A Novel Separability Based Objective Function of CNN for Extracting Linearly separable Features from SAR Image Target

ABSTRACT: Convolutional Neural Network (CNN) has become a promising method for Synthetic Aperture Radar (SAR) target recognition. Existing CNN models aim at seeking the best division between classes, but rarely care about the separability of them. In this paper, a novel objective function is designed to maximize the separability between classes. By analyzing the property of linear separability, we perform a separability measure and construct an objective function. Finally, we train a CNN to extract linearly separable features. The experimental results indicate the output features are linearly separable, and the classification results are comparable with the state-of-the-art.

Keywords: Synthetic Aperture Radar, Convolution Neural Network, Classification, Linear Separability, Softmax

I. Introduction

Synthetic Aperture Radar (SAR) [1] is an important means of remote sensing in many fields. Because it provides all-time and all-weather imaging capability. Therefore, SAR images target detection[2] and recognition technologies have become very important. However, it is difficult to recognize targets on SAR images.

Recently, Convolutional Neural Network (CNN) has become an increasingly import tool in image target classification. Since CNN was reported in [3] , it has made great success in both theory and practice of image target recognition. For example, in 2015, the ResNet [4] won the championship in the contest of ImageNet Large Scale Visual Recognition Challenge (ILSVRC), obtained a 3.75% error rate which was even unprecedentedly better than humans. The achievements of CNN on image processing are credited to its ability to quickly extract two-dimensional features despite the shift, scaling, and rotation of image targets. Besides, CNN does not need to focus on the imaging form and interpretation principle of SAR. These capacities make CNN applicable on SAR automatic target recognition (SAR-ATR) tasks. Many researchers have introduced CNN to SAR target recognition. Razavian et al. [5] added the mounting evidence that the generic descriptors extracted from the CNN are very powerful. They reported a series of experiments conducted for different recognition tasks using CNN. The results suggested that features obtained CNN should be the primary candidate in most visual recognition tasks. Chen et al. [6] attempted to adapt the optical camera-oriented CNN to its microwave counterpart, such as SAR. Experiments on the MSTAR (the abbreviation of Moving and Stationary Target Acquisition and Recognition) dataset [7] showed that the accuracy of 90.1% was achievable on three types of target classification tasks, and 84.7% on ten types. Li et al. [8] proposed a fast training method for CNN using convolutional auto-encoder and shallow neural network. The results of the experiments on the MSATR database showed that their method tremendously reduced the training time with very little loss of recognition rates. Ding et al. [9] investigated the capability of a deep CNN combined with three types of data augmentation operations in SAR target recognition. The experimental results showed that CNN was a practical approach for target recognition in challenging conditions of target translation, random speckle noise, and missing poses.

The training of CNN is actually an iterative process to minimizing or maximizing a predefined objective function using back propagation and gradient descent methods. For classification problems, there are many objective functions available, such as Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Logarithmic Error (MSLE), Binary Cross Entropy (BCE), and Categorical Cross Entropy (CCE) [10]. Among them the CCE combined with the widely used Softmax classifier performs remarkably well in multi-classification tasks [11].

The major problem in applying CNN to SAR target classification is the shortage of the training data. It is difficult for people to collect a large number of high-quality SAR images. Therefore, there are no other available data sets except MSTAR, published by United States Department of Defense Advanced Research Program. There are thousands of samples in MSTAR. The shortage of data leads to overfitting of the networks. As a result, how to achieve better generalization performance with a small number of samples is also overwhelming.

Nevertheless, existing classifiers mainly focus on the accuracy of the classification and all the objective functions mentioned above just aim at seeking a best division between classes, but never care about the separability of the categories. The benefits of using linearly separable features are obvious. It helps us to process the linearly separable data, such as classification and clustering. In this paper, we try to find a linear separability measure between classes, and build an objective function for CNN to output features with linear separability. The Principal Component Analysis (PCA), which is presented as a data dimension reduction method, prompts us to use a covariance matrix to measure linear separability. [12, 13]

The remainder of this paper is organized as follows. Section II shows related work, which mainly contains the details of CNN and PCA. Section III introduces the proposed method, including the analysis and computation procedure of linear separability and the construction of the proposed objective function. Experimental results are presented in Section III, where we present the experiment dataset and the analysis of the results. Section IV concludes our work.

II. Related work

1. Convolutional Neural Network [3]

Convolutional Neural Network (CNN) is a kind of Deep Learning Network. It generally contains an input layer, at least one 2-dimensional convolutional layer, several polling layers, dense layers and an output layer. Each convolutional layer processes the output of the previous layer with several convolution kernels by convocation operation. The outputs of the convolution layer are always transferred to a pooling layer to shrink the feature size. The benefit of taking these operations is the output feature will not change regardless of shift, scaling and rotation.

Each neuron of CNN has weights for each input, $w_1, w_2, \ldots$, and an overall bias, b. And the output is $\sigma(w \cdot x + b)$, where σ is called the activate function. We can present the outputs of the l^{th} layer as:

$$\begin{cases} \mathbf{z}^l = W^l \cdot \mathbf{x}^{l-1} + \mathbf{b}^l \\ \quad \mathbf{a}^l = \sigma(\mathbf{z}^l) \end{cases} \tag{1}$$

where W^l is a matrix of all the weights, $\mathbf{b}^l$ is the bias vector, $\mathbf{x}^{l-1}$ is the input as well as the output of the former layer, and $\mathbf{a}^l$ is the output of the l^{th} layer. Assume the L^{th} layer is the last one, $\mathbf{a}^L$ represents the final output. The aim of training the network is to derive the best W^l and $\mathbf{b}^l$.

The Back Propagation (BP) [14] algorithm iteratively computes W^l and $\mathbf{b}^l$ by defining an objective function $C(\mathbf{a}^L)$ and following the Gradient Descent (GD) principle.

The goal of backpropagation is to compute the partial derivatives of the objective function $C(\mathbf{a}^L)$ with respect to any weight w or bias b in the network, denoted as $\partial C / \partial w$ and $\partial C / \partial b$. Firstly, BP computes the error in the output layer. The error of the j^{th} neuron in the last layer is defined as:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L) \tag{2}$$

where δ_j^l denotes the error of neuron j in layer l , where $\delta_j^l \equiv \partial C / \partial z_j^l$. Obviously, the above expression can be presented in a matrix-based form:

$$\boldsymbol{\delta}^L = \nabla_{\mathbf{a}^L} C \,\square\, \sigma'(\mathbf{z}^L) \tag{3}$$

where the vector $\nabla_{\mathbf{a}^L} C$ is defined to contain all the layer L's partial derivatives $\partial C / \partial a_j^L$.

Then, express the error $\boldsymbol{\delta}^l$ by the next layer's error $\boldsymbol{\delta}^{l+1}$:

$$\boldsymbol{\delta}^l = ((W^{l+1})^T \boldsymbol{\delta}^{l+1}) \,\square\, o'(\mathbf{z}^l) \tag{4}$$

where the operation $\square$ is called the Hadamard product, meaning the elementwise product of the two vectors have the same dimension. Here, applying the transpose weight matrix $(W^{l+1})^T$ means moving the error backward through the network. And the Hadamard product with $\sigma'(\mathbf{z}^l)$ means moving the error backward through the activation function at layer l .

After that, $\partial C / \partial w$ and $\partial C / \partial b$ can be obtained:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \qquad \frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l \tag{5}$$

Then a small step is added on W and B iteratively :

$$\begin{aligned} W' &= W - lr \cdot \partial C / \partial w \\ B' &= B - lr \cdot \partial C / \partial b \end{aligned} \tag{6}$$

Therefore, C will get smaller and smaller until get the minimal value, when the training complete.

Through the backpropagation process we can find that the objective function plays a vital role in the training of CNN. Currently, most of the objective functions are defined directly to the final goal, such as getting the best classification result. For example, the MSE objective function is defined as

$$C = \frac{1}{N}\sum (y_{pred} - y_{true})^2 \tag{7}$$

where y_{pred} is the output of the network and y_{true} is the label of the input samples. Hence the more C gets closer to 0, the more similar y_{pred} and y_{true} are.

As another example, the widely used CCE objective function, which calculates the cross entropy between classes, is defined as:

$$C = -\frac{1}{N}\sum\sum[y_{true}\ln y_{pred} + (1-y_{true})\ln(1-y_{pred})] \tag{8}$$

where the negative sign denotes maximizing the cross entropy.

The minimum MSE and maximum CCE, as well as criteria of other objective functions mentioned above, are used to demonstrate classification results. That makes the output of CNN be the classification results.

2. Covariance Matrix and PCA

Before talking about the separability measure, we introduce some related properties of covariance matrix. In probability theory, A covariance matrix is a matrix whose element in $(i,\ j)$ position is the covariance between the i^{th} and j^{th} elements of a random vector.

For a set of data $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \mathbf{x}_i \in \mathbb{R}^{n\times 1}$. The covariance matrix of X is:

$$S = \operatorname{cov}(X) = E[(X - E[X])(X - E[X])^T] \tag{9}$$

where $E[*]$ is the mathematical expectation of *: $E[X] = (\mathbf{x}_1 + \mathbf{x}_2 +, \ldots, + \mathbf{x}_N)/N$.

The covariance matrix S is a Hermitian Conjugate Matrix. The Singular Value Decomposition (SVD) of S will be:

$$S = Q^T \Sigma Q \text{ , where } Q = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \ldots \\ \mathbf{q}_n \end{bmatrix}, \quad \Sigma = \mathrm{dlag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \tag{10}$$

where Q is an orthogonal matrix, $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ are the eigenvectors of S . Σ is a diagonal matrix, and $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of S . $\mathbf{q}_\mathbf{i}$ is the eigenvector corresponding to eigenvalue λ_i .

Then we have

$$\mathbf{q}_i S = \lambda_i \mathbf{q}_i \tag{11}$$

According to the property of covariance matrix, there is:

$$\mathrm{var}(\mathbf{q}_i X) = \mathrm{cov}(\mathbf{q}_i X) = \mathbf{q}_i S \mathbf{q}_i{}^T = \lambda_i \tag{12}$$

where $\mathrm{var}(\mathbf{q}_i X)$ is the variance of vector $\mathbf{q}_i X$.

This means the linear transformation $\mathbf{q}_i$ maps X to a single dimensional vector $\mathbf{q}_i X$ whose variance equals λ_i . Therefore, the larger λ_i is, the more variable $\mathbf{q}_i X$ is. The variance of $\mathbf{q}_i X$ indicates the variance of X on the direction of $\mathbf{q}_i$. And because of this, in Principal Component Analysis (PCA), $\mathbf{q}_i X$ which corresponds to the largest λ_i is defined as the most important principal component of X . And λ_i measures the importance of each component. [12]

The variance of X on the direction of $\mathbf{q}_i$ also measures the scatter condition on the same direction. And for different classes, the more scattered they distribute, the more separable they are.

III. Proposed Method

Figure 1 shows the basic structure and training process of the traditional CNN. The output is used as the input of the loss function, and the network is trained by backpropagation. There is a Softmax activation function on the last layer of the CNN. The training process uses CCE as the objective function.

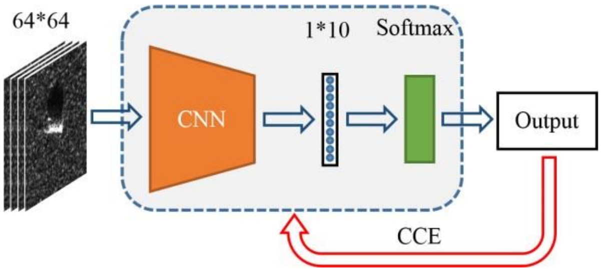


Figure 1 The training process of traditional CNN.

However, the purpose of the objective function proposed here is to make sure the features with the most separability characteristics. The structure of the whole network is shown in Figure 2. The dimension of the extracted feature $m > 9$, which is related to the number of the categories. For classification tasks, we can train an extra linear classifier, such as a linear SVM [15].

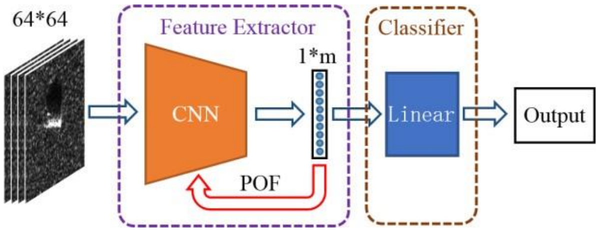


Figure 2 The training process of proposed method using proposed objective function (POF).

So, in the next two subsections, we will deal with two main problems: the dimension of the output of CNN, and to construct the objective function.

1. Linear separability

In Euclidean geometry, linear separability is a geometric property of a pair of sets of points. For two dimensional situations, these two sets are linearly separable if there exists at least one line in the plane to separate them. This property can be generalized to higher-dimensional spaces if the line is replaced by a hyperplane.

In the case of multi-classification, there are two main classification strategies for linear classification, One-Vs-Rest and Pairwise [16]. The former can be described as that, for any class, there exists a hyperplane that can separate it from the others. The latter is that, for any two classes, there exists a hyperplane that can separate them from each other.

In this section, we will try to obtain a measure of separability.

(1) Dispersion between classes

Assume that there are N samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}, \mathbf{x} \in \mathbb{R}^{m\times 1}$. Where N_i samples belong to class $c_i, i = 1, 2, ..., C$.

(a) Binary classification

A simple case in classification problems is binary classification, where C=2. Ignoring the distribution of the data itself, to separate the two classes, we can simply choose the mean of each class as follows:

$$\boldsymbol{\mu}_i = \frac{1}{N_i}\sum_{\mathbf{x}\in c_i}\mathbf{x} \tag{13}$$

The distance between the two mean values $d = |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|$ is thought as a measurement of separability.

Let $X = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2\}$, the covariance matrix is:

$$S = \operatorname{cov}(X) = \frac{1}{2}\left[(\boldsymbol{\mu}_1 - \boldsymbol{\mu})(\boldsymbol{\mu}_1 - \boldsymbol{\mu})^T + (\boldsymbol{\mu}_2 - \boldsymbol{\mu})(\boldsymbol{\mu}_2 - \boldsymbol{\mu})^T\right] \tag{14}$$

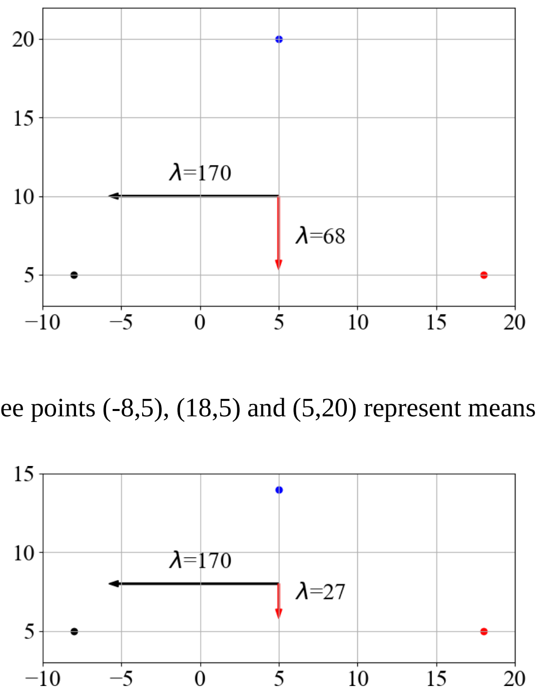
where $\bar{\boldsymbol{\mu}} = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) / 2$. S is a matrix with shape of $m \times m$, and it has m eigenvalues. S has only one nonzero eigenvalue $\lambda = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)/4$, and its corresponding eigenvector is $\mathbf{v} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$. Obviously, $\lambda \propto d^2$, and the two coordinates $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are scattered in the same direction of $\mathbf{v}$. So λ can be a separability measure.

If the number of the classes is greater than two, it is relatively complex to evaluate the separability.

(b) Ternary classification

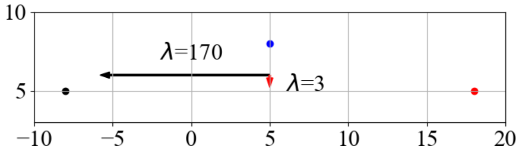
When C=3, the three mean vectors are $\{\boldsymbol{\mu}_i, i = 1, 2,3\}$. Let $X = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3\}$, Figure 3 shows the eigenvalues and eigenvectors of $S = \mathrm{cov}(X)$. The arrows represent the eigenvectors of S. Each arrow has the length of the corresponding eigenvalue.

Experiments show that when one point moves to the other two points, the eigenvalues and the direction of the eigenvectors will change. The more collinear the three means become, the smaller the minimum eigenvalue λ_s is. When three points become inseparable, λ_s reduces to zero. On the contrary, only when λ_s is large enough, the three means are non-collinear. So λ_s can be a measurement of the total separability between the three classes.



(a) There are three points (-8,5), (18,5) and (5,20) represent means of three classes.

(b) When one point becomes closer to the other two, from (5,20) to (5,14), the smaller λ reduces to 27.



(c) When the same point moves to (5,8), the smaller λ reduces to 3.

Figure 3 In each figure, we denote the three mean vectors by dots. The two arrows are the eigenvectors of the covariance matrix. Each eigenvector has the length of the corresponding eigenvalue. When separability gets worst, the smallest eigenvalue correspondingly gets smaller.

However, if the data samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, $\mathbf{x} \in \mathbb{R}^{m\times1}$, $m > 2$, then $S \in \mathbb{R}^{m\times m}$ has m eigenvalues. Therefore, the largest λ_l and its corresponding eigenvectors $\mathbf{v}_l$ denote that in the direction of $\mathbf{v}_l$ classes are the farthest from the others. And the second large λ_{2ed} and $\mathbf{v}_{2ed}$ denote that in the direction of $\mathbf{v}_{2ed}$ classes are the second far from the others. There is no other direction for three classes to analyze separability. So λ_{2ed} can be the measure of the separability of the three classes.

As a generalization, we conclude that the separability of C class can be represented by the C-1th eigenvalue λ_{C-1th} of the covariance matrix $\text{cov}\left(\left[\boldsymbol{\mu}_1 - \bar{\boldsymbol{\mu}}, \boldsymbol{\mu}_2 - \bar{\boldsymbol{\mu}}, ..., \boldsymbol{\mu}_C - \bar{\boldsymbol{\mu}}\right]\right)$. In multi-classification tasks, this conclude is useful. $\lambda_{C-1th} \neq 0$ can guarantee that every three classes are not collinear, and every four classes do not coplanar, and so on. So we can extend separability as arbitrary number of classes can be linearly separated from the others.

An implicit condition is that C classes are not separable if the dimension of the data samples is smaller than C-1. So, the output dimension of our CNN must be bigger than C-1.

(2) Intra class scatter

Fisher Linear Discriminant (FLD) shows that two classes can be discriminated considering both the distribution of intra and between two classes [17]. FLD defines a within-scatter s_w and a between-scatter s_b. s_b denotes the distance between the two classes, and s_w measures the average scatter degree in each class. FLD defines a separability measure by normalizing s_b with s_w:

$$J = \frac{s_b}{s_w} \tag{15}$$

The calculation details of s_b with s_w are as follows. Let $\mathbf{w} \in \mathbb{R}^{m\times1}$ be a liner mapping from the feature space $\mathbb{R}^{m\times1}$ to $F^{1\times1}$. Then $y = \mathbf{w}^T\mathbf{x}$. The mean vector of each class is

$$\mu_i = \frac{1}{N_i}\sum_{y\in c_i} y \tag{16}$$

and the variance of each class is

$$s_i^2 = \frac{1}{N_i - 1}\sum_{y\in c_i}(y-\mu_i)^2 \tag{17}$$

Then $S_b = |\mu_1 - \mu_2|^2$ and $S_w = s_1^2 + s_2^2$. By maximizing J, FLD can find a $\mathbf{w}$ to project the origin two classes to the scalar space where the distribution of each class is tight, whilst, the distinction between classes is sharp.

For C classes, we can obtain C covariance matrixes $S_1, S_2, \ldots, S_C$. The average covariance matrix is

$$S_w = \frac{1}{C}\sum_{i=1}^{C} S_i \tag{18}$$

Then the biggest eigenvalue λ_w of S_w measures the average scatter degree. The smaller λ_w is, the denser each class is.

2. The Proposed Objective Function

(1) The calculation of between class scatter

For Multi-classification and high-dimensional situations, we assume there are N samples $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}, \mathbf{x} \in \mathbb{R}^{m\times 1}$ that can be separated into C classes. N_i of samples belong to class c_i, $i = 1, 2, \ldots, C$. The mean vector of each class is

$$\boldsymbol{\mu}_i = \frac{1}{N_i}\sum_{\mathbf{x}\in c_i}\mathbf{x} \tag{19}$$

To quantify the between-scatter, we define the covariance matrix of $\boldsymbol{\mu}_i$ as the between-scatter-matrix:

$$S_b = \frac{1}{C-1}\sum_{i=1}^{C}(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})^T \tag{20}$$

where $\bar{\boldsymbol{\mu}}$ is the expectation of $\boldsymbol{\mu}_i$:

$$\bar{\boldsymbol{\mu}} = \frac{1}{C}\sum_{i=1}^{C}\boldsymbol{\mu}_i \tag{21}$$

We need only $C-1$ principal components of S_b. And the $(C-1)th$ largest eigenvalue λ_b of S_b quantifies the amount of scatters between classes, where a larger λ_b declares better discrimination between classes.

(2) The calculation of Intra class scatter

The covariance matrix of each class is

$$cov_i = \frac{1}{N_i - 1} \sum_{\mathbf{x} \in c_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T \tag{22}$$

And to quantify the between-scatter, we define the sum of cov_i as the within-scatter-matrix:

$$S_w = \sum_{i=1}^{C} cov_i \tag{23}$$

We need only 1 principal components of S_w. S_w's largest eigenvalue λ_w quantifies the degree of dispersion within each class, where a smaller λ_w represents denser classes.

(3) construct the objective function

Thus, considering the variance of x has positive correlation with x^2, we define an objective function as:

$$\mathrm{O} = \sqrt{\lambda_w} + 1/\sqrt{\lambda_b} \tag{24}$$

By minimizing O, the CNN's output will have the best separability. Then a simple liner classifier, such as a Support Vector Machine (SVM) [15], can easily discriminate each class.

IV. Experiments

1. Network Structure

To evaluate the objective function, we build a CNN as a feature extractor. Table 1 shows the structure of the CNN. Except the output layer it contains three convolutional layers, two dense layers and one output layer. For instance, the parameter "20@5*5" means there are 20 convolutional kernels with shape 5*5, and parameter "128" means there are 128 neurons in a full-connected layer. The output layer has no activate function. And the objective function is the proposed one.

Table 1 The structure of the feature extractor

	Layer	Setting
1	Convolutional	20@5*5
2	Convolutional	40@3*3
3	Convolutional	80@3*3
4	Dense	2880
5	Dense	5760
6	Output Dense	9

In order to compare the system performance on classification tasks, we also build a standard CNN for Multi-classification shown in **Error! Reference source not found.**. The difference is the output layer has 10 neurons and a Softmax activate function [11] while the output layer in Table 1 has no activate function. And the objective function is the CCE cost function.

Table 2 The structure of the standard CNN

	Layer	Setting
1	Convolutional	20@5*5
2	Convolutional	40@3*3
3	Convolutional	80@3*3
4	Dense	2880
5	Dense	5760
6	Output Dense	10

2. The Dataset

The training and test samples in this paper are from the MSTAR database [7], including 10 classes: 2S1, ZSU234, BMP2, BRDM2, BTR60, BTR70, D7, ZIL131, T62, and T72. The size of each sample is 64*64, and the resolution is 0.3 m*0.3 m, whilst the depression angle 17 degrees and 15 degrees. We use the 17 degrees' pitch angle images as the training data and the 15 degrees' as the test data. Table 3 shows the detail of the data.

Table 3 The training and test set

Types	Training set		Testing set	
	Depression	Number	Depression	Number
2S1	17°	299	15°	274
ZSU234	17°	299	15°	274
BRDM2	17°	298	15°	274
BTR60	17°	256	15°	195
BMP2	17°	233	15°	195
BTR70	17°	233	15°	196
D7	17°	299	15°	274
ZIL131	17°	299	15°	274
T62	17°	299	15°	273
T72	17°	232	15°	196
	Total:2747		Total:2425	

3. Experimental Results

(1) Separability analysis

For visualization, we project the outputs of every two classes to a two-dimensional space using PCA, and draw all the points in Figure 5. Before that, to compare with the original data, we also project the original samples of the two classes using PCA. The results are shown in **Error! Reference source not found.**, and they are not completely linearly separable.

The lower left area in Figure 5 shows every two classes on the training set are linearly separable. But on the test set the performance is not as good as the training set because of the overfitting, where some classes gather together but some do not, and some points even fall into the wrong classes. In the next subsection we will train a linear classifier to count the wrong points.In addition, we draw a group of four classes in a three dimensional space to observe their spatial distributions, as shown in Figure 6. And they are linearly separable, but on the test set, the performance is not as good as that of the training set.

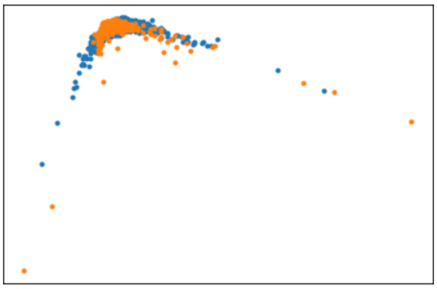


Figure 4 The distribution of a pairs of classes on original train set.

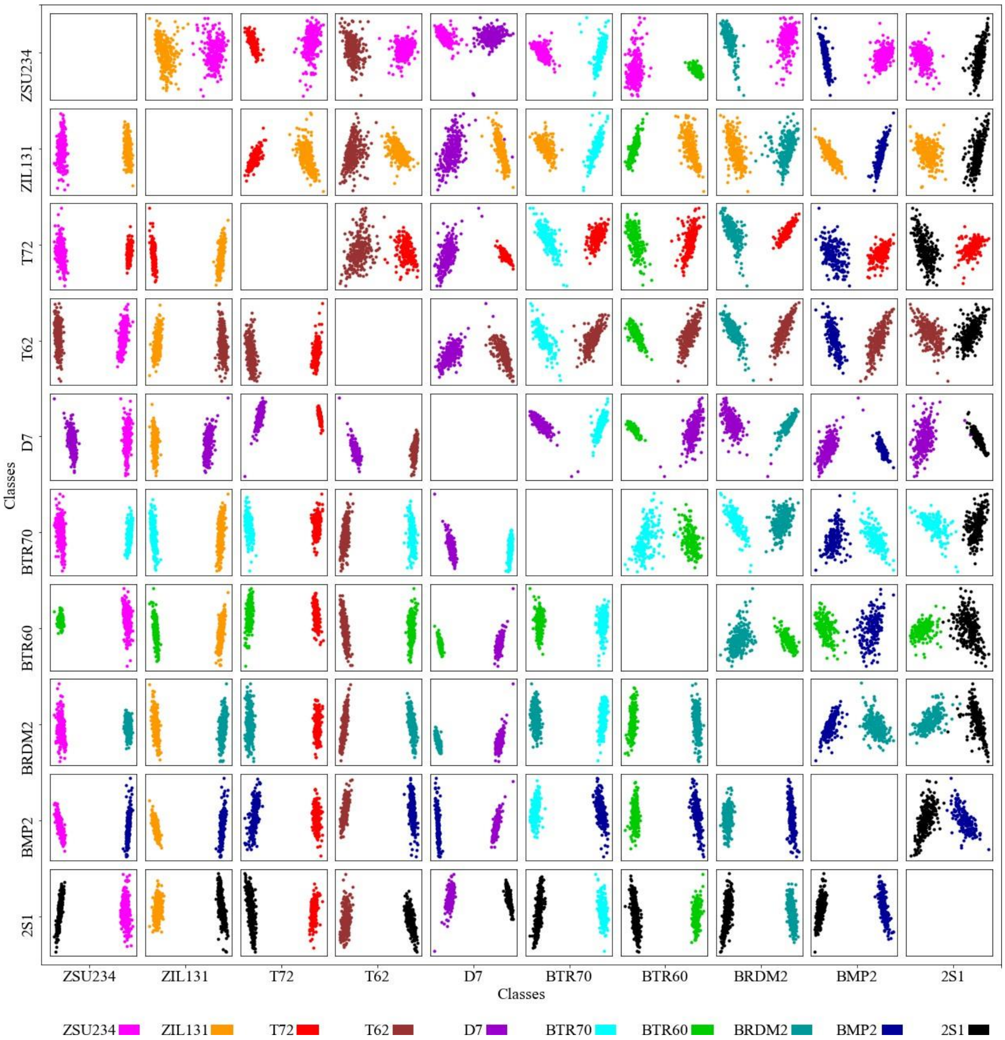
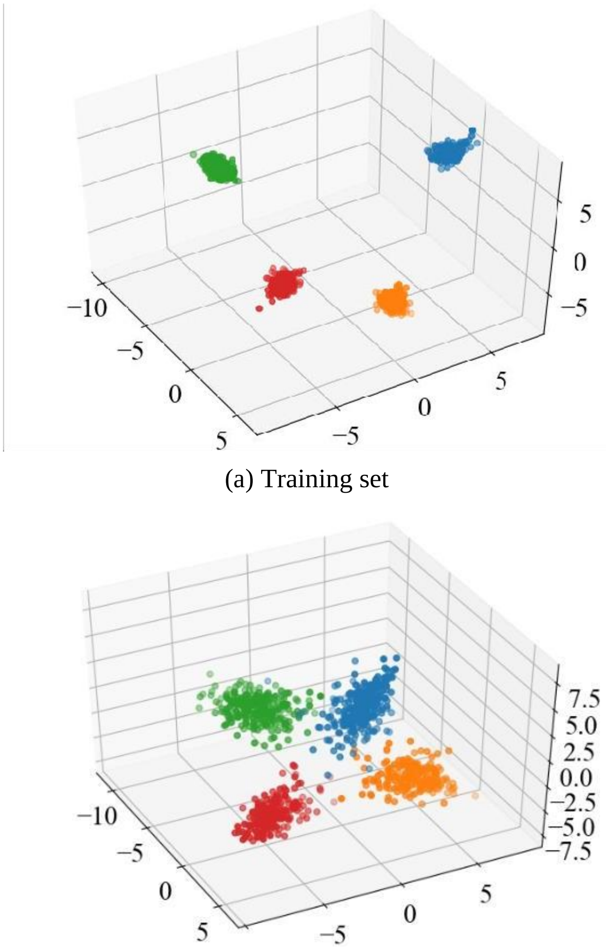


Figure 5 The distribution of all pairs of classes. The lower left area is the training set and the upper right area is the test set. Using PCA we project the outputs of every two classes to a two dimensional space. The coordinates of each square are different.



(a) Training set

(b) Test set

Figure 6 The distribution of four classes on training and test set. Obviously, both the two cases are linearly separable.

(2) Classification Results

The experiment aims to classify the 10 types of targets from the MSTAR database. Several indexes are necessary to evaluate the experimental results, including: the training accuracy, evaluation accuracy and the convergence.

The training accuracy of the two experiments is 100%. The evaluation accuracy of the proposed experiment is 98.85%, while the standard method is 98.22 %.

Table 4 and Table 5 show the classification results of the two methods, where precision, recall, f-measure, and support are selected as the evaluation indexes. Precision, Recall [18]and F_1 - score [19] are defined as follows:

$$Precision=\frac{N_{cr}}{N_r} \tag{25}$$

$$Recall=\frac{N_{cr}}{N_t} \tag{26}$$

$$F_1-score=2\times\frac{Precision\times Recall}{Precision+Recall} \tag{27}$$

where N_{cr}, N_t and N_r represent the number of samples correctly recognized as the specified class, the number of samples recognized as the specified class, the total number of samples in the specified class, respectively.

The F1-score is a comprehensive measure. It is the average of the precision and recall.1 means the best performance while 0 indicates the worst. The two tables show the proposed method performs better in all three aspects.

Table 4 The result of proposed method

Class	Precision	Recall	F1-score	Support
2S1	1.00	0.95	0.97	274
BMP2	0.99	0.99	0.99	195
BRDM2	0.99	0.99	0.99	274
BTR70	0.95	1.00	0.97	196
BTR60	1.00	0.99	0.99	195
D7	1.00	0.97	0.99	274
T62	1.00	1.00	1.00	273
T72	0.99	1.00	0.99	196
ZIL131	0.98	1.00	0.99	274
ZSU234	0.98	1.00	0.99	274
Avg/Total	0.99	0.99	0.99	2425

Table 5 The result of the standard method

Class	Precision	Recall	F1-score	Support
2S1	0.99	0.93	0.96	274
BMP2	0.97	1.00	0.99	195
BRDM2	0.98	0.98	0.98	274
BTR60	0.99	0.98	0.99	196
BTR70	1.00	0.99	0.99	195
D7	0.99	0.99	0.99	274
T62	0.94	0.99	0.97	273
T72	0.99	0.97	0.98	196
ZIL131	0.98	0.99	0.99	274
ZSU234	0.99	0.99	0.99	274
Avg/Total	0.98	0.98	0.98	2425

Table 6 and Table 7 show the confusion matrixes of each classification [20]. Where each number in the tables indicates the number that the class of the ordinates is predicted to be the horizontal coordinate class. The blue background means the two classifiers which make mistakes, and the yellow background means the individual mistakes. The comparison of the two tables shows that the proposed method misclassifies less classes.

Table 6 The details of the classification by proposed method

Class	2S1	BMP2	BRDM2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU234
2S1	260	1	2	8	0	0	1	2	0	0
BMP2	0	194	0	1	0	0	0	0	0	0
BRDM2	0	0	271	0	0	0	0	0	3	0
BTR70	0	0	0	196	0	0	0	0	0	0
BTR60	0	0	0	2	193	0	0	0	0	0
D7	0	0	0	0	0	267	0	0	2	5
T62	0	0	0	0	0	0	272	0	1	0
T72	0	0	0	0	0	0	0	196	0	0
ZIL131	0	0	0	0	0	0	0	0	274	0
ZSU234	0	0	0	0	0	0	0	0	0	274

Table 7 The details of the standard method

Class	2S1	BMP2	BRDM2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU234
2S1	256	2	4	1	0	0	11	0	0	0
BMP2	0	195	0	0	0	0	0	0	0	0
BRDM2	0	0	269	0	0	0	0	0	4	1
BTR70	1	2	0	193	0	0	0	0	0	0
BTR60	0	0	1	0	193	0	1	0	0	0
D7	0	0	1	0	0	272	0	0	1	0
T62	0	0	0	0	0	1	270	1	0	1
T72	1	1	0	0	0	0	3	191	0	0
ZIL131	1	0	0	0	0	0	0	0	272	1
ZSU234	0	0	0	0	0	2	1	0	0	271

Figure 7 shows the changing process of accuracy during the training. Obviously, the proposed method obtains the best accuracy slower but more smoothly, while the standard method performs faster but less smoothly. The lack of training data leads to overfitting, which makes the test valuation curves fluctuation. Therefore, the proposed method is more suitable for the situation of less training data.

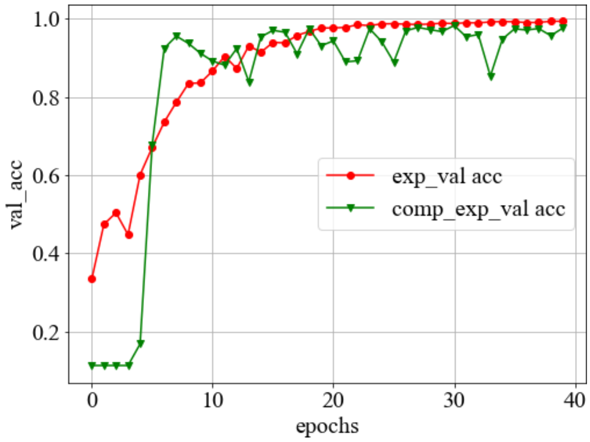


Figure 7 The evaluation accuracy during the training process of each CNN. The “exp_val acc” means the accuracies of the proposed experiment and ‘comp_exp_val acc’ denotes the standard experiment.

V. Conclusion

Aiming at extracting linearly separable features from different categories of SAR image targets, we proposed a novel linear separability objective function. We extend the mathematical meaning of linear separability of multiple classes, which is an arbitrary subset of the classes linearly separable from the rest. As a sufficient condition for linear separability, this proposition has a simple mathematical description, and is easy to establish an algorithm for optimization. The optimization objective of the proposed objective function is to be zero but that is impossible. Then a threshold can be taken as the optimization target according to the training situation. In addition, we derived an inference that the output dimension of the CNN must be bigger than C-1, where C is the number of classes. And it is a guidance for dimensional reduction tasks.

We have used experiments to judge the separability of output features. Instead of testing the separability by all the subsets, we have just tested the criterion of One-Vs-Rest and Pairwise. By mapping all classes to the two-dimensional space by pair and projecting a group of four classes to a three dimensional space, we can intuitively find the obvious separability on the test set, though is not as good as the training set.

Extra experiments on classification task have counted the number of misclassified samples and the total classification accuracy. Meanwhile, the experiments compared the proposed method and the CCE cost function. The experimental results show that the former is more stable in case of few samples. Moreover, we can add the proposed objective function and other objective functions together to train a network.

In summary, we believe the proposed method has a certain application prospect in the aspects of feature extraction, nonlinear dimensionality reduction and classification.

References

1. Shi, H., et al., *SAR Slow Moving Target Imaging Based on Over-Sampling Smooth Algorithm.* Chinese Journal of Electronics, 2017. **26**(4): p. 876-882.
2. Wang, G., et al., *Multiple model particle flter track-before-detect for range ambiguous radar.* Chinese Journal of Aeronautics, 2013. **26**(6): p. 1477-1487.
3. Lécun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.
4. He, K., et al. *Deep Residual Learning for Image Recognition*. in *Computer Vision and Pattern Recognition*. 2016.
5. Razavian, A.S., et al., *CNN Features Off-the-Shelf: An Astounding Baseline for Recognition.* 2014: p. 512-519.
6. Chen, S. and H. Wang. *SAR target recognition based on deep learning*. in *International Conference on Data Science and Advanced Analytics*. 2015.
7. Yang, Y., Y. Qiu, and C. Lu. *Automatic Target Classification " Experiments on the MSTAR SAR Images*. in *International Conference on Software Engineering, Artificial Intelligence, NETWORKING and Parallel/distributed Computing and First Acis International Workshop on Self-Assembling Wireless Network*. 2005.
8. Li, X., et al. *SAR ATR based on dividing CNN into CAE and SNN*. in *Synthetic Aperture Radar*. 2015.
9. Ding, J., et al., *Convolutional Neural Network With Data Augmentation for SAR Target Recognition.* IEEE Geoscience & Remote Sensing Letters, 2016. **13**(3): p. 364-368.
10. Razavian, A.S., et al. *CNN Features Off-the-Shelf: An Astounding Baseline for Recognition*. in *Computer Vision and Pattern Recognition Workshops*. 2014.
11. Dugas, C., et al. *Incorporating Second-Order Functional Knowledge for Better Option Pricing*. in *neural information processing systems*. 2001.
12. Wold, S., K. Esbensen, and P. Geladi, *Principal component analysis.* Chemometrics & Intelligent Laboratory Systems, 1987. **2**(1–3): p. 37-52.
13. Wang, X., W. Zhang, and Q. Ji, *A Kernel PCA Shape Prior and Edge Based MRF Image Segmentation.* Chinese Journal of Electronics, 2016. **25**(5): p. 892-900.
14. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* Nature, 1986. **323**(6088): p. 533-536.
15. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine Learning, 1995. **20**(3): p. 273-297.
16. Kumar, M.A. and M. Gopal, *A comparison study on multiple binary-class SVM methods for unilabel text categorization.* Pattern Recognition Letters, 2010. **31**(11): p. 1437-1444.
17. Fisher, R.A., *The Use of Multiple Measurements in Taxonomic Problems.* Annals of Human Genetics, 1936. **7**(2): p. 179-188.
18. Davis, J. and M. Goadrich. *The relationship between Precision-Recall and ROC curves*. in *ICML '06 : Proceedings of the International Conference on Machine Learning, New York, Ny, Usa*. 2006.
19. Hripcsak, G. and A.S. Rothschild, *Agreement, the F-Measure, and Reliability in Information Retrieval.* J Am Med Inform Assoc, 2005. **12**(3): p. 296-298.
20. Stehman, S.V., *Selecting and interpreting measures of thematic classification accuracy.* Remote Sensing of Environment, 1997. **62**(1): p. 77-89.