Cognitive Computation
https://doi.org/10.1007/s12559-019-09639-x

# A Novel Semi-Supervised Convolutional Neural Network Method for Synthetic Aperture Radar Image Recognition

Zhenyu Yue[1] · Fei Gao[1] · Qingxu Xiong[1] · Jun Wang[1] · Teng Huang[1] · Erfu Yang[2] · Huiyu Zhou[3]

Received: 20 October 2018 / Accepted: 10 March 2019


**Abstract**
Synthetic aperture radar (SAR) automatic target recognition (ATR) technology is one of the research hotspots in the field of image cognitive learning. Inspired by the human cognitive process, experts have designed convolutional neural network (CNN)-based SAR ATR methods. However, the performance of CNN significantly deteriorates when the labeled samples are insufficient. To effectively utilize the unlabeled samples, we present a novel semi-supervised CNN method. In the training process of our method, the information contained in the unlabeled samples is integrated into the loss function of CNN. Specifically, we first utilize CNN to obtain the class probabilities of the unlabeled samples. Thresholding processing is performed to optimize the class probabilities so that the reliability of the unlabeled samples is improved. Afterward, the optimized class probabilities are used to calculate the scatter matrices of the linear discriminant analysis (LDA) method. Finally, the loss function of CNN is modified by the scatter matrices. We choose ten types of targets from the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset. The experimental results show that the recognition accuracy of our method is significantly higher than other semi-supervised methods. It has been proved that our method can effectively improve the SAR ATR accuracy when labeled samples are insufficient.

**Keywords** Synthetic aperture radar · Image recognition · Convolutional neural network · Semi-supervised learning · Linear discriminant analysis

## Introduction

Synthetic aperture radar (SAR) has been widely used due to its high resolution and penetrating ability [1–3]. SAR automatic target recognition (ATR) technology aims to automatically recognize the targets from SAR images. With an increasing amount of data acquired by SAR imaging systems, SAR ATR has become one of the research hotspots [4, 5]. Based on the cognitive system, humans are able to recognize targets quickly and accurately. Inspired by this, various methods that imitate the human cognitive system have been proposed to improve the SAR ATR accuracy.

During the human cognitive process, image signals acquired by the retina first go through the primary visual cortex for extracting edge and orientation features, followed by the generation of shape and contour features [6]. In this way, image signals pass through the higher-level visual cortex and we can obtain the more abstract features. Hence, human image cognition is a process of obtaining abstract features through layer-by-layer visual cortex [7, 8]. Inspired by this process, people have established various neural network models. By simulating the whole process of the human visual system from the retina to the visual cortex, an effective SAR image feature extraction method was proposed in [9]. Using the hierarchical perceptual inference process embedded in the cortex, Spratling et al. proposed a hierarchical neural network for visual object recognition [10]. Ren et al. proposed a multiple convolutional neural network (CNN) based on the human visual system [11].

CNN simulates the visual cortex using convolution layers and each convolution layer contains several convolution kernels for extracting abstract features. Compared with the other

✉ Fei Gao
feigao2000@163.com

[1] School of Electronic Information Engineering, Beihang University, Beijing 100191, China

[2] Strathclyde Space Institute, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK

[3] Department of Informatics, University of Leicester, Leicester LE1 7RH, UK

Published online: 19 March 2019

neural network models, CNN has been successfully applied to SAR ATR due to its powerful feature extraction capability [12–14]. Chen et al. proposed a CNN model to transform SAR images into a set of feature maps [15]. Gao et al. proposed a new SAR ATR method by combining CNN and support vector machine (SVM) [16]. It has been proved that the CNN model can effectively improve the SAR ATR accuracy. However, the CNN model needs a large number of labeled samples in the training process. When the labeled samples are insufficient, the recognition accuracy of the CNN decreases significantly [17]. Because of the imaging nature, speckle noise and clutters exist in SAR images, which increase the difficulty of the sample annotation. As a result, the number of the labeled samples is insufficient, which restricts the application of CNN in SAR ATR. In recent years, researchers are focusing on improving the SAR ATR accuracy with a small labeled dataset [18]. However, compared with labeled samples, unlabeled samples are easier to acquire. Besides, unlabeled samples also contain a wealth of information which helps to improve the SAR ATR accuracy.

The human cognition does not need a large number of labeled samples [19, 20]. Based on the labeled samples, we are able to utilize the unlabeled samples and revise the object recognition criteria that we learned previously, which is known as the semi-supervised mechanism [21]. Inspired by this mechanism, semi-supervised learning methods have been designed to improve the SAR ATR accuracy when the labeled samples are insufficient [22, 23]. The common semi-supervised learning methods include self-training, co-training, graph-based methods, and semi-supervised SVM [24–26]. Lv et al. presented a semi-supervised predictive sparse decomposition method for feature learning [27]. To solve the online semi-supervised learning problems, Ding et al. proposed a novel manifold regularized model in a reproducing kernel Hilbert space [28].

Recently, researchers are devoted to combining semi-supervised learning methods with neural network models. To effectively utilize the unlabeled samples, a semi-supervised deep learning model based on ladder networks was proposed in [29]. Samuli and Timo presented two simple and efficient semi-supervised CNN models, i.e., the Pi model and the temporal ensembling model [30]. The two models are based on the self-ensembling method where the "pseudo labels" of the unlabeled samples are generated by the outputs of CNN. Although the above semi-supervised methods are proved to be effective, it has been found that semi-supervised methods cannot always improve the image recognition accuracy [31, 32]. For example, the Pi model and the temporal ensembling model use CNN to generate the pseudo labels of unlabeled samples. However, the reliability of the unlabeled samples will be significantly reduced if the pseudo labels are incorrect. As a result, the performance of CNN will be worse. The reliability of the unlabeled samples has a decisive influence on semi-supervised methods.

In this paper, a novel semi-supervised CNN method is proposed to improve the SAR ATR accuracy. The innovations of our method are as follows.

CNN is utilized to obtain the class probabilities of the unlabeled samples. To improve the reliability of the unlabeled samples, we perform thresholding processing on the class probabilities.2. Based on the optimized class probabilities, we design a new linear discriminant analysis (LDA) method to utilize the information contained in the unlabeled samples. Then the loss function of CNN is modified by the scatter matrices of the new LDA method.

The rest of this paper is arranged as follows. In the "Preliminary" section, CNN and the LDA methods are briefly introduced. "The Proposed Method" section describes the principle of our method in detail. The experiments are performed in the "Experiments" section. We summarize our contribution in the "Conclusion" section.

## Preliminary

### Convolutional Neural Network

CNN is mainly composed of convolution, pooling, and fully connected layers. The convolution layers are used to extract image features. The pooling layers decrease the risk of overfitting by reducing the dimensions of features. The fully connected layers are used to integrate the image features. The training process of CNN consists of forward and backward propagation [33, 34].

In the forward propagation process, the current layer of CNN receives the output of the previous layer, which is expressed as follows:

$$\left.\begin{array}{l} z^l = w^l a^{l-1} + b^l \\ a^l = \sigma\left(z^l\right) \end{array}\right\} \tag{1}$$

where $l$ denotes the $l^{th}$ layer. $z^l$, $w^l$, and $b^l$ represent the weighted input of the $l^{th}$ layer, the weight matrix, and the bias matrix, respectively. $\sigma$ denotes the nonlinear activation function and $a^l$ represents the actual output of the $l^{th}$ layer. If $l = 1$, $a^0$ represents the pixel value of the input image.

In the backward propagation process, the parameters $w^l$ and $b^l$ of CNN are updated using the back propagation (BP) algorithm. In detail, the BP algorithm first constructs a loss function based on the actual and the expected outputs of CNN. Afterward, the gradient descent method is utilized to update the parameters $w^l$ and $b^l$ along the gradient decent direction of

the loss function. Suppose that $E_0$ is the loss function and $L$ denotes the number of the layers of CNN. Then the error vector of the output layer is expressed as follows:

$$\delta^L = \frac{\partial E_0}{\partial z^L} \tag{2}$$

The error vector of the $(l-1)^{th}$ layer can be calculated by the error vector of the $l^{th}$ layer. Therefore, the error vector $\delta^l$ of each layer can be calculated by the Chain Rule:

$$\delta^l = w^{l+1}\delta^{l+1} \circ \sigma'\left(z^l\right) \tag{3}$$

where the symbolic $\circ$ represents the element-wise product of the two vectors. The partial derivative of $E_0$ to $w^l$ and $b^l$ can be calculated by Eq. (4):

$$\left.\begin{aligned}\frac{\partial E_0}{\partial w^l} &= \frac{\partial E_0}{\partial z^l} \circ \frac{\partial z^l}{\partial w^l} = \delta^l \circ a^{l-1} \\ \frac{\partial E_0}{\partial b^l} &= \frac{\partial E_0}{\partial z^l} \circ \frac{\partial z^l}{\partial b^l} = \delta^l\end{aligned}\right\} \tag{4}$$

Then the changes of $w^l$ and $b^l$ are calculated:

$$\left.\begin{aligned}\Delta w^l &= -\eta\frac{\partial E_0}{\partial w^l} \\ \Delta b^l &= -\eta\frac{\partial E_0}{\partial b^l}\end{aligned}\right\} \tag{5}$$

where $\eta$ denotes the learning rate.

### Linear Discriminant Analysis

LDA is used to search a subspace where the samples of different classes are distant from each other while the samples of the same class are close to each other [35, 36]. In the case of binary classification, given the training dataset $D = \{(x_i, y_i)\}_{i=1}^m$, where $x_i$ denotes the training samples and $y_i \in \{0, 1\}$ denotes their labels, $m$ represents the number of the samples in the training dataset. Suppose that $\mu_i$ and $C_i$ represent the mean vector and covariance matrices of the $i^{th}$ class, respectively, and $w$ denotes the projection vector. In order to make the samples of the same class as close as possible in the subspace, $w^T C_0 w + w^T C_1 w$ should be small. While $\|w^T\mu_0 - w^T\mu_1\|_2^2$ should be large to make the samples of different classes as distant as possible. Taking these into consideration, we get the optimization objective function as follows:

$$J = \frac{\|w^T\mu_0 - w^T\mu_1\|_2^2}{w^T C_0 w + w^T C_1 w} = \frac{w^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T(C_0 + C_1)w} \tag{6}$$

We define the within-class scatter matrix $S_w = C_0 + C_1$ and the between-class scatter matrix $S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$. Then the objective function Eq. (6) can be rewritten as Eq. (7), which is called the "generalized Rayleigh quotient" of $S_w$ and $S_b$.

$$J = \frac{w^T S_b w}{w^T S_w w} \tag{7}$$

Next, the LDA algorithm is extended to the field of multi-classification. Suppose that there are $N$ classes and the number of the samples of the $i^{th}$ class is $m_i$, then the total-class scatter matrix is defined as follows:

$$S_t = S_b + S_w = \sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T \tag{8}$$

where $m$ denotes the total number of the samples and $\mu$ represents the mean vector of all the samples. The within-class scatter matrix is defined as the sum of the covariance matrices for each class:

$$S_w = \sum_{i=1}^{N} C_i \tag{9}$$

According to Eqs. (8) and (9), the between-class scatter matrix is obtained:

$$S_b = S_t - S_w = \sum_{i=1}^{N} m_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{10}$$

There are various ways to construct the optimization objective function of LDA for multi-classification, and one of the common ways is expressed in Eq. (7).

## The Proposed Method

We first define the system parameters. The training dataset $X$ consists of two parts: $X = [L, U] \in R^{d \times N}$, where $L = [x_1, x_2, \cdots, x_l] \in R^{d \times l}$ represents the labeled dataset and $U = [x_{l+1}, x_{l+2}, \cdots, x_{l+u}] \in R^{d \times u}$ represents the unlabeled dataset. $d$ denotes the dimension of the samples. $N$, $l$, and $u$ represent the number of the samples in $X$, $L$, and $U$, respectively.

As shown in Fig. 1, the training process of our method is composed of three parts: feature extraction, supervised learning, and semi-supervised learning. In the feature extraction, we use CNN to extract the features of the samples. $L'$ and $U'$ represent the feature vector of the labeled and unlabeled datasets, respectively. In the process of supervised learning, the labeled samples are utilized to obtain the supervised component of the loss function for CNN. The semi-supervised learning process which consists of two steps is the core of our method. We first utilize CNN to obtain the class probabilities of the unlabeled samples. To improve the reliability of the unlabeled samples, thresholding processing is performed to optimize the class probabilities. Afterward, we utilize the optimized class probabilities to calculate the scatter matrices of the new LDA method. Then the unsupervised component of the loss function is constructed using

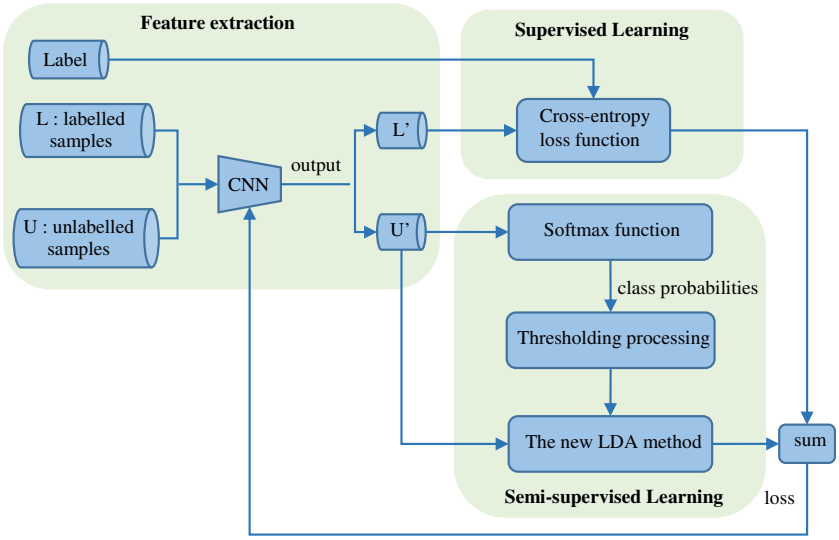


**Fig. 1** The flowchart of the training process

the scatter matrices. Next, the two steps of the semi-supervised learning process are described in detail.

## Class Probabilities of Unlabeled Samples

To effectively utilize the unlabeled samples, we adopt CNN to obtain their class probabilities. Compared with optical images, the signal-to-noise ratio (SNR) and resolutions of SAR images are relatively low. Therefore, CNN models such as AlexNet and VGGNet for optical images are not suitable for SAR images. The network structure of our CNN model is shown in Fig. 2. The size of the input images is $64 \times 64$. Conv1, Conv2, and Conv3 represent the convolution layers, which contain 20, 40, and 80 kernels with sizes $3 \times 3$, $4 \times 4$, and $3 \times 3$, respectively. We adopt the Relu activation function in the convolution layers. Maxpool denotes the maximum pooling operation, and the pool size is $2 \times 2$. The Flatten layer stretches the output of Conv3 to create a 2880 dimensional column vector. Linear1, Linear2, and Linear3 represent the fully connected layers whose output dimensions are 2880, 2880, and 10, respectively. The Relu activation function is also adopted in the fully connected layers.

Suppose that the number of the neurons in the output layer is K, that is, the CNN eventually divides the input images into K classes. As expressed in Eq. (11), we utilize the softmax function to normalize the output of CNN, and the class probabilities of the unlabeled samples are obtained.

$$p_k = \frac{e^{a_k}}{\sum_{j=1}^{K} e^{a_j}} \tag{11}$$

where $[a_1, a_2, \cdots, a_K]$ is the output of CNN. Hence, the class probabilities of a sample can be represented as $[p_1, p_2, \cdots, p_K]$, where $p_k$ denotes the probability of the sample belonging to the $k^{th}$ class. The larger the value of $p_k$ is, the greater the probability is. $\sum_{k=1}^{K} p_k = 1$, and if one item increases, the sum of the others will be decreased.

The reliability of a sample is related to its real label and class probabilities. We define the reliability factor (RF) to measure the reliability of samples. As expressed in Eq. (12), $p_{\text{real}}$ denotes the probability of a sample belonging to the class that corresponds to its real label. The larger the value of RF is,

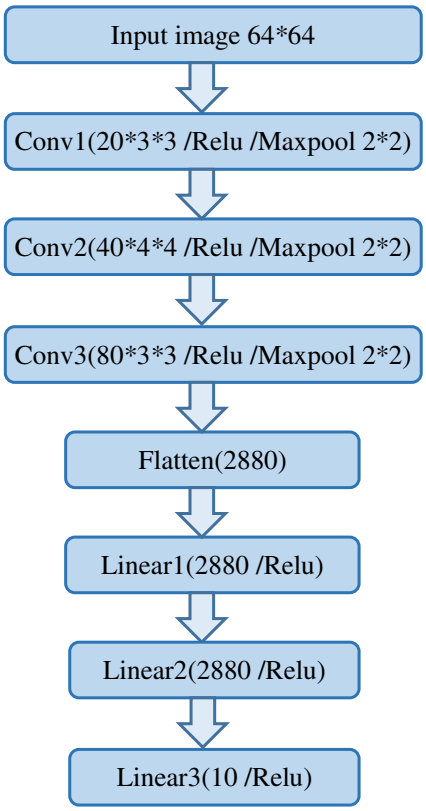


**Fig. 2** The network structure of our CNN model

the more reliable of the sample. In general, a sample with the RF value greater than 0.9 can be regarded as a reliable sample.

$$\mathrm{RF} = \frac{p_{\mathrm{real}}}{\sum_{k=1}^{K} p_k} \tag{12}$$

Suppose that $p^i \in R^{1\times N}$ denotes the class probabilities of the unlabeled samples belonging to the $i^{th}$ class. To improve the reliability of the unlabeled samples, we apply thresholding processing to $p^i$:

$$p_j^i = \begin{cases} 0, p_j^i < t \\ p_j^i, \text{others} \end{cases}, i\in[1,2,\cdots,K], j\in[1,2,\cdots,u] \tag{13}$$

where $p_j^i$ represents the $j^{th}$ element of $p^i$ and $t$ is the threshold. The greater the value of $t$ is, the stricter the reliability requirement is. If the maximum class probability of a sample is less than $t$, all the class probabilities of the sample will be set to 0. The unlabeled samples are utilized based on the class probabilities and the new LDA method. Thus, the unlabeled samples whose class probabilities are all set to 0 will not be utilized in the training process.

## The New LDA Method

After obtaining the class probabilities, how to utilize the unlabeled samples is the key to improving the recognition accuracy of the CNN model. Most of the semi-supervised CNN methods extend the labeled dataset by using the CNN to label the unlabeled samples. Then the CNN is retrained with the extended labeled dataset. However, when the labeled samples are insufficient, the generalization ability of the CNN model is weak. As a result, the pseudo labels for the unlabeled samples are not credible, which will lead to accuracy reduction. The LDA method constructs an optimization function based on the within-class and between-class distance of the samples. Using the projection vector, the samples of different classes are distant from each other while the samples of the same class are close to each other. We design a new LDA method to exploit the unlabeled samples. The scatter matrices of our new LDA method are calculated based on the class probabilities of the unlabeled samples. Then the loss function of CNN is modified by the scatter matrices.

In the new LDA method, we redefine the within-class mean vector $u_i$ and the total mean vector $u$:

$$\left.\begin{aligned} u_i &= \frac{\sum_{j=1}^{N} p_j^i x_j}{\sum_{j=1}^{N} p_j^i} = X\left(p_j^i / \sum_{j=1}^{N} p_j^i\right) = X\widetilde{p^i} \\ u &= \frac{\sum_{i=1}^{K}\sum_{j=1}^{N} p_j^i x_j}{\sum_{i=1}^{K}\sum_{j=1}^{N} p_j^i} = X\left(\sum_{i=1}^{K} p^i / \sum_{i=1}^{K}\sum_{j=1}^{N} p_j^i\right) = X\tilde{p} \end{aligned}\right\} \tag{14}$$

Compared with the standard LDA method, we use the class probabilities to calculate the mean vectors. The larger the probability is, the greater the impact on the mean vectors is. Since the information contained in the unlabeled samples is effectively utilized, the mean vectors are more reliable.

Afterward, we define the new scatter matrices:

$$\begin{aligned} S_b &= \sum_{i=1}^{K} m_i (u_i - u)(u_i - u)^T \\ &= X\left[\sum_{i=1}^{K} m_i \left(\widetilde{p^i} - \tilde{p}\right)\left(\widetilde{p^i} - \tilde{p}\right)^T\right] X^T \\ &= X\,\widetilde{S}_b\, X^T \end{aligned} \tag{15}$$

where $m_i = \sum_{j=1}^{N} p_j^i$,

$$\begin{aligned} S_w &= \sum_{i=1}^{k}\sum_{j=1}^{N} p_j^i \left(x_j - u_i\right)\left(x_j - u_i\right)^T \\ &= X\left[\sum_{i=1}^{k}\sum_{j=1}^{N} p_j^i \left(h_j^i - \widetilde{p^i}\right)\left(h_j^i - \widetilde{p^i}\right)^T\right] X^T \\ &= X\,\widetilde{S}_w\, X^T \end{aligned} \tag{16}$$

$$\begin{aligned} S_t &= \sum_{i=1}^{k}\sum_{j=1}^{N} p_j^i \left(x_j - u\right)\left(x_j - u\right)^T \\ &= X\left[\sum_{i=1}^{k}\sum_{j=1}^{N} p_j{}^i \left(h_j^i - \tilde{p}\right)\left(h_j^i - \tilde{p}\right)^T\right] X^T \\ &= X\,\widetilde{S}_t\, X^T \end{aligned} \tag{17}$$

where $h_j^i$ is expressed as follows:

$$h_j^i = \begin{cases} 1, i = j \\ 0, \text{else} \end{cases} \tag{18}$$

Compared with the standard LDA method, we redefine the $m_i$ in the between-class scatter matrix. In addition, the class probability $p_j^i$ is added as the weight coefficient in the within-class and total-class scatter matrices. The larger the class probability is, the greater the impact on the scatter matrices is. The new LDA method controls the impact of the unlabeled samples by the class probabilities. Thus, the reliability of the unlabeled samples is improved.

When constructing the generalized Rayleigh quotient optimization function, we can use any two scatter matrices, and one of the common ways is expressed in Eq. (19).

$$J = \frac{W^T S_w W}{W^T S_b W} \tag{19}$$

where $W = (w_1, w_2, \cdots, w_K)$ denotes the projection matrix. Since both the numerator and denominator of Eq. (19) are matrices, the optimization function cannot be

optimized as a scalar function. Therefore, an alternative optimization function is adopted:

$$J^* = \prod_{i=1}^{K} \frac{w_i^T S_b w_i}{w_i^T S_w w_i} \tag{20}$$

According to the nature of the generalized Rayleigh quotient, the minimum value of $J^*$ is the minimum eigenvalue of $S_w^{-1}S_b$. Afterward, the unsupervised component of the loss function for CNN is obtained, as shown in Eq. (21).

$$min\left(J^*\right) = \min\left[eig\left(S_w^{-1}S_b\right)\right] \tag{21}$$

Because of the simplicity and convergence rate of the cross-entropy function, we utilize it to construct the supervised component of the loss function for CNN, as shown in Eq. (22).

$$E_0 = -\frac{1}{N}\sum_x \sum_K y_k \ln a_k + (1-y_k)\ln(1-a_k) \tag{22}$$

where $(y_1, y_2, \cdots, y_K)$ represents the expected output of CNN and $(a_1, a_2, \cdots, a_K)$ denotes the actual output. Based on Eqs. (21) and (22), the loss function of CNN is the sum of the two components:

$$E = \left[-\frac{1}{N}\sum_x \sum_K y_k \ln a_k + (1-y_k)\ln(1-a_k)\right] + \min\left[eig\left(S_w^{-1}S_b\right)\right] \tag{23}$$

After the training process is finished, we use the well-trained CNN to predict the labels of the testing samples.

## Experiments

The experiments consist of two parts. We first discuss the effectiveness of the relevant steps in our method. Then we compare the performance of our method with other semi-supervised methods. The experiments are performed on the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset which contains multiple types of targets. In our experiments, we choose ten types of targets: 2S1, ZSU234, BRDM2, BTR60, BMP2, BTR70, D7, ZIL131, T62, and T72. Figure 3 shows the SAR and optical images of each type. Although the optical images are distinct from each other, the SAR images are difficult to be recognized because of the imaging nature. The dataset used in this paper consists of the training and testing datasets. The detailed information is listed in Table 1.

## Evaluation of our Method

### Evaluation of the New LDA Method

A new LDA method is designed to utilize the unlabeled samples. To verify the effectiveness of the new LDA method, we compare the overall accuracy and the Kappa score of our method with the supervised CNN method which only utilizes the labeled samples. The overall accuracy refers to the ratio of the number of correctly recognized samples to the number of all the samples. The calculation of Kappa score is based on the confusion matrix which can well represents the recognition accuracy of each class. The definition of Kappa score is shown in Eq. (24), where $p_o$ is the relative observed agreement between the recognition results of the testing data and the real labels and $p_e$ represents the hypothetical probability of the chance agreement.

$$k = \frac{p_o - p_e}{1 - p_e} \tag{24}$$

In the experiments, the training dataset is divided into a labeled dataset $L$ and an unlabeled dataset $U$. We randomly select the same number of the samples from each class in the training dataset and add them to $L$, while the remaining samples are added to $U$. We conduct a set of experiments under six different partitions, and the numbers of samples in $L$ and $U$ are shown in Table 2. We adopt Adam optimizer when training the CNN, and the parameters are set experimentally as follows: $\eta = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. When performing the thresholding processing on the class probabilities, the value of $t$ is set to 0.2. We repeat the experiments for ten times and the average results are shown in Table 3.

As can be seen, the overall accuracy and the Kappa score of our method are higher than the supervised CNN method. The fewer the labeled samples are, the more significant the difference of the performance is. This is because our method makes effective use of the unlabeled samples. As a result, the generalization ability of our method is enhanced and the overall accuracy and the Kappa score are improved. As the number of labeled samples increases, the generalization ability of the CNN model is augmented; hence, the performance difference between the two methods decreases.

Next, we use visual figures to illustrate the effectiveness of the new LDA method. We extract the feature vectors of the testing samples outputted by our model and the supervised CNN model. Then we transform the feature vectors to two-dimensional ones using the t-Distributed Stochastic Neighbor Embedding (t-SNE) method. In this experiment, we adopt four different partitions in Table 2 to train the two models.


**Fig. 3** The SAR and optical images of ten types of targets in the MSTAR dataset

The distribution of the feature vectors outputted by our model and the supervised CNN model are shown in Fig. 4. Different colors represent different classes. It can be seen that compared with the supervised CNN method, our method can effectively reduce the within-class distance and increase the between-class distance. This means that the recognition accuracy of our method is improved, which is consistent with the experimental results shown in Table 3.

### Evaluation of the Thresholding Processing

After the class probabilities have been obtained, thresholding processing is applied to improving the reliability of the unlabeled samples. We will discuss the effectiveness of the thresholding processing in this section. We train our model under two different partitions in Table 2, and the number of the samples in $U$ is 2347 and 1947, respectively. In this experiment, we utilize the softmax function to calculate the class probabilities of the unlabeled samples. Then the RF value of the unlabeled samples is obtained based on the class probabilities and the real labels. We regard the samples with the RF value greater than 0.9 as reliable samples, and the remaining samples are regarded as unreliable samples. In the thresholding processing, we set the threshold $t$ as 0, 0.2, 0.7, and 1, respectively. During the experiment, we record the numbers of the reliable samples, unreliable samples, and available samples in $U$. The experimental results are shown in Table 4.

Compared with the case of $t = 0$, the number of the unreliable samples is reduced when $t = 0.2$, and the number of the reliable samples is increased. Hence, the thresholding

**Table 1** The training and testing datasets in our experiments

| Type | Tops | Model | Training set | | Testing set | |
|---|---|---|---|---|---|---|
| | | | Depression | Number | Depression | Number |
| 2S1 | Artillery | B_01 | 17° | 299 | 15° | 274 |
| ZSU234 | | D_08 | 17° | 299 | 15° | 274 |
| BRDM2 | Truck | E_71 | 17° | 298 | 15° | 274 |
| BTR60 | | K10YT_7532 | 17° | 256 | 15° | 195 |
| BMP2 | | SN_9563 | 17° | 233 | 15° | 195 |
| BTR70 | | C_71 | 17° | 233 | 15° | 196 |
| D7 | | 92V_13015 | 17° | 299 | 15° | 274 |
| ZIL131 | | E_12 | 17° | 299 | 15° | 274 |
| T62 | Tank | A_51 | 17° | 299 | 15° | 273 |
| T72 | | #A64 | 17° | 232 | 15° | 196 |
| | | | Sum:2747 | | Sum:2425 | |

**Table 2** Numbers of samples in $L$ and $U$ under different partitions of the training dataset

| | Number of $L$ | Number of $U$ |
|---|---|---|
| 1 | 300 | 2447 |
| 2 | 400 | 2347 |
| 3 | 500 | 2247 |
| 4 | 600 | 2147 |
| 5 | 800 | 1947 |
| 6 | 1000 | 1747 |

processing can effectively improve the reliability of the unlabeled samples. However, if the threshold continues to increase, the number of the available samples will be reduced. That is because if the maximum probability of a sample is less than the threshold, all the probabilities of the sample will be set to 0. As shown in Fig. 1, the unlabeled samples are utilized based on the class probabilities and the new LDA method. The unlabeled samples whose class probabilities are all set to 0 will not be utilized in the training process. Therefore, as the threshold increases, the number of the available unlabeled samples drops.

Next, we will analyze the performance of our method with different thresholds. The experimental results are shown in Fig. 5. When the number of the labeled samples is less than 500, the performance of our method is better while the threshold is set to 0.2. As the number of the labeled samples increases, the recognition accuracy is almost the same with different thresholds. This is because the generalization ability of the CNN model is weak when the labeled samples are insufficient. Compared with the case of $t = 0$, setting $t = 0.2$ helps to improve the reliability of the unlabeled samples. Thus, the recognition accuracy is improved. However, if we continue to increase the threshold, the number of the available unlabeled samples drops and the recognition accuracy becomes worse. As the number of the labeled samples increases, the generalization ability of the CNN model is improved. As a result, the reliability of the unlabeled samples is improved, and the impact of the thresholding processing becomes weaker. Thus, in order to achieve the best recognition performance, the threshold should be set to 0.2.

## Comparison With Other Semi-Supervised Methods

In this section, we compare the performance of our method with semi-supervised ladder network model [29], Pi model, and temporal ensembling model [30]. The semi-supervised ladder network model combines the semi-supervised learning with deep learning methods. Based on the self-ensembling method, the Pi and temporal ensembling models are both semi-supervised CNN methods.

### Recognition Accuracy

In this section, we compare the overall accuracy of these methods. As shown in Fig. 6, our method outperforms the semi-supervised ladder network. The reason is that the ladder network is composed of fully connected layers whose feature extraction ability is weaker than the CNN model. Our method is also superior to the other two models. This is because the Pi and temporal ensembling models use the CNN to generate the pseudo

**Table 3** The performance of the supervised CNN method and our method under different partitions of the training dataset

| Training set | $L$:300, $U$:2447 | | $L$:400, $U$:2347 | | $L$:500, $U$:2247 | | $L$:600, $U$:2147 | | $L$:800, $U$:1947 | | $L$:1000, $U$:1747 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | CNN | Ours | CNN | Ours | CNN | Ours | CNN | Ours | CNN | Ours | CNN | Ours |
| 2S1 | 0.70 | 0.73 | 0.68 | 0.79 | 0.77 | 0.84 | 0.83 | 0.87 | 0.84 | 0.89 | 0.91 | 0.95 |
| BMP2 | 0.64 | 0.69 | 0.74 | 0.85 | 0.81 | 0.95 | 0.88 | 0.92 | 0.89 | 0.95 | 0.91 | 0.97 |
| BRDM2 | 0.63 | 0.77 | 0.77 | 0.86 | 0.84 | 0.88 | 0.86 | 0.90 | 0.91 | 0.92 | 0.86 | 0.95 |
| BTR70 | 0.58 | 0.74 | 0.76 | 0.89 | 0.79 | 0.89 | 0.81 | 0.90 | 0.89 | 0.94 | 0.88 | 0.96 |
| BTR60 | 0.64 | 0.67 | 0.81 | 0.88 | 0.79 | 0.86 | 0.86 | 0.87 | 0.90 | 0.93 | 0.91 | 0.95 |
| D7 | 0.88 | 0.89 | 0.91 | 0.92 | 0.96 | 0.97 | 0.96 | 0.96 | 0.97 | 0.98 | 0.99 | 0.98 |
| T62 | 0.64 | 0.73 | 0.76 | 0.83 | 0.79 | 0.81 | 0.87 | 0.87 | 0.86 | 0.89 | 0.91 | 0.94 |
| T72 | 0.55 | 0.64 | 0.70 | 0.82 | 0.80 | 0.87 | 0.83 | 0.87 | 0.84 | 0.92 | 0.89 | 0.94 |
| ZIL131 | 0.59 | 0.71 | 0.75 | 0.87 | 0.77 | 0.86 | 0.74 | 0.86 | 0.83 | 0.90 | 0.88 | 0.92 |
| ZSU234 | 0.75 | 0.81 | 0.86 | 0.90 | 0.91 | 0.90 | 0.91 | 0.93 | 0.94 | 0.96 | 0.96 | 0.97 |
| Overall accuracy | 0.67 | **0.74** | 0.78 | **0.86** | 0.83 | **0.88** | 0.85 | **0.90** | 0.89 | **0.93** | 0.91 | **0.95** |
| Kappa score | 0.63 | **0.71** | 0.75 | **0.85** | 0.81 | **0.87** | 0.84 | **0.88** | 0.87 | **0.92** | 0.90 | **0.95** |

The supervised CNN method only utilizes the labeled samples while our method utilizes both the labeled and unlabeled samples. The better overall accuracies and Kappa scores between the two methods are indicated in bold

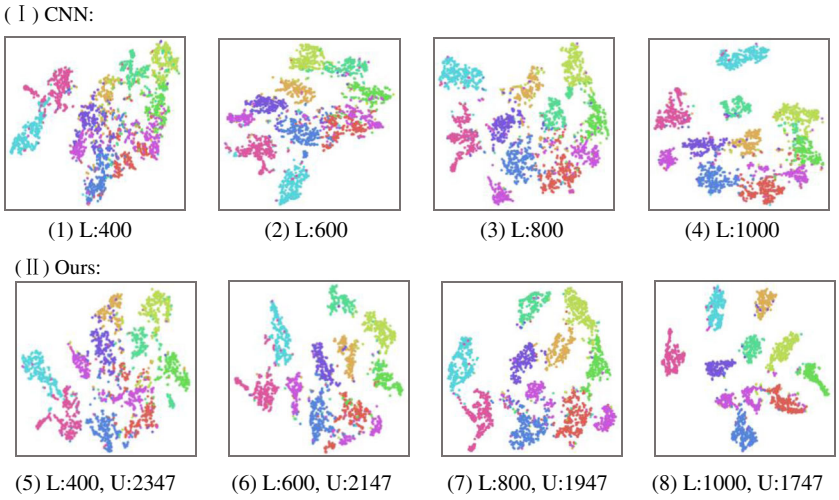


**Fig. 4** The distribution of the feature vectors outputted by our model and the supervised CNN model. The first row represents the supervised CNN model's output and the second row represents our model's output. Different colors represent different classes

labels of the unlabeled samples. However, the recognition accuracy will be reduced if the pseudo labels are incorrect. In contrast, our method can accurately estimate the class probabilities of the unlabeled samples. Based on the class probabilities, the information contained in the unlabeled samples is well utilized. As a result, the recognition accuracy is increased.

### Training Time

To evaluate the computation complexity of our method and the other three semi-supervised methods, we calculate the average training time. We use 600 labeled samples and 2147 unlabeled samples to train the four methods. The number of the epochs is set to 400. The experiments are implemented with the Pytorch 0.3.1 framework. And the main configurations of the computer are GPU: Tesla K20c, video memory: 4G, operating system: Ubuntu 16.04.

As shown in Table 5, the average training time of our method is 2. 53 s/epoch, much less than the semi-supervised ladder network. The reason is that the structure of the ladder network is more complex than the CNN used in our method. Thus, there are more parameters that need to be trained in the ladder network, resulting in longer training time. Besides, the average training time of the Pi and temporal ensembling models is less than our method. This is because the Pi and temporal ensembling models utilize the CNN to generate the pseudo labels of the unlabeled samples. Afterward, the unsupervised component of the loss function is obtained based on the pseudo labels. Thus, the computation complexity of the two methods is less than our method. However, our method can effectively maintain the reliability of the unlabeled samples. Although the computation complexity of our method is increased, the recognition accuracy is also improved.

## Conclusion

Inspired by the neural network structure and the semi-supervised learning mechanism of the human cognitive system, a new semi-supervised CNN method is presented in this paper. In the training process, the information contained in the unlabeled samples is integrated into the loss function of CNN relying on a new LDA method. Specifically, we utilize the CNN to obtain the class probabilities of the unlabeled samples and then adopt the thresholding processing to optimize the class probabilities. The experimental results on the MSTAR dataset

**Table 4** The numbers of the reliable samples, unreliable samples, and available samples in the unlabeled dataset with different thresholds

| Training set | $L$:400, $U$:2347 | | | | $L$:800, $U$:1947 | | | |
|---|---|---|---|---|---|---|---|---|
| Threshold | $t=0$ | $t=0.2$ | $t=0.7$ | $t=1$ | $t=0$ | $t=0.2$ | $t=0.7$ | $t=1$ |
| Unreliable samples | 329 | 259 | 198 | 20 | 125 | 112 | 80 | 8 |
| Reliable samples | 2018 | 2088 | 2034 | 891 | 1822 | 1835 | 1841 | 1646 |
| Available samples | 2347 | 2347 | 2232 | 911 | 1947 | 1947 | 1921 | 1654 |

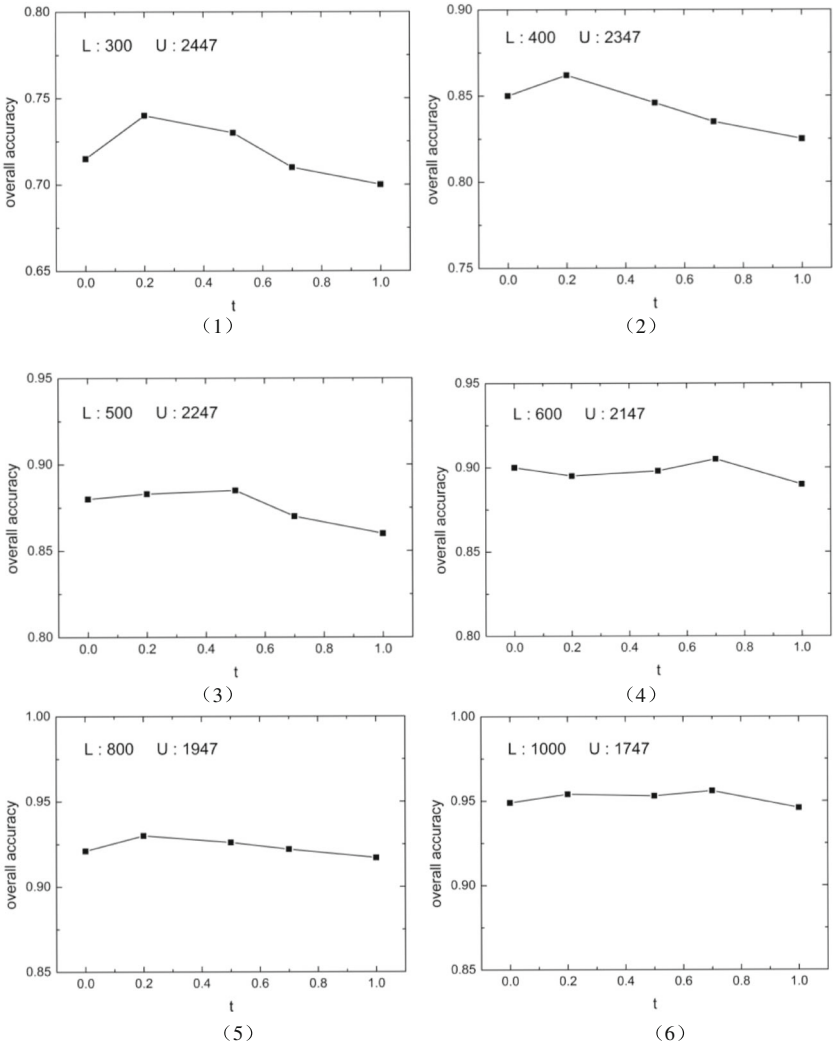


**Fig. 5** The recognition accuracy of our method with different thresholds and different training dataset partitions

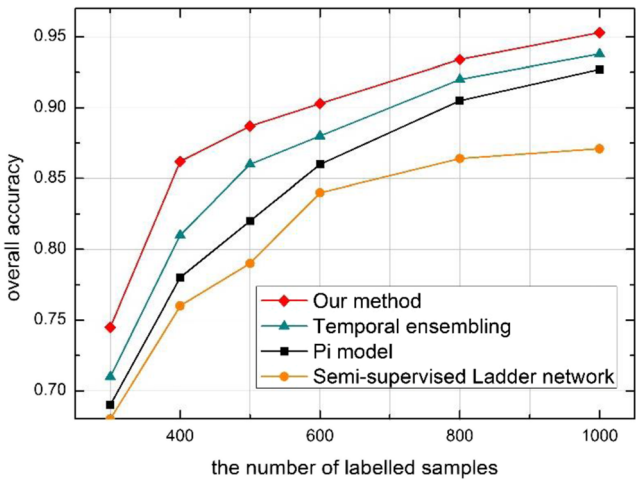


**Fig. 6** The recognition accuracy of our method, temporal ensembling model, Pi model, and semi-supervised ladder network with different partitions of the training dataset

demonstrate that the thresholding processing can improve the reliability of the unlabeled samples. Based on the optimized class probabilities, the scatter matrices of the new LDA method are designed to introduce the unlabeled samples in the loss function of CNN. The distribution of the feature vectors verifies that the new LDA method can reduce the within-class distance and

**Table 5** The average training time of our method, temporal method, Pi model, and semi-supervised ladder network. All the four methods are trained with 600 labeled samples and 2147 unlabeled samples

| Methods | Training time (s/epoch) |
|---|---|
| Our method | 2.53 |
| Temporal ensembling | 1.56 |
| Pi model | 2.45 |
| Semi-supervised ladder network | 5.08 |

increase the between-class distance. As a result, our method can effectively improve the SAR ATR accuracy when the labeled samples are insufficient and outperforms other semi-supervised methods.

**Funding Information** This research was funded by the National Natural Science Foundation of China (No. 61771027, No. 61071139, No. 61471019, No. 61501011, and No. 61171122). E. Yang is supported in part under the RSE-NNSFC Joint Project (2017–2019) (No. 6161101383) with China University of Petroleum (Huadong). H. Zhou was supported by UK EPSRC under Grant EP/N011074/1, Royal Society-Newton Advanced Fellowship under Grant NA160342, and European Union's Horizon 2020 research and innovation program under the Marie-Sklodowska-Curie grant agreement No 720325.

## Compliance with Ethical Standards

**Conflict of Interest** The authors declare that they have no conflict of interest.

**Ethical Approval** This article does not contain any studies with human participants performed by any of the authors.

## References

1. Gao F,Yang Y, Wang J, Sun J, Yang E, Zhou H. A Deep Convolutional Generative Adversarial Networks (DCGANs)-Based Semi-Supervised Method for Object Recognition in Synthetic Aperture Radar (SAR) Images. Remote Sens. 2018;10(6):846.
2. Wang G, Tan S, Guan C, Wang N, Liu Z. Multiple model particle filter track-before-detect for range ambiguous radar. Chin J Aeronaut. 2013;26:1477–87.
3. Ma F, Gao F, Sun J, Zhou H, Hussain A. Weakly Supervised Segmentation of SAR Imagery Using Superpixel and Hierarchically Adversarial CRF. Remote Sens. 2019;11(5):512.
4. Chen H, Zhang F, Tang B, Yin Q, Sun X. Slim and efficient neural network design for resource-constrained SAR target recognition. Remote Sens. 2018;10(10):1–15.
5. Zhang F, Yao X, Tang H,Yin Q, Hu Y, Lei B. Multiple Mode SAR Raw Data Simulation and Parallel Acceleration for Gaofen-3 Mission. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 2018:1-12.
6. Garagnani M, Wennekers T, Pulvermüller F. Recruitment and consolidation of cell assemblies for words by way of hebbian learning and competition in a multi-layer neural network. Cogn Comput. 2009;1(2):160–76.
7. Zhang S, He B, Rui N, Wang J, Han B, Lendasse A. Fast image recognition based on independent component analysis and extreme learning machine. Cogn Comput. 2014;6(3):405–22.
8. Gao F, Ma F, Wang J, Sun J, Yang E, Zhou H. Visual saliency modeling for river detection in high-resolution SAR imagery. IEEE Access. 2018; 6:1000-14.
9. Gao F, Ma F, Zhang Y, Wang J, Sun J, Yang E, et al. Biologically inspired progressive enhancement target detection from heavy cluttered SAR images. Cogn Comput. 2016;8(5):955–66.
10. Spratling M. A hierarchical predictive coding model of object recognition in natural images. Cogn Comput. 2017;9(2):151–67.
11. Ren P, Sun W, Luo C, Hussain A. Clustering-oriented multiple convolutional neural networks for single image super-resolution. Cogn Comput. 2018;10(1):165–78.
12. Chen Y, Jiang H, Li C, Jia X, Ghamisi P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans Geosci Remote Sens. 2016;54(10): 6232–51.
13. Amrani M, Jiang F. Deep feature extraction and combination for synthetic aperture radar target classification. J Appl Remote Sens. 2017;11(4):1.
14. Zhao J, Guo W, Cui S, Zhang Z, Yu W. Convolutional neural network for SAR image classification at patch level. Geoscience and Remote Sensing Symposium IEEE. 2016:945–8.
15. Chen S, Wang H. SAR target recognition based on deep learning. International Conference on Data Science and Advanced Analytics IEEE 2015; pp 541–7
16. Gao F, Huang T, Sun J, Wang J, Hussain A, Yang E. A new algorithm of SAR image target recognition based on improved deep convolutional neural network. Cogn Comput. 2018:1–16.
17. Liu B, Yu X, Zhang P, Tan X, Yu A, Xue Z. A semi-supervised convolutional neural network for hyperspectral image classification. Remote Sens Lett. 2017;8(9):839–48.
18. Fu Z, Zhang F, Yin Q, Li R, Hu W, Li W. Small sample learning optimization for ResNet based SAR target recognition. International Geoscience and Remote Sensing Symposium IEEE. 2018:2330–3.
19. Zhu X, Rogers T, Qian R, Kalish C. Humans perform semi-supervised classification too. National Conference on artificial intelligence AAAI Press 2007;864–9.
20. Haeusser P, Mordvintsev A, Cremers D. Learning by association—a versatile semi-supervised training method for neural networks. In Proceedings of IEEE conference on computer vision and pattern recognition 2017; pp. 626–35.
21. Gibson B, Rogers T, Zhu X. Human semi-supervised learning. Top Cogn Sci. 2013;5(1):132–72.
22. Hänsch R, Hellwich O. Semi-supervised learning for classification of polarimetric SAR-data. Geoscience and Remote Sensing Symposium IEEE. 2010:987–90.
23. Uhlmann S, Kiranyaz S, Gabbouj M. Semi-supervised learning for ill-posed polarimetric SAR classification. Remote Sens. 2014;6(6): 4801–30.
24. Basu S. Semi-supervised learning. Pacific-Asia conference on advances in knowledge discovery and data mining 2010;588–95.
25. Leng Y, Xu X, Qi G. Combining active learning and semi-supervised learning to construct SVM classifier. Knowl-Based Syst. 2013;44:121–31.
26. Zhang X, Song Q, Liu R, Wang W, Jiao L. Modified co-training with spectral and spatial views for semisupervised hyperspectral image classification. IEEE J Sel Top Appl Earth Observ Remote Sens. 2014;7(6):2044–55.
27. Lv L, Zhao D, Deng Q. A semi-supervised predictive sparse decomposition based on task-driven dictionary learning. Cogn Comput. 2016;9(1):1–10.
28. Ding S, Xi X, Liu Z, Qiao H, Zhang B. A novel manifold regularized online semi-supervised learning model. Cogn Comput. 2018;10(1):49–61.
29. Rasmus A, Valpola H, Honkala M, Berglund M, Raiko T. Semi-supervised learning with ladder networks. Comput Sci. 2015;9(Suppl 1):1–9.
30. Laine S, Aila T. Temporal ensembling for semi-supervised learning. International conference on learning representations 2017.

31. Le T, Kim S. A hybrid selection method of helpful unlabeled data applicable for semi-supervised learning algorithms. The IEEE International Symposium on Consumer Electronics. 2014:1–2.
32. Li Y, Zhou Z. Towards making unlabeled data never hurt. IEEE Trans Pattern Anal Mach Intell. 2015;37(1):175–88.
33. Huang Y, Wang S, Ren S. Pinning exponential synchronization and passivity of coupled delayed reaction-diffusion neural networks with and without parametric uncertainties. Int. J. Control. 2017:1-35.
34. Huang Y, Qiu S, Ren S. Finite-time synchronization and passivity of coupled memristive neural networks. Int. J. Control. 2019.
35. Moskowitz L. The LDA—an integrated diagnostics tool. IEEE Aerosp Electron Syst Mag. 1986;1(7):22–6.
36. Huan R, Liang R, Pan Y. SAR target recognition with the fusion of LDA and ICA. International Conference on Information Engineering and Computer Science IEEE. 2009:1–5.