR} &= 10 \times \log_{10} \left(\frac{P_x}{P_{\text{Noise}}} \right) \\
 &= 10 \times \log_{10} \left(\frac{\sum_{l=1}^L P_{x_l}}{L \times P_{\text{Noise}}} \right)
 \end{aligned}
 \tag{11}$$

where P_x and P_{x_l} respectively denote the average power of HRRP and the power of the original echo per range cell, L denotes the number of range cells (here $L = 256$), and P_{Noise} denotes the power of noise. For SCNN₃, the noisy spectrogram is derived from the corresponding noisy time domain HRRP. Figure 12 depicts the average recognition rates versus SNR, in which we compare our TCNN₃, SCNN₃ models with several existing HRRP recognition

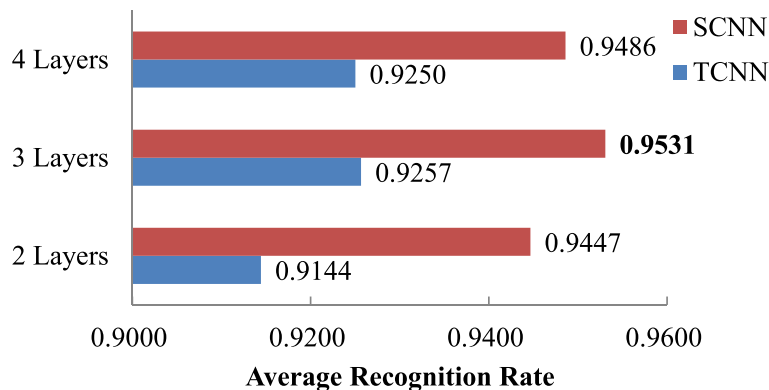
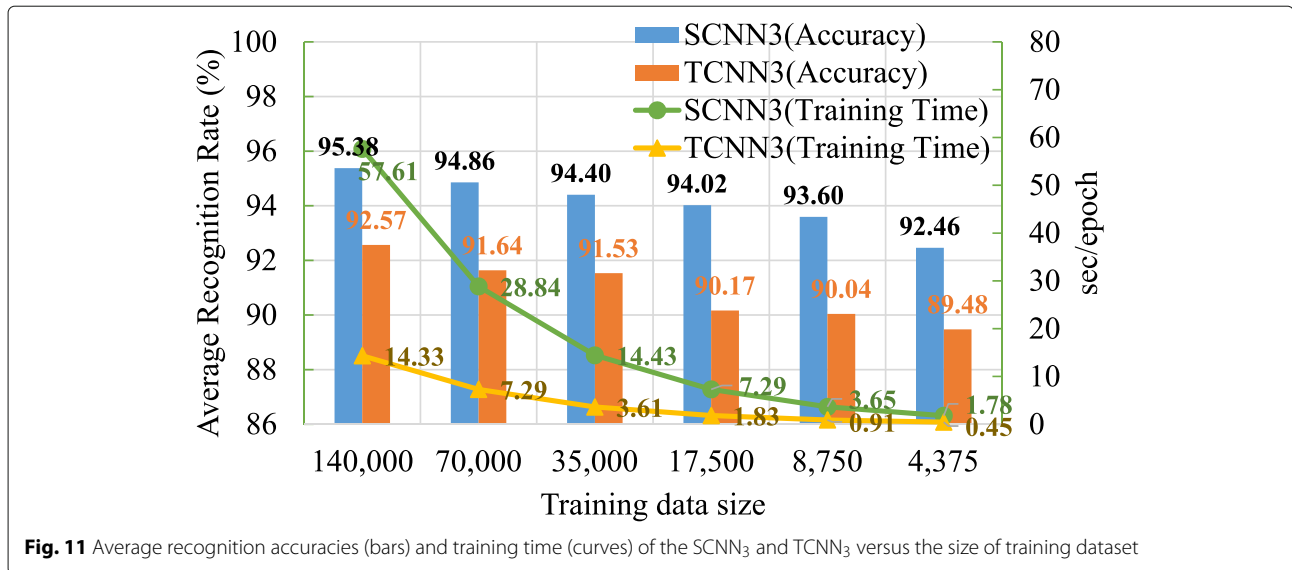


Fig. 10 Average recognition rates of TCNN and SCNN with various number of convolutional layers. The blue and red bars illustrate the recognition rates of TCNN and SCNN models, respectively



methods, including (1) maximum correlation coefficient (MCC) [9], (2) adaptive Gaussian classifier (AGC) [11], (3) linear support vector machine (LSVM) [28, 41], (4) linear discriminant analysis + LSVM (LDA), (5) principal component analysis + LSVM (PCA), (6) stacked denoising autoencoders (SDAE) [27], and (7) stacked corrective autoencoders + LSVM (SCAE) [19]. Here, “+” denotes two-stage model. MCC and AGC are two classic statistical recognition methods. LSVM is an efficient machine learning algorithm aimed at minimizing structural risk for good generalization performance. In LDA + LSVM and PCA + LSVM, LDA and PCA are only used to extract features. SDAE and SCAE are two deep fully connected neural network models. Since the three-layer SDAE and SCAE models achieve better classification accuracy in [19], we only exhibit the results of these models with three

hidden layers. It should be noted that all the models are trained with 140,000 training dataset and they take the time domain HRRP as input except SCNN₃. Generally, the recognition performance of all methods decreases with the increasing noise. When SNR ≥ 15 dB, TCNN₃ and SCNN₃ outperform other models; when SNR < 15 dB, four deep models SDAE, SCAE, TCNN₃, and SCNN₃ obtain the similar performance and are superior to shallow models. We also list the average recognition rates and computational time of these models on the raw test dataset in Table 2. In the case of time domain HRRP, the TCNN₃ outperforms other shallow or deep models, while SCNN₃ delivers the best performance, since it uses the spectrogram as the HRRP representations. Although the computation costs of TCNN₃ and SCNN₃ in training phase are pretty expensive, they can be ignored for an off-

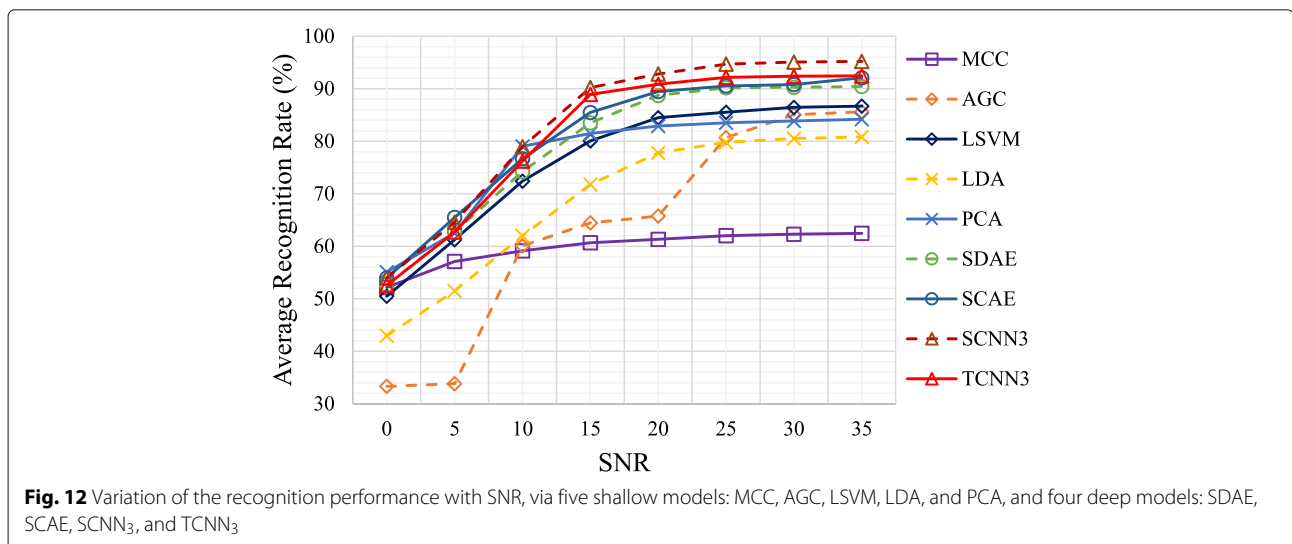


Table 2 Average recognition rates, training time, and test time of the proposed model (denoted TCNN₃, SCNN₃) with comparisons to several methods from the literature [19]

Method	Accuracy(%)	Training time (s)	Test time (s)
MCC	62.42	0.13	0.86
AGC	85.63	0.84	13.1
LSVM	86.70	0.93	0.06
LDA	81.30	0.09	0.01
PCA	83.81	0.14	0.11
SDAE	90.42	3.1/epoch	0.13
SCAE	92.03	2.7/epoch	0.13
TCNN ₃	92.57	13.8/epoch	0.10
SCNN ₃	95.31	56.6/epoch	0.65

line learning (or training) system and it is more important to evaluate the computation cost in the test phase. For test time, TCNN₃ is faster than most shallow or deep models, and it is acceptable that SCNN₃ provides significant performance improvement over a slightly longer period of time (0.65 s).

In addition, to offer an insight into the performance of the TCNN₃ and SCNN₃ on different targets, their confusion matrixes are listed on Tables 3 and 4. Obviously, the SCNN₃ benefits mainly from the decreasing misclassification of Cessna to An-26. We analyze that An-26 is a propeller-driven aircraft and its HRRPs are unstable due to the modulability of the propellers, which results in that some samples of the Cessna and An-26 are quite similar in time domain, after all, the representation of each range cell in time domain is a scalar. However, this similarity can be reduced by the spectrogram feature as it introduces the phase information to represent the signal. As shown in Fig. 13, the spectrogram representation of two highly similar time domain HRRPs can be very different, where we select two test samples which the TCNN₃ misclassified while the SCNN₃ correctly classified.

We also show the first two principal components of the three aircrafts based on different features in Fig. 14. Compared to TCNN₃ (Fig. 14b), the features extracted from SCNN₃ (Fig. 14c) are more separable, which supports again that the spectrogram is an informative representation of HRRP.

Table 3 Confusion matrix of TCNN

	Time domain feature			
	An-26	C.C. S/II	Yark-42	Total
An-26	0.9495	0.0215	0.029	1.0
C.C. S/II	0.115	0.8655	0.0195	1.0
Yark-42	0.0308	0.0108	0.9538	1.0
Avg. accuracy rate (%)	92.29%			

Table 4 Confusion matrix of SCNN

	Spectrogram feature			
	An-26	C.C. S/II	Yark-42	Total
An-26	0.9515	0.012	0.0365	1.0
C.C. S/II	0.043	0.9485	0.0085	1.0
Yark-42	0.0242	0.001	0.9658	1.0
Avg. accuracy rate (%)	95.19%			

4.2.4 Feature visualization

Besides the numerical results, it is of interest to examine what the networks “learn” at different layers. We can visualize the first-layer filters directly. Each higher layer filter can be visualized as a weighted linear combination of the lower layer filters [34, 41]. Specifically, assume the CNN model consists of L layers, a filter \mathbf{k} at layer $l > 1$ corresponds to a set of max-pooled feature maps from layer $l - 1$. One may sequentially map a filter from any layer $l > 1$ to a set of feature maps below, until at the lowest level the feature maps correspond to a segment in the input. We treat the segment in the input as the feature of filter \mathbf{k} .

Several learned filters of three-layer TCNN and SCNN at layer $l(l = 1, 2, 3)$ are exhibited in Figs. 15 and 16, respectively. For each learned filter, we show ten segments from the test samples with the highest activations on the feature maps. We can find that for both the models, filters at layer 1 (Figs. 15a and 16a) learn the specific structural features that exist in signals. Since the higher layer learns to combine the lower layer features, compared to the filters at layer 1, the layer 2 and 3 are more complex and have a small translation (Figs. 15b, c and 16b, c).

4.3 Combine recognition with rejection

The target rejection problem is taken into consideration in this section, and we add a three-layer deconvolution decoder to a three-layer SCNN to represent the data, and the kernel sizes of decoder correspond to those of the encoder, 1×9 , 1×9 , and 32×6 . As mentioned before, the rejection function of our model is realized by measuring the error between input \mathbf{x}_s and its corresponding reconstruction signal $\tilde{\mathbf{x}}_s$. Specifically, after training, we expect the in-class targets will have smaller reconstruction errors while the outlier targets produce larger ones. By setting an error threshold T_e , the rejection task can be transformed to a two-class calcification problem. Then, we use the area under a receiver operation characteristic (ROC) curve (AUC) to evaluate the rejection performance [42].

In the experiment, three airplane targets are regarded as in-class targets, while the simulated data containing 18,000 truck HRRP samples are considered as outliers. A histogram of reconstruction errors of our model ($\lambda = 10^{-6}$) between the three airplanes and the outlier

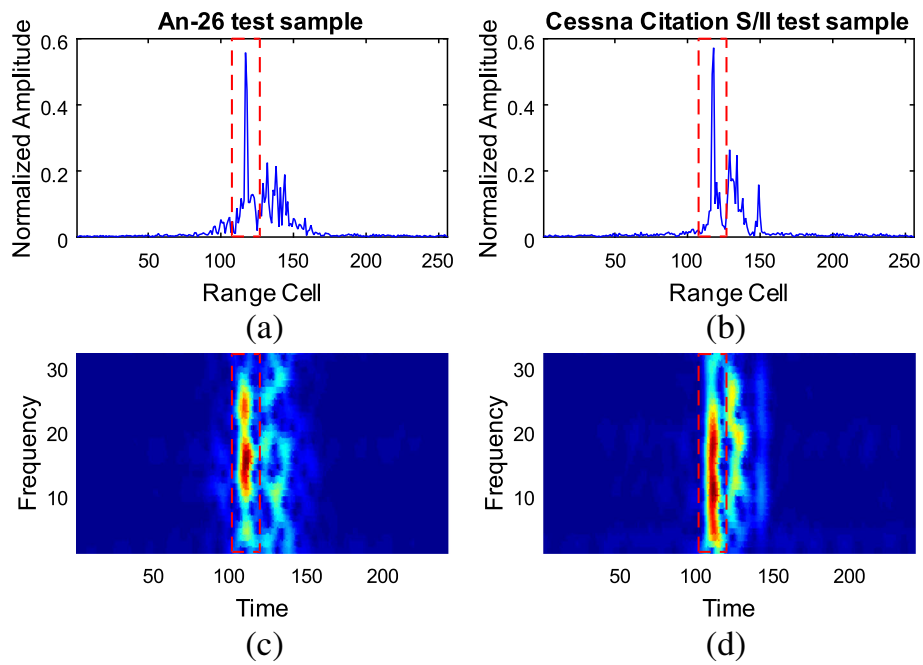


Fig. 13 Comparison of the test sample's time domain and its corresponding spectrogram representations from different targets. **a, c** An-26. **b, d** Cessna Citation S/II

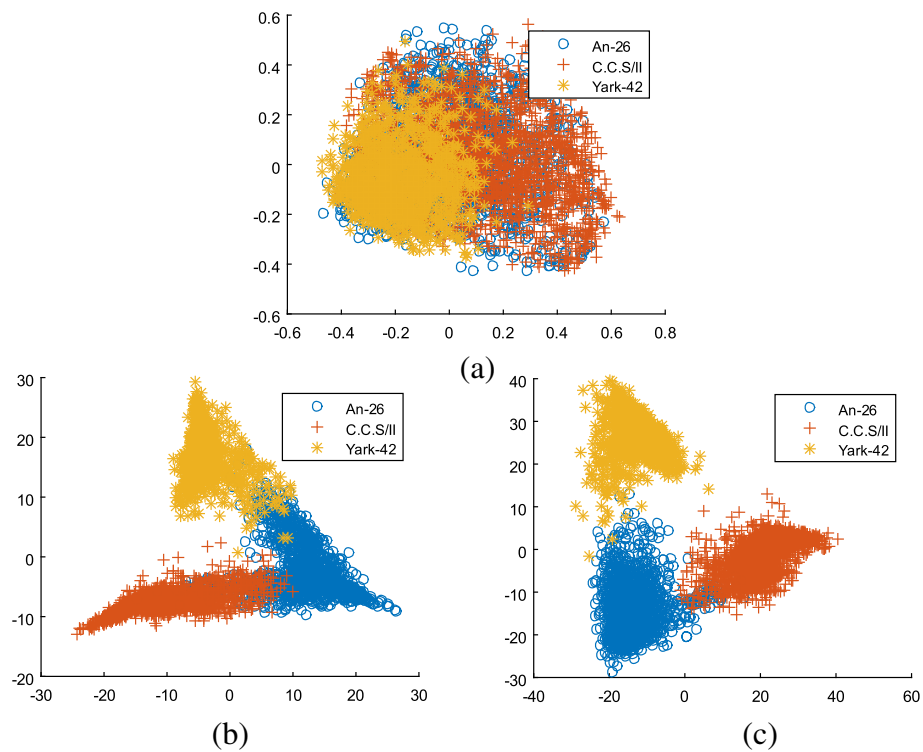


Fig. 14 Visualizations of the first two principal components of the test HRRP samples and their corresponding features from different models. **a** The original time domain. **b** TCNN. **c** SCNN

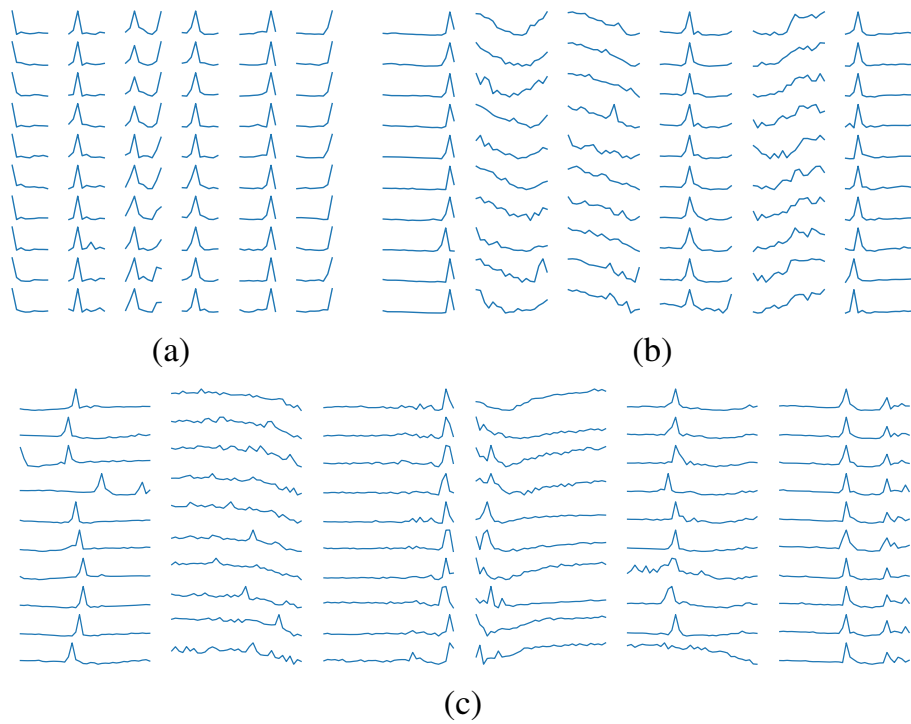


Fig. 15 Visualization of filters learned from a three-layer TCNN where each column represents a type of filter. **a** Layer 1. **b** Layer 2. **c** Layer 3

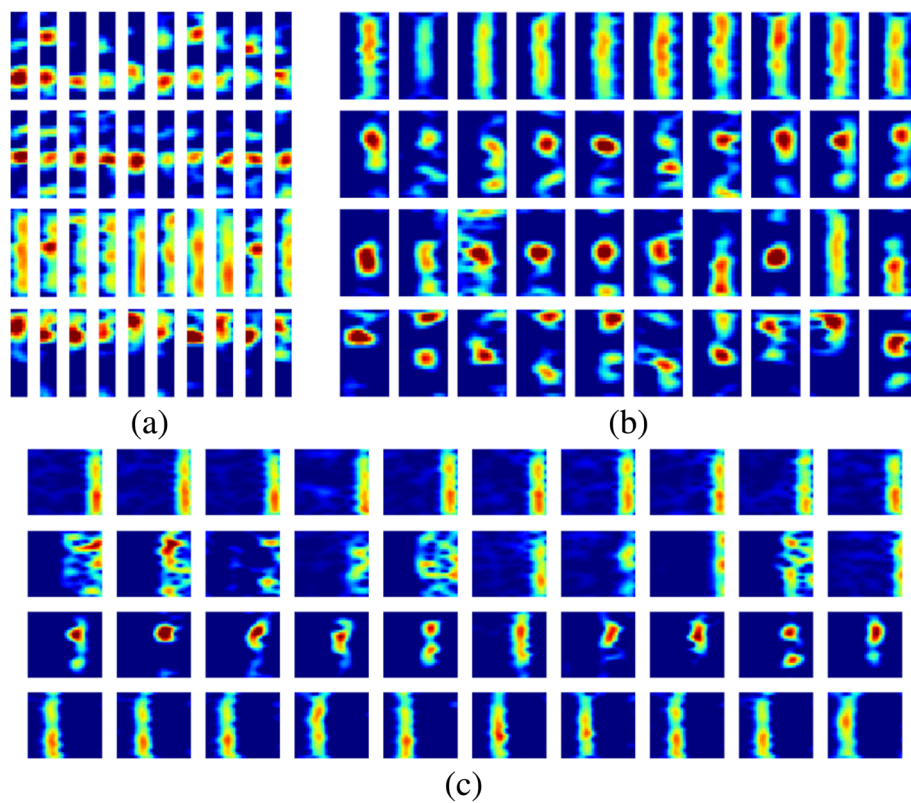
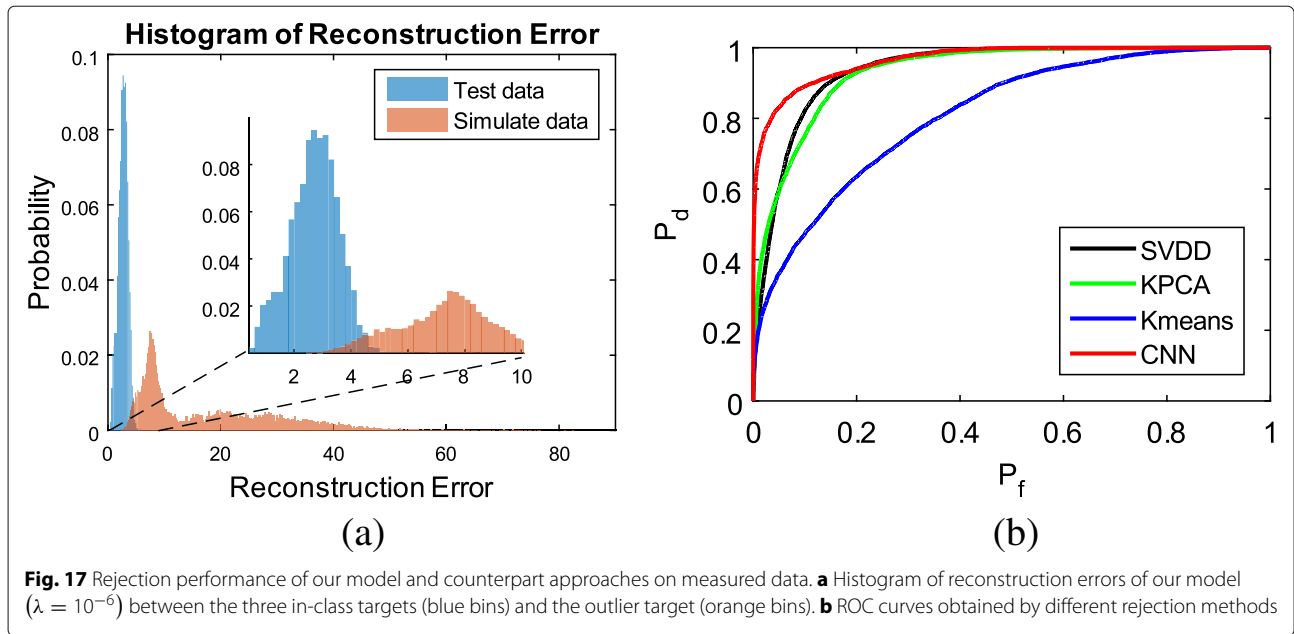


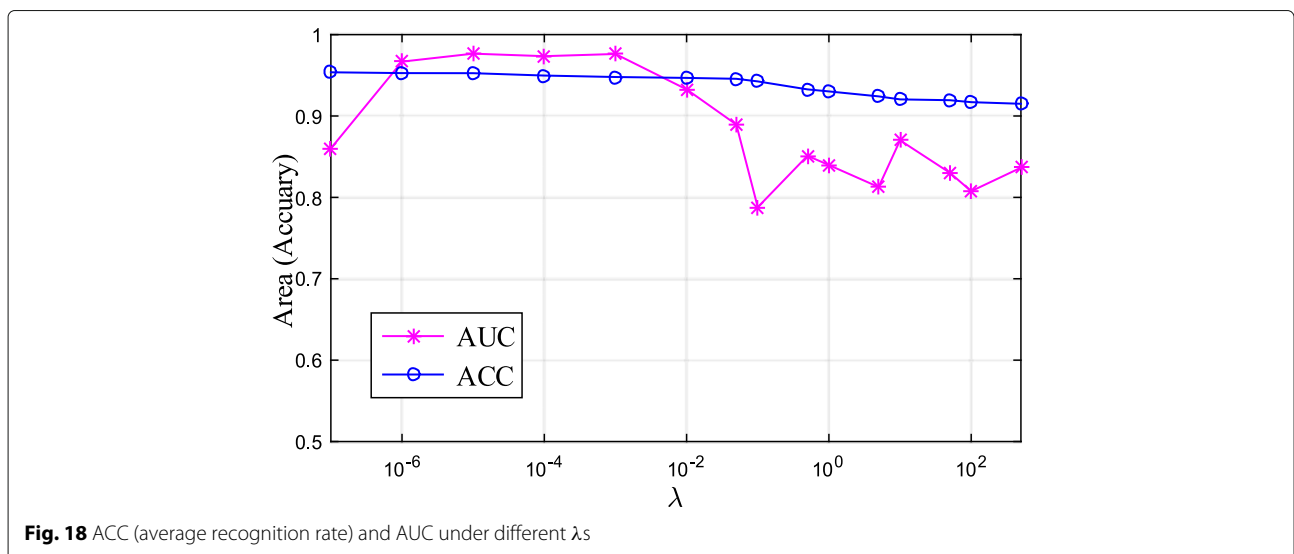
Fig. 16 Visualization of filters learned from a three-layer SCNN where each row represents a type of filter. **a** Layer 1. **b** Layer 2. **c** Layer 3



target is shown in Fig. 17a. We also consider support vector domain description with Gaussian kernel (KSVDD) [43], Kernel PCA with Gaussian kernel (KPCA) [44], and K-means [43] as counterparts in the experiment. For KSVDD and KPCA, the kernel parameters and the number of eigenvectors (for KPCA) are chosen by cross-validation. The ROC curves of KSVDD, KPCA, K-means, and our model ($\lambda = 10^{-6}$) are shown in Fig. 17b, and the AUC values of them are 0.9385, 0.9335, 0.8128, and 0.9662, respectively. We can see that the proposed model achieves better rejection of the outliers than other classifiers (or models). This can be attributed to the fact that the highly nonlinear transformations in decoder not only

provide our model a good reconstructor but also capture features that are relevant to the training data.

The hyper-parameter λ is used to balance the recognition loss and the reconstruction loss. In order to figure out how the λ influences the recognition and rejection result, several λ s are chosen in our experiments to investigate. Figure 18 shows the AUC and the recognition rate varies with λ . We can notice that the parameter λ affects both the recognition and rejection performance. When λ is small, e.g., $\lambda = 10^{-7}$, the model focuses on recognition and leads to a poor reconstruction. When the λ gets larger, e.g., around 10^{-6} to 10^{-3} , the recognition decreases a little and the rejection performs well. However, it is



interesting to see that with λ going too large ($\lambda > 10^{-3}$), the model does not exhibit good performance on rejection task, since it has strong capability of signal reconstruction for both of existing and outlier targets, which will make the model ambiguous. In practical, the values that satisfy the radar system recognition and rejection performance requirements can be selected as λ . Considering the recognition and rejection performance, a λ around 10^{-6} to 10^{-3} is suitable for our HRRP data.

5 Conclusion

In this paper, in order to explore the correlation between range cells and extract the structured discriminative features in HRRP, we have proposed a novel approach to address HRRP RATR and rejection problem based on convolutional neural network. Since the spectrogram is more informative than time domain HRRP with frequency and phase information, besides using one-dimensional CNN to handle HRRP in time domain, we also specialized a two-dimensional CNN model for its spectrogram representations.

Furthermore, we plugged a deconvolutional network into the model to solve the outlier rejection problem. By measuring the reconstruction error between the input and the output in the decoder part, the model can identify whether a sample is an outlier or not. Experiments based on the measure radar data have shown that our model is competitive on both of recognition and rejection task. In the future, to make use of the phase information, we will design the complex-valued CNNs to handle the complex HRRPs.

Abbreviations

1-D: One-dimensional; 2-D: Two-dimensional; AGC: Adaptive Gaussian classifier; AUC: Area under an ROC curve; BN: Batch normalization; CNN: Convolutional neural network; DBN: Deep belief networks; DFT: Discrete Fourier transformation; FA: Factor analysis; HRR: High-resolution radar; HRRP: High-resolution range profile; KPCA: Kernel PCA; KSVDD: Support vector domain description with Gaussian kernel; LDA: Linear discriminant analysis; LOS: Line-of-sight; LSVM: Linear support vector machine; MCC: Maximum correlation coefficient; PCA: Principal components analysis; RATR: Radar automatic target recognition; ROC: Receiver operation characteristic; SCAE: Stacked corrective autoencoders; SCNN: CNN for spectrogram domain; SDAE: Stacked denoising autoencoder; STFT: Short-time Fourier transform; TCNN: CNN for time domain

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of this manuscript.

Funding

This work is partially supported by the Thousand Young Talent Program of China, NSFC (61771361), the 111 Project (B18039), and the National Science Fund for Distinguished Young Scholars of China (61525105).

Availability of data and materials

Not available online. Please contact author for data requests.

Authors' contributions

All authors have contributed equally. All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 14 May 2018 Accepted: 10 January 2019

Published online: 23 January 2019

References

1. D. Orlando, G. Ricci, Adaptive radar detection and localization of a point-like target. *IEEE Trans. Signal Process.* **59**(9), 4086–4096 (2011)
2. A. Aubry, A. De Maio, G. Foglia, C. Hao, Orlando D., Radar detection and range estimation using oversampled data. *IEEE Trans. Aerosp. Electron. Syst.* **51**(2), 1039–1052 (2015)
3. R. A. Mitchell, J. J. Westerkamp, Robust statistical feature based aircraft identification. *IEEE Trans. Aerosp. Electron. Syst.* **35**(3), 1077–1094 (1999)
4. S. P. Jacobs, J. A. O'Sullivan, Automatic target recognition using sequences of high resolution radar range-profiles. *IEEE Trans. Aerosp. Electron. Syst.* **36**(2), 364–381 (2000)
5. X. Liao, P. Runkle, L. Carin, Identification of ground targets from sequential high-range-resolution radar signatures. *IEEE Trans. Aerosp. Electron. Syst.* **38**(4), 1230–1242 (2002)
6. F. Zhu, X. D. Zhang, Y. F. Hu, D. Xie, Nonstationary hidden Markov models for multiaspect discriminative feature extraction from radar targets. *IEEE Trans. Signal Process.* **55**(5), 2203–2214 (2007)
7. J. Chai, H. Liu, Z. Bao, Combinatorial discriminant analysis: supervised feature extraction that integrates global and local criteria. *Electron. Lett.* **45**(18), 934–935 (2009)
8. X.-D. Zhang, Y. Shi, Z. Bao, A new feature vector using selected bispectra for signal classification with application in radar target recognition. *IEEE Trans. Signal Process.* **49**(9), 1875–1885 (2001)
9. L. Du, H. Liu, Z. Bao, M. Xing, Radar HRRP target recognition based on higher order spectra. *IEEE Trans. Signal Process.* **53**(7), 2359–2368 (2005)
10. L. Du, H. Liu, Z. Bao, J. Zhang, Radar automatic target recognition using complex high-resolution range profiles. *IET Radar Sonar Navig.* **1**(1), 18–26 (2007)
11. L. Du, H. Liu, Z. Bao, Radar HRRP statistical recognition: parametric model and model selection. *IEEE Trans. Signal Process.* **56**(5), 1931–1944 (2008)
12. L. Shi, P. Wang, H. Liu, L. Xu, Z. Bao, Radar HRRP statistical recognition with local factor analysis by automatic Bayesian Ying-Yang harmony learning. *IEEE Trans. Signal Process.* **59**(2), 610–617 (2011)
13. L. Du, P. Wang, H. Liu, M. Pan, F. Chen, Z. Bao, Bayesian spatiotemporal multitask learning for radar HRRP target recognition. *IEEE Trans. Signal Process.* **59**(7), 3182–3196 (2011)
14. L. Du, H. Liu, P. Wang, B. Feng, M. Pan, Z. Bao, Noise robust radar HRRP target recognition based on multitask factor analysis with small training data size. *IEEE Trans. Signal Process.* **60**(7), 3546–3559 (2012)
15. X. Zhang, B. Chen, H. Liu, L. Zuo, B. Feng, Infinite max-margin factor analysis via data augmentation. *Pattern Recog.* **52**, 17–32 (2016)
16. B. Feng, L. Du, H. W. Liu, F. Li, in *IEEE Cie International Conference on Radar. Radar HRRP target recognition based on K-SVD algorithm* (IEEE, Piscataway, 2012), pp. 642–645
17. D. Zhou, Radar target HRRP recognition based on reconstructive and discriminative dictionary learning. *Signal Process.* **126**, 52–64 (2016)
18. H. Yan, Z. Zhang, G. Xiong, W. Yu, Radar HRRP recognition based on sparse denoising autoencoder and multi-layer perceptron deep model. *UPINLBS.* **1**, 283–288 (2016)
19. B. Feng, B. Chen, H. Liu, Radar HRRP target recognition with deep networks. *Pattern Recog.* **61**, 379–393 (2017)
20. M. Pan, L. Du, P. Wang, H. Liu, Z. Bao, Multi-task hidden Markov modeling of spectrogram feature from radar high-resolution range profiles. *EURASIP J. Adv. Sig. Proc.* **2012**(1), 411 (2012)
21. B. Chen, H. Liu, J. Chai, Z. Bao, Large margin feature weighting method via linear programming. *IEEE Trans. Knowl. Data Eng.* **21**(10), 1475–1488 (2009)
22. L. Du, H. Liu, Z. Bao, in *International Conference on Radar. Radar automatic target recognition based on complex high-resolution range profiles* (IEEE, Piscataway, 2006), pp. 1–5

23. L. Du, H. Liu, Z. Bao, J. Zhang, A two-distribution compounded statistical model for radar HRRP target recognition. *IEEE Trans. Signal Process.* **54**, 2226–2238 (2006)
24. B. Chen, H. Liu, L. Yuan, Z. Bao, Adaptively segmenting angular sectors for radar HRRP automatic target recognition. *EURASIP J. Adv. Signal Process.* **2008**(1), 641709 (2008)
25. H.-w. Liu, B. Chen, B. Feng, L. Du, Radar high-resolution range profiles target recognition based on stable dictionary learning. *IET Radar Sonar Navig.* **10**(2), 228–237 (2016)
26. G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks. *Science.* **313**(5), 504–507 (2006)
27. P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders. *ICML*, 1096–1103 (2008)
28. Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives. *IEEE Trans. Pattern. Anal. Mach. Intell.* **35**(8), 1798–828 (2013)
29. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks. *NIPS*, 1097–1105 (2012)
30. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. *CoRR*. [abs/1409.1556](https://arxiv.org/abs/1409.1556) (2014). <http://arxiv.org/abs/1409.1556>. arXiv, 1409.1556
31. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions. *CVPR*. **1**, 1–9 (2015)
32. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. *CVPR*. **1**, 770–778 (2016)
33. S. Ioffe, C. Szegedy, in *Proceedings of the 32nd International Conference on Machine Learning. vol. 37, Proceedings of Machine Learning Research*, ed. by F. Bach, D. Blei. Batch normalization: Accelerating deep network training by reducing internal covariate shift (PMLR, Lille, 2015), pp. 448–456. 07–09 July
34. H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, in *the 26th Annual International Conference. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations* (ACM Press, New York, 2009), pp. 1–8
35. T. N. Sainath, A. R. Mohamed, B. Kingsbury, B. Ramabhadran, in *IEEE International Conference on Acoustics, Speech and Signal Processing. Deep convolutional neural networks for LVCSR* (IEEE, Piscataway, 2013), pp. 8614–8618
36. Y. Zhang, W. Chan, N. Jaitly, Very deep convolutional networks for end-to-end speech recognition. *Int. Conf. Acoust. Speech Signal Process.* **1**, 4845–4849 (2017)
37. V. Nair, G. E. Hinton, Rectified linear units improve restricted Boltzmann machines. *ICML*, 807–814 (2010)
38. R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, C. D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions. *EMNLP*, 151–161 (2011)
39. D. P. Kingma, D. J. Rezende, S. Mohamed, M. Welling, Semi-supervised learning with deep generative models. *Adv. Neural Inf. Process. Syst.* **4**, 3581–3589 (2014)
40. J. L. Walker, Range-Doppler imaging of rotating objects. *IEEE Trans. Aerosp. Electron. Syst.* **16**(1), 23–52 (1980)
41. B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, L. Carin, Deep learning with hierarchical convolutional factor analysis. *IEEE Trans. Pattern. Anal. Mach. Intell.* **35**(8), 1887–1901 (2013)
42. J. Davis, M. Goadrich, The relationship between Precision-Recall and ROC curves. *ICML*. **1**, 233–240 (2006)
43. D. M. J. Tax, One-class classification. PhD thesis, Delft University of Technology (2001)
44. H. Hoffmann, Kernel PCA for novelty detection. *Pattern Recognit.* **40**(3), 863–874 (2007)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com