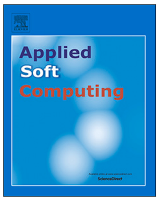
Applied Soft Computing 123 (2022) 108922

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc



Spatial feature-based convolutional neural network for PolSAR image classification[☆]

Ronghua Shang [a,*], Jiaming Wang [a], Licheng Jiao [a], Xiaohui Yang [b], Yangyang Li [a]

[a] *Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi Province 710071, China*
[b] *Data Analysis Technology Lab, Institute of Applied Mathematics, Henan University, Kaifeng 475004, China*

ARTICLE INFO

Article history:
Received 23 June 2019
Received in revised form 7 January 2022
Accepted 19 April 2022
Available online 30 April 2022

Keywords:
Deep learning
PolSAR image classification
Convolutional neural network

ABSTRACT

The deep learning algorithm has made great breakthroughs in optical image processing. Some deep learning algorithms require a large number of labeled samples for training. For PolSAR data sets, due to the influence of speckle noise and other factors, high-quality labeled data are limited. Therefore, it is meaningful to use deep learning algorithm to solve PolSAR classification problem in limited labeled dataset. This paper proposes a spatial feature-based convolutional neural network (SF-CNN). The network adopts a dual-branch CNN structure. Both of the two branches have the same structure and share parameters. SF-CNN can receive more than one sample as input. SF-CNN's special structure can expand the original training set by combining different samples, and alleviate the problem of insufficient labeled training data in PolSAR image classification tasks. When training, SF-CNN maps high-dimensional PolSAR image to low-dimensional feature space. In low-dimensional feature space, SF-CNN enhances the ability of network to extract discriminative features by maximizing or minimizing the distance between feature centers of different classes. In order to dig up the relationship between the samples, the test sample features are compared with every training sample feature when testing. Finally, labels of test samples are determined by the comparison result. The result of SF-CNN in PolSAR image classification task is better than that of standard CNN.

1. Introduction

Polarimetric synthetic aperture radar (PolSAR) overcomes the shortcomings of optical remote sensing images which are sensitive to illumination and weather. PolSAR system can work at all-weather condition, and has a strong penetrating ability. It can accurately describe the polarimetric scattering characteristics of objects on the earth's surface. The advantages of PolSAR make it widely used in earth observation, resource exploration, disasters pre-assessment and troop disposition [1,2].

The PolSAR classification problem has attracted more and more attention from researchers. It is one of the essential steps in PolSAR image interpretation. In this task, every pixel in PolSAR image will be labeled as one class.

1.1. Related work on three kinds of polsar classification algorithms

The first category is the algorithm based on the scattering mechanism of PolSAR data, which mainly considers the physical principle and imaging mechanism of PolSAR images. We can get different information by decomposing the covariance matrix, scattering matrix or coherent matrix of PolSAR image. Through these information, different features of PolSAR data can be obtained. The representative algorithms include Pauli decomposition [3], Cameron decomposition [4], Freeman decomposition [5], H/alpha decomposition [6], Huynen decomposition [7], Krogager decomposition [8] and so on. The algorithms based on scattering mechanism are simple and effective, and can be applied to many PolSAR dataset.

The second category is the algorithm based on statistical distribution. Researchers usually model PolSAR data based on different distributions. Lee et al. proposed an unsupervised PolSAR classification method which utilizes H/alpha decomposition and complex Wishart classifier [9]. Liu et al. proposed Wishart deep belief network (W-DBN) to classify PolSAR images with local spatial information [10]. Jiao and Liu put forward a classification algorithm by combining Wishart distribution with Deep Stacking Network (W-DSN) [11]. Xie et al. proposed Wishart-AE and Wishart-CAE algorithms, which are based on complex Wishart distribution and Convolutional Autoencoder (CAE) [12]. In addition, polarimetric G distribution [13] and polarimetric K distribution [14] are also often used to describe PolSAR data.

[☆] The code of the proposed algorithm has been uploaded and available at the following link: https://github.com/Vita427/SF-CNN.

[*] Corresponding author.
E-mail address: rhshang@mail.xidian.edu.cn (R. Shang).

https://doi.org/10.1016/j.asoc.2022.108922
1568-4946/© 2022 Elsevier B.V. All rights reserved.

The third category is the algorithm based on machine learning or deep learning. Some commonly used methods include Support Vector Machine (SVM) [15,16], K-Singular Vector Decomposition (KSVD) [17], methods based on graph cut [18] and Markov Random Field (MRF) [19,20] and so on. Deep neural networks show extraordinary talents in natural images processing. The representative networks include LeNet [21], VGG [22], ResNet [23], DenseNet [24] and so on. Some deep learning algorithms also performs well in PolSAR image classification. Hou et al. designed multilayer autoencoders with superpixels to distinguishing the multiple classes of PolSAR data [25]. Zhou et al. designed a new CNN to extract spatial features of PolSAR data and achieved good results [26]. Zhang et al. designed a complex convolutional neural network (CV-CNN) [27], which get better results than real-valued CNN in PolSAR classification task.

1.2. Motivation and innovation

Although deep learning algorithm [28,29] has stronger fitting ability than other algorithms, the performance of deep learning algorithms is restricted by the number of training samples. Insufficient samples are more likely to bring about overfitting problem. In order to improve the performance of deep learning algorithms under the condition of insufficient training sample, some researchers have proposed deep metric learning methods. Deep metric learning can extract simple and effective feature from complicated data using deep neural networks. Due to the influence of speckle noise and other factors in polarimetric SAR data set, high quality annotation data is very limited [30]. These networks can accept multiple samples input at the same time in the architecture to achieve the purpose of sample expansion to alleviate the problem of insufficient annotation data [31]. There are a lot of deep metric neural networks. Siamese neural network [32,33] is a representative deep metric learning algorithm, it was first proposed by Bromley et al. to solve the signature verification problem. Matching Nets were proposed by Vinyals et al. [34], which use attention and external memories in network to do rapid feature learning. Snell et al. proposed prototypical networks [35]. It can prototype representations of different classes to classify every sample in metric space. Sung et al. proposed Relation Network [36], which computes relation scores between query samples and support samples to do classification effectively.

While in the current PolSAR classification task, there are two problems that should not be ignored. One is the limited number of available labeled samples, and the other one is the low quality of the samples, which are affected by the speckle noise of PolSAR data. Therefore, the number of labeled samples that can be used is very scarce. To solve this problem, we design a new spatial feature-based network (SF-CNN) inspired by deep metric learning and CNN. Compared with standard CNN, SF-CNN have the following characteristics:

(1) SF-CNN uses a dual branch CNN structure, and the two branches have the same structure and share the training weights. The network uses a sample group instead of single sample as input. For each branch of SF-CNN, one of training set categories is randomly selected, and then several samples are randomly collected from the selected category as a sample group. SF-CNN uses the sample group formed by multiple samples as input, which greatly enhance the number of original training samples and alleviating the problem of lacking labeled data.

(2) SF-CNN can generate low-dimensional features from high-dimensional data. Each branch CNN extracts a feature group from the input sample group, and the feature center of the extracted feature group will be calculated. Then, according to the labels of feature centers, the feature centers belonging to the same class will be pulled together, while the feature centers from different class will be pushed away. In this way, the feature center distance of same class is closer, and for different class it is far away, which makes the features extracted by CNN more distinguishable.

(3) When predict the label of test data, SF-CNN first transforms the samples in the training set into feature vectors. Then, a k-Nearest Neighbor (KNN) [37] model is constructed by the feature vectors of training samples, and the distance between the test sample feature vectors and every training sample feature vector is calculated. Finally, the test samples are labeled according to the calculation results. This method takes account of the relationship between samples and strengthens the utilization of sample information.

SF-CNN and standard CNN are compared on several real PolSAR datasets. The experimental results show that SF-CNN is more effective than standard CNN in multiple datasets.

The structure of this paper is arranged as follows. In Section 2, this paper will introduce the specific methods of SF-CNN. In Section 3, SF-CNN will be tested and the experimental results will be analyzed. Finally, in Section 4, research content of this article will be summarized.

2. The structure and method of SF-CNN

In this section, the preprocessing of PolSAR data will be given. The input and the entire structure of SF-CNN will be introduced. The training method, test method and network settings of SF-CNN will be detailed explained.

2.1. Preprocessing of PolSAR data

In PolSAR image, according to scattering mechanism, each PolSAR pixel can be briefly described by a complex scattering matrix $\mathbf{S}$ and its size is 2×2. $\mathbf{S}$ can be expressed as:

$$\mathbf{S} = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{vh} & S_{vv} \end{bmatrix} \tag{1}$$

There are four scattering parameters in S, namely S_{hh}, S_{hv}, S_{vh}, S_{vv}. The First subscript of S_{hh} is the transmitting signal polarization, and the second subscript is the receiving signal polarization. v and h are the polarization direction of electromagnetic wave (vertical direction and horizontal direction). If the data was collected by a monostatic PolSAR radar system, then $S_{hv} = S_{vh}$ and S is symmetric [10].

Covariance matrix and coherent matrix of PolSAR data have fully polarimetric information of target. They can well describe the statistical scattering characteristic of data. With Eq. (1), we can get the coherent matrix of each pixel in PolSAR image. The coherency matrix $\mathbf{T}$ is written as:

$$\mathbf{T} = [a, b, c]^{\mathrm{T}} \left[a^*, b^*, c^*\right] = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{12}^* & T_{22} & T_{23} \\ T_{13}^* & T_{23}^* & T_{33} \end{bmatrix}$$
$$= \begin{bmatrix} |a|^2 & ab^* & ac^* \\ a^*b & |b|^2 & bc^* \\ a^*c & b^*c & |c|^2 \end{bmatrix} \tag{2}$$

where "*" is conjugate of number and a, b, c are three scattering vectors obtained by the Pauli decomposition of $\mathbf{S}$, which is collected by a monostatic PolSAR radar system. a, b, c can be written as:

$$a = \frac{\sqrt{2}}{2}\left(S_{hh} + S_{vv}\right), b = \frac{\sqrt{2}}{2}\left(S_{hh} - S_{vv}\right), c = \sqrt{2}S_{hv} \tag{3}$$

In coherency matrix **T**, diagonal elements including T_{11}, T_{22}, T_{33} are real number and off-diagonal elements including T_{12}^{*}, T_{13}^{*}, T_{23}^{*}, T_{21}, T_{31}, T_{32} are complex number. **T** is a complex matrix. For the convenience of calculation, every elements' imaginary parts are converted to real value. After removing repetitive elements, a 9-dimensional real-valued vector T_v is obtained:

$$T_v = \{T_{11}, T_{22}, T_{33}, \mathrm{real}(T_{12}), \mathrm{real}(T_{13}), \mathrm{real}(T_{23}), \mathrm{imag}(T_{12}), \mathrm{imag}(T_{13}), \mathrm{imag}(T_{23})\} \tag{4}$$

Through the above operations, each pixel in PolSAR data is transformed into T_v.

2.2. Training process of SF-CNN

SF-CNN adopts dual-branch CNN structure, in which each branch CNN accepts a sample group instead of a single sample as input. Suppose that there are totally C classes in training data set, and each class contains N training samples. SF-CNN randomly selects $2k$ ($k < N$) samples from the training data set as input every time as shown in Fig. 1. The first branch CNN randomly selects 1 class from the C classes of training set, and then randomly selects k samples from this class. The k samples form a sample group, and the sample group is recorded as x_a, where $x_a = \{x_{a1}, x_{a2}, \ldots, x_{ak}\}$. Similarly, the sample group of the second branch CNN is recorded as x_b, where $x_b = \{x_{b1}, x_{b2}, \ldots, x_{bk}\}$. x_a and x_b are input to SF-CNN, and the combination of them, $\{x_a, x_b\}$, can form a new data set. Compared with original data set, the sample number of new data sets is greatly increased.

The number of negative samples (x_a and x_b do not belong to the same class) is:

$$N_0 = \binom{C}{2} \cdot \binom{N}{k} \cdot \binom{N}{k} = \frac{C \cdot (C-1)}{2} \cdot \left(\frac{N!}{k! \cdot (N-k)!}\right)^2 \tag{5}$$

The number of positive samples (x_a and x_b belong to the same class) is:

$$N_1 = C \cdot \left[\binom{\binom{N}{k}}{2} + \binom{N}{k} \right] \tag{6}$$

The total number of new samples is:

$$N' = \binom{C}{2} \cdot \binom{N}{k} \cdot \binom{N}{k} + C \cdot \left[\binom{\binom{N}{k}}{2} + \binom{N}{k} \right] \tag{7}$$

Suppose there are 5 classes and 5 samples for each class. The size of the sample group is 2, namely, $C = 5$, $N = 5$, $k = 2$. Using Eq. (7), it can be calculated that the sample number of new data set is 1275, while the original sample number is only 25. Therefore, the sample number is greatly expanded.

In sample group, each sample is represented as a window patch instead of a single pixel point. It contains a $U_x \times U_x$ local pixel block centered on one pixel. The local window patch not only captures the polarimetric information of data, but also contains the neighboring spatial information around the central pixels. It is beneficial to CNN to learn effective features of data, and many algorithms use it to improve their performance.

The network structure and training process of SF-CNN are shown in Fig. 1.

As can be seen from Fig. 1, SF-CNN has two parallel CNN branches. These two branches share the same parameters as shown in Table 1 in Section 2.5. This dual-branch CNN structures can reduce network parameters by half. During training, sample group x_a and x_b are input to SF-CNN. After feature extraction and dimension reduction, low-dimensional feature groups will be generated. Assuming that the feature extraction process of SF-CNN can be expressed as $F_W(x)$ where and W is the weight of the network. The feature generated by the sample x can be written as $f = F_W(x)$. The feature group generated by sample group x_a is $\{f_{a1}, f_{a2}, \ldots, f_{ak}\}$. Through the feature group, the group center can be calculated. We use the mean value of multiple feature vectors to represent the feature group center. The feature group center f_a of x_a can be written as:

$$f_a = \frac{f_{a1} + f_{a2} + \cdots + f_{ak}}{k} \tag{8}$$

Similarly, the feature group center f_b of x_b can be written as:

$$f_b = \frac{f_{b1} + f_{b2} + \cdots + f_{bk}}{k} \tag{9}$$

Thus, when training (learning process), a gradient update process is like this: input group x_a into the CNN 1 to get the feature f_a, input group x_b into the CNN 2 (same parameters as CNN 1) to get the feature f_b, according to f_a, f_b, calculate the loss and carry on the back propagation to the loss to update the network parameters.

2.3. Loss function

Through SF-CNN, two high-dimensional sample groups x_a and x_b are mapped to the low dimensional feature space, and the corresponding feature centers f_a and f_b are obtained. The distance between the two feature centers can be represented as:

$$D_W = |F_W\,(x_a) - F_W\,(x_b)| = |f_a - f_b| \tag{10}$$

When the input samples x_a and x_b belong to the same class, D_W should be relatively a small value. And when x_a and x_b' are not belong to the same class, it should be relatively a large value. The proving process can be found in [32].

$$D_W(f_a, f_b) + \delta < D_W(f_a, f_b') \tag{11}$$

where δ is a positive constant. In Eq. (11), a proper W can be obtained by training the network, which make the distance between the same-class samples are smaller than the distance between the different-class samples.

In order to make SF-CNN have the ability to extract discriminative features, we use the contrastive loss function to train the entire network. Suppose that there are N' samples in new data set. Each sample is recorded as $(x_a, x_b)^i$, and its label is y_i, where $i = 1, 2, \ldots, N'$. When x_a and x_b belong to the same class, $y_i = 1$. When x_a and x_b belong to different classes, then $y_i = 0$. The loss function is written as:

$$Ld\,(W) = \frac{1}{2}\sum_{i=1}^{N}(D_W)^2 \cdot y_i + \max(\alpha - D_W, 0)^2 \cdot (1 - y_i) \tag{12}$$

The threshold value α is a constant, and $\alpha > 0$. When $y_i = 1$, the input two sample groups belong to the same class. The first term of the formula works, and the loss function reduces the distance between the same features. When $y_i = 0$, the two input sample groups do not belong to the same class. At this time, the second term of the formulas works, and the loss function increases the distance between different-class features until the distance is greater than the threshold value α. Therefore, the penalty range of Euclidean distance is not so important when the value of α is small to a certain extent.

2.4. Test process of SF-CNN

When testing the network, test sample x_t is input to the network and generate the sample feature f_t. Similarly, training samples $\{x_1, x_2, \ldots, x_{CN}\}$ can generate multiple features $\{f_1, f_2, \ldots,$

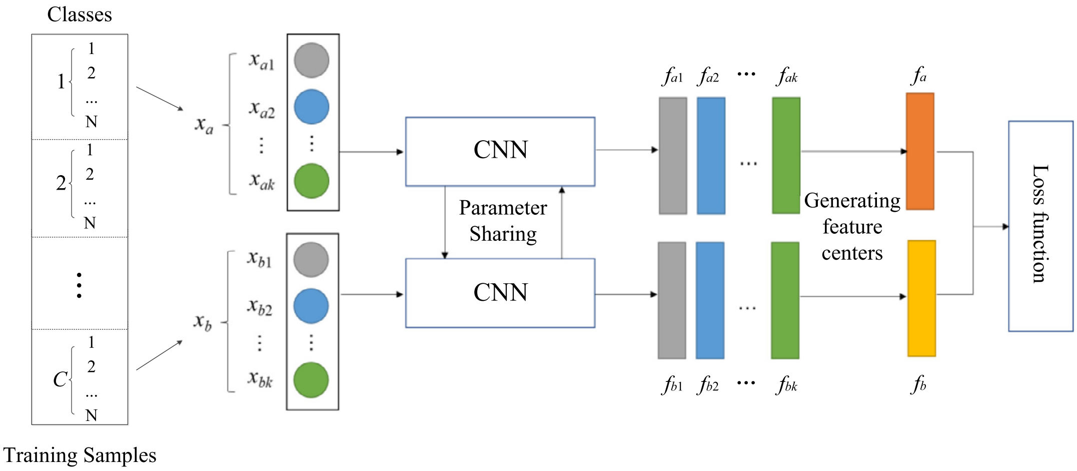


Fig. 1. Network structure and training process of SF-CNN.

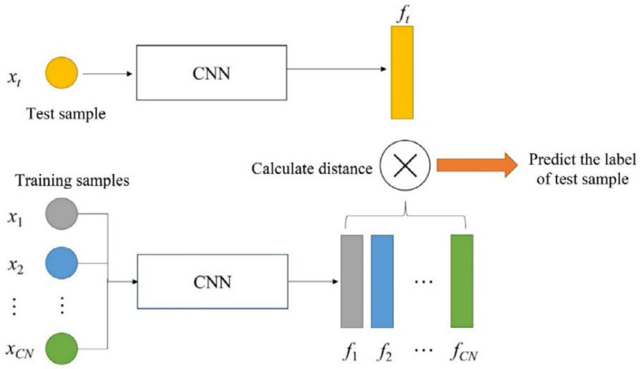


Fig. 2. Test process of SF-CNN.

f_{CN}}. With D_W, the distance between feature f_t and the training sample features {f_1, f_2, …, f_{CN} } can be calculated. k_t training samples feature having the shortest distance with f_t are selected. Counting their true labels, the most frequent label will be the label of x_t.

The testing process of SF-CNN is shown in Fig. 2.

The whole testing algorithm process is shown in Algorithm 1.

```
Algorithm 1: SF-CNN test algorithm
Input: Iterations: $iter$, test samples $x_t$, training samples $\{x_1, x_2, \ldots, x_{CN}\}$, network parameter $W$ and $k_t$.
Output: $y_t$.
for 1: $iter$
    $f_t$=$F_W(x_t)$←input $x_t$ into the CNN network;
    feature group{ $f_1, f_2, \ldots, f_{CN}$ }←input { $x_1, x_2, \ldots, x_{CN}$ } into the CNN network;
    for $i$=1: $CN$
        $D_W(x_t, x_i)$ ← Eq. (10);
    end
    $D_{W1}$←select the smallest $k_t$ value of $D_W$;
    for $j$=1: $k_t$
        count the label of $D_{W1}(x_t, x_j)$;
    end
end
output: the most-counted label $y_t$ .
```

This process is equivalent to using KNN algorithm to model the features of training samples, and then use the built KNN model to classify test samples. This method uses training sample features to predict the labels of test samples. It takes account of the relations between samples, which is conducive to improving the performance of the network.

Table 1
Network structure of SF-CNN.

Layer	Type/stride	Filter	Input feature map
1	Conv/s1	6 × 6 × 9 × 32	15 × 15 × 9
2	Max Pool/s2	2 × 2	10 × 10 × 32
3	Conv/s1	3 × 3 × 64	5 × 5 × 32
4	Dropout	0.5	3 × 3 × 64
5	Conv/s1	3 × 3 × 128	3 × 3 × 64

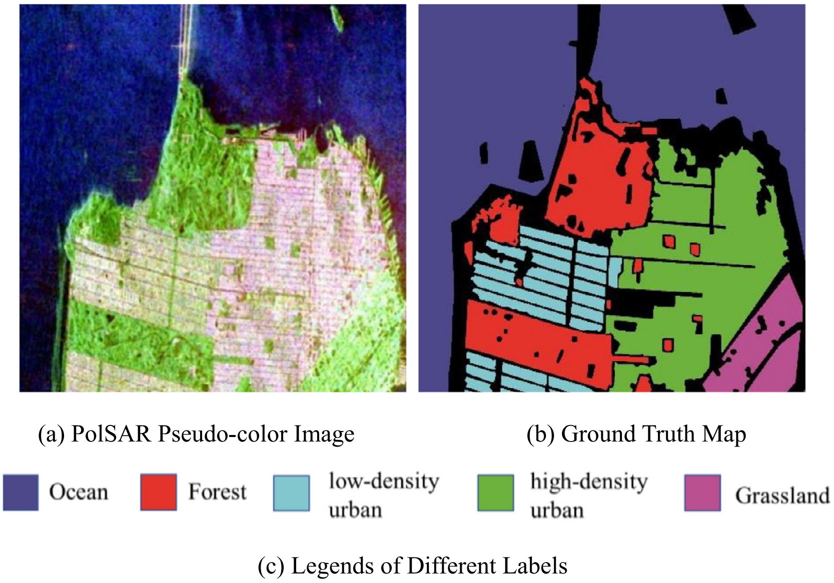
2.5. Network settings

SF-CNN adopts a dual-branch CNN structure, the two branches have same structure, and the structure of CNN is shown in Table 1.

In Table 1, the s1 represents the stride is 1, and s2 represents the stride is 2. The network is a full convolutional neural network. It only contains convolutional layer and pooling layer, and the fully connected layer is removed. With this special structure, the parameters of the network are greatly reduced, which effectively prevents the network from overfitting. The whole network has three convolutional layers. The first convolutional layer is used for extracting the low-level features of the image, it is followed by a max pooling layer to reduce the size of feature maps. The size of max pooling layer is 2 × 2, and the stride is 2. In max pooling layer, the output feature's size changes to half of the input feature map [28].

The second convolutional layer can extract the middle-level features of the image. And the last convolutional layer works for extracting the high-level features of the image, its filter size is 3 × 3. Dropout layer [38] is also applied in the network. Dropout can introduce some noise component into the network and increase the generalization performance of the network. Finally, the input 15 × 15 × 9 image blocks are transformed into 1 × 1 × 128 dimensional vectors through CNN to achieve dimensionality reduction. The activation function of the network is sigmoid function, sigmoid is a continuous function and has derivative everywhere. The range of the sigmoid function is between (0, 1).

When training SF-CNN, back propagation (BP) algorithm [39] is used to optimize the network. Adam optimizer is used to update the training parameters. Adam algorithm's decay rate $\rho 1 = 0.9$ and $\rho 2 = 0.999$. And the learning rate is 0.001. The training batch size is 32, and the dropout rate is 0.5. It can be seen from Eqs. (6) and (7) that the positive and negative samples are unbalanced in the new sample set. In order to ensure the balance of positive and negative samples, we put the same number of positive and negative samples in each batch, and both of them are 16 per batch.



(a) PolSAR Pseudo-color Image

(b) Ground Truth Map

(c) Legends of Different Labels

Fig. 3. San Francisco PolSAR data set.

3. Experimental results and discussion

3.1. Comparing algorithms and data sets

In order to evaluate the performance of SF-CNN, we tested SF-CNN on the real PolSAR dataset. The three datasets are collected by the NASA Jet Propulsion Laboratory (JPL) using the Airborne SAR system (AIRSAR), and the Canadian Space Agency using the RADARSAT-2 system. These PolSAR images cover the scene of San Francisco, Xi'an and Flevoland.

We compare the experimental performance of SF-CNN with several commonly used algorithms, including SVM [16], W-DBN [10], CV-CNN [27], standard CNN [27] and VMW [40] on three different PolSAR datasets. The test evaluation indexes include overall accuracy (OA), average accuracy (AA) and Kappa coefficient. In the three experiment, the size of data patch window U_x is set to 15, and the size of the input data of the network is 15 × 15 × 9, the threshold of loss function is set to 5, the size of the sample group k is 5, and the kt value in KNN model for testing is set to 5.

3.2. Evaluation and analysis in San Francisco PolSAR data set

The data set covers the scene of San Francisco area in the United States and it is a classic data set in the PolSAR terrain classification task. The data set was collected by RADARSAT-2 platform on April 2, 2008 under 8 m spatial resolution and C band conditions. The size of the original map is 2820 × 14416, and only a part of the original map is used for the experiment. Its size is 1300 × 1300. The pseudo-color image of the data set is shown in Fig. 3(a). The ground truth map is shown in Fig. 3(b) and the legends of different label are shown in Fig. 3(c). Each color represents a category, and black represents the unlabeled category, which will not be included in the final test result. As can be seen from the figure, the data set contains 5 categories, including ocean, forest, low-density urban, high-density urban and grassland.

The number of training set samples and total samples of San Francisco PolSAR data set is shown in Table 2.

As can be seen from Table 2, the number of ocean samples is very large while the number of grassland samples is relatively small. In order to better reflect the robustness of SF-CNN algorithm, cross validation is used to obtain the classification results. The specific method is to randomly select 1000 different samples in each class as the training set and the remaining samples as the testing set. The average of the ten classification results is shown as the final result. Other algorithms do not use cross validation. Taking San Francisco data as an example, the expanded sample size is ∞ according to Eq. (7). The final results are shown in Table 3.

The final result maps of each algorithm are shown in Fig. 4.

From Table 3, we can see that SF-CNN achieves the best results on OA, AA and Kappa, which have reached 97.24%, 98.1% and 0.9715 respectively. It is easy to distinguish ocean samples, and there are significant differences between ocean samples and other samples in Fig. 3(a). All the algorithms have achieved nearly 100% accuracy on ocean category. For the classification of urban areas, SF-CNN shows obvious advantages. Especially in low-density urban areas, SF-CNN has greatly improved compared with other algorithms. Its OA is 3.7% higher than CV-CNN algorithm and 3.89% higher than standard CNN algorithm. In Fig. 4, there are very little low-density urban samples being misclassified in SF-CNN classification result map. While for CNN, CV-CNN and W-DBN, some low-density urban samples are falsely assigned as high-density urban category or forest category. For the high-density urban category, SF-CNN's OA is 2.41% higher than W-DBN algorithm and 2.69% higher than standard CNN algorithm. In vegetation category, there is little difference between SF-CNN and CV-CNN in the classification result. The OA of SVM algorithm is only 92.68%, which is much lower than other algorithms.

3.3. Evaluation and analysis in Xi'an PolSAR data set

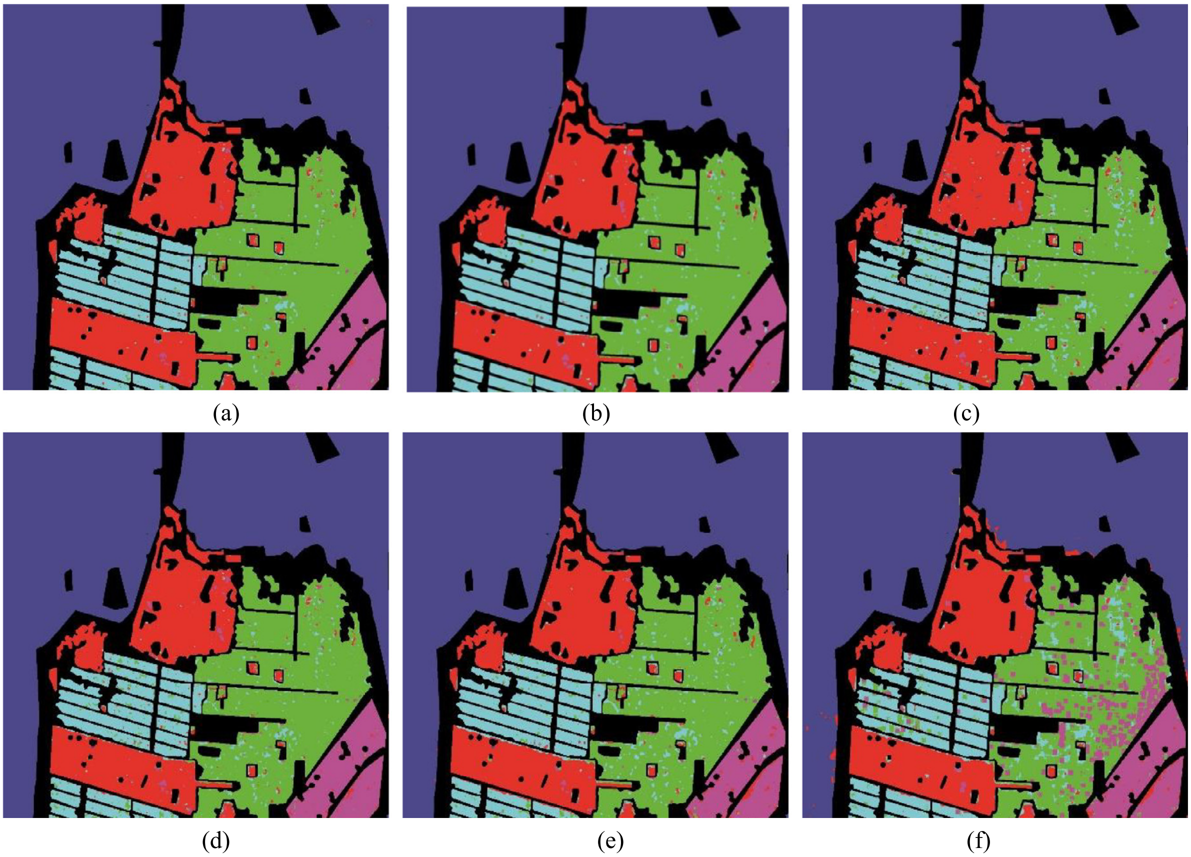
The second experimental PolSAR data set is from Xi'an area. The data set is collected by RADARSAT-2 platform under C-band condition in 2009. It captures the scene of the western area of Xi'an City, Shanxi Province, China. Its size is 512 × 512. The PolSAR pseudo-color image of this data set is shown in Fig. 5(a). The ground truth map is shown in Fig. 5(b) and the legends of different labels are shown in Fig. 5(c). As can be seen from the figure, the data set contains 4 categories, including urban, grass, water and crop.

The number of training set samples and total samples of Xi'an PolSAR data set is shown in Table 4.

From Table 4 and Fig. 5(b), it can be seen that the number of crop samples is the smallest and mainly concentrated around

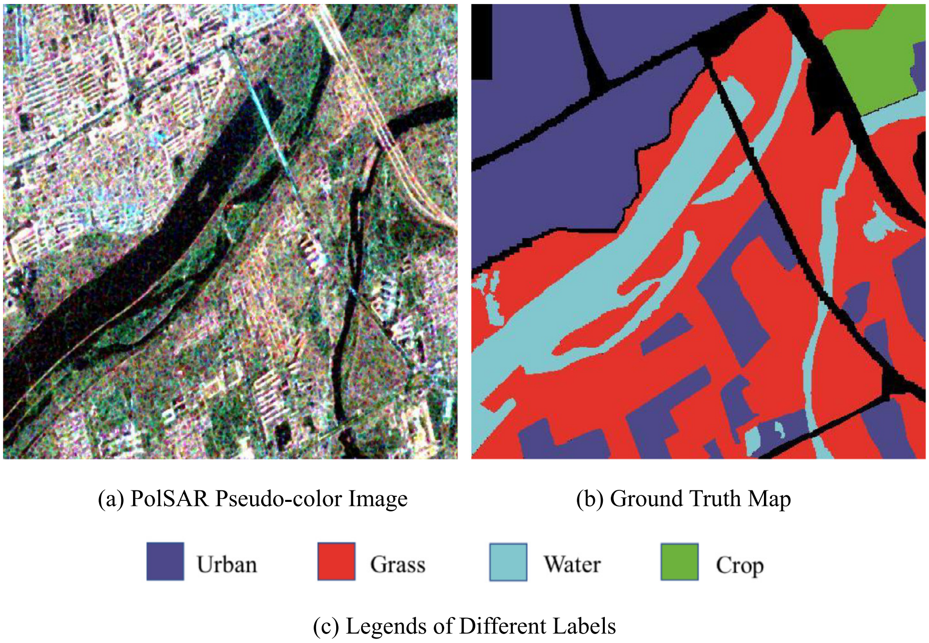
Table 2
Training sample number and total sample number of San Francisco PolSAR data set.

Number	Ocean	Forest	Low-density urban	High-density urban	Grassland
Training set number	1000	1000	1000	1000	1000
Testing set number	688 807	197 602	111 261	274 674	64 573
Total number	689 807	198 602	112 261	275 674	65 573


(a) (b) (c)

(d) (e) (f)

Fig. 4. Classification result maps of San Francisco PolSAR data set: (a) SF-CNN (b) CNN (c) CV-CNN (d) W-DBN (e) SVM.



(a) PolSAR Pseudo-color Image (b) Ground Truth Map

(c) Legends of Different Labels

Fig. 5. Xi'an PolSAR data set.

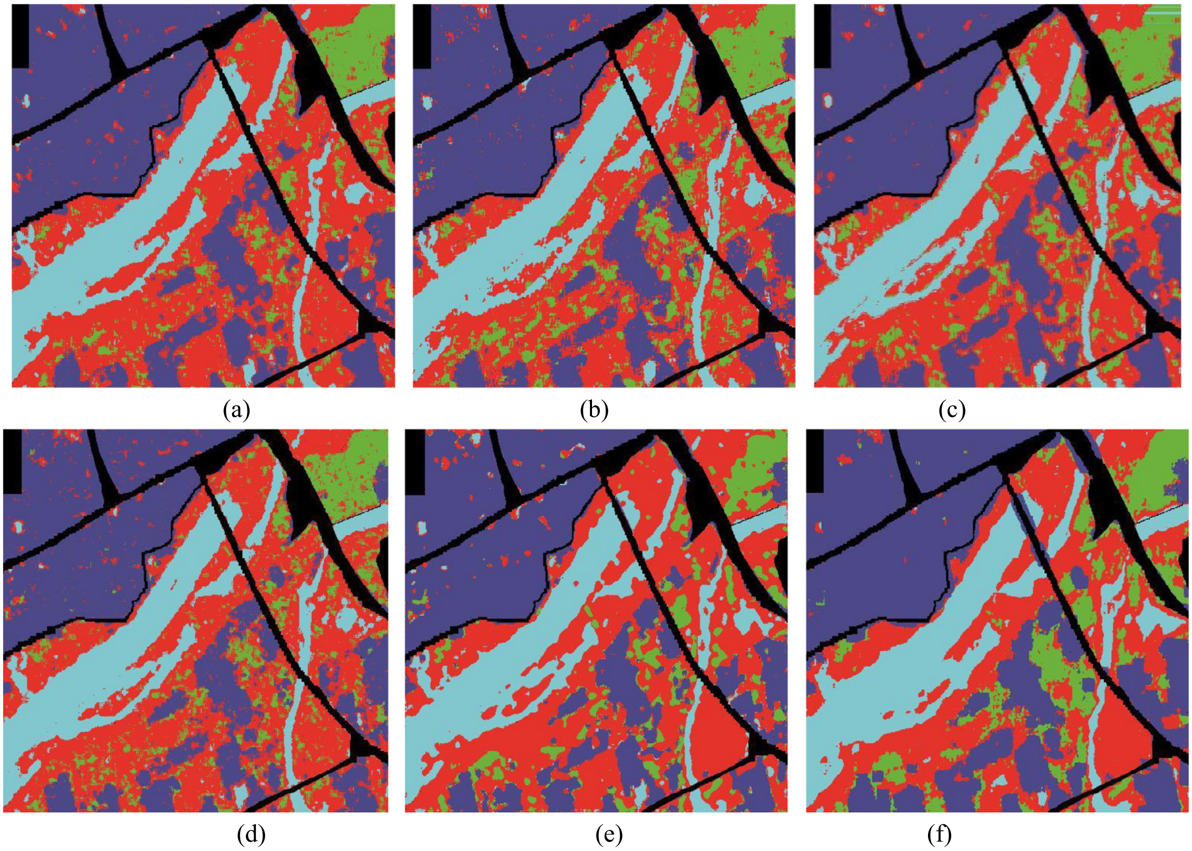


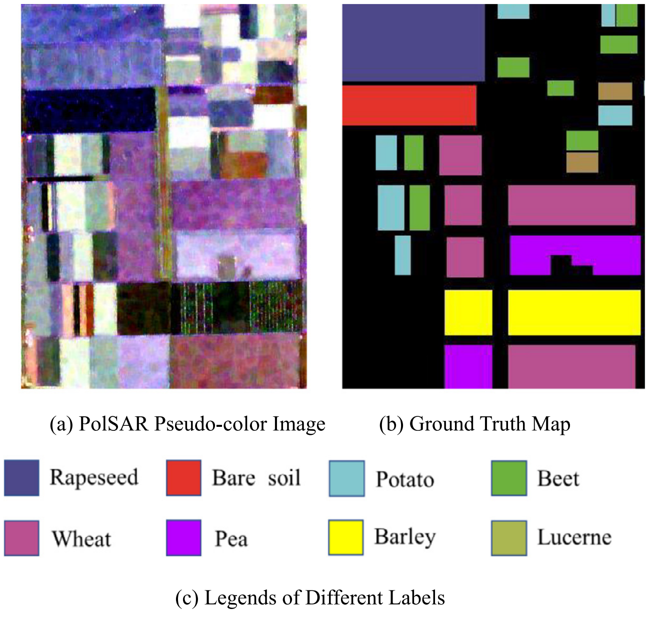
Fig. 6. Classification result maps of Xi'an PolSAR data set: (a) SF-CNN (b) CNN (c) CV-CNN (d) W-DBN (e) SVM.

Table 3
Classification results of San Francisco PolSAR data set.

Class	SF-CNN	VMW	CNN	CV-CNN	W-DBN	SVM
Ocean	0.9991	0.9999	0.9998	0.9999	0.9997	0.9935
Forest	0.9615	0.9498	0.946	0.9515	0.9477	0.9479
Low-density urban	0.9658	0.9741	0.9269	0.9288	0.9437	0.8915
High-density urban	0.9581	0.9566	0.9312	0.9498	0.9340	0.7598
Grassland	0.9773	0.9772	0.9457	0.9725	0.9373	0.9452
AA	**0.9724**	0.9715	0.9499	0.9605	0.9525	0.9076
OA	**0.9810**	0.9764	0.9686	0.9749	0.9706	0.9268
Kappa	**0.9715**	0.9601	0.9529	0.9623	0.9558	0.8915

Table 4
Training sample number and total sample number of Xi'an PolSAR data set.

Number	Urban	Grass	Water	Crop
Training set number	500	500	500	500
Testing set number	80 571	109 936	36 265	8651
Total number	81 071	110 436	36 765	9151



(a) PolSAR Pseudo-color Image (b) Ground Truth Map

(c) Legends of Different Labels

Fig. 7. Flevoland PolSAR data set.

the upper right corner of the ground truth map. While the sample numbers of other three classes are relatively adequate and geographically distributed. In order to better reflect the robustness of SF-CNN algorithm, cross validation is used to obtain the classification results. The specific method is to randomly select 500 different samples in each class as the training set and the remaining samples as the testing set. The average of the ten classification results is shown as the final result. Other algorithms do not use cross validation. The final results are shown in Table 5 and Fig. 6.

From Table 5, we can see that SF-CNN achieves the best results in AA, OA compared with standard CNN, CV-CNN, W-DBN and SVM. Although the VMW algorithm gets the highest value on kappa, the SF-CNN algorithm can still get higher kappa value than the other 4 algorithms.

As can be seen from the PolSAR pseudo-color image of Fig. 5(a), grass samples and crop samples in this data set are very close and easy to be confused. As can be seen from Table 5 and Fig. 6, the classification accuracies of standard CNN and CV-CNN for grass category are 70.14% and 73.58%, and for crop category

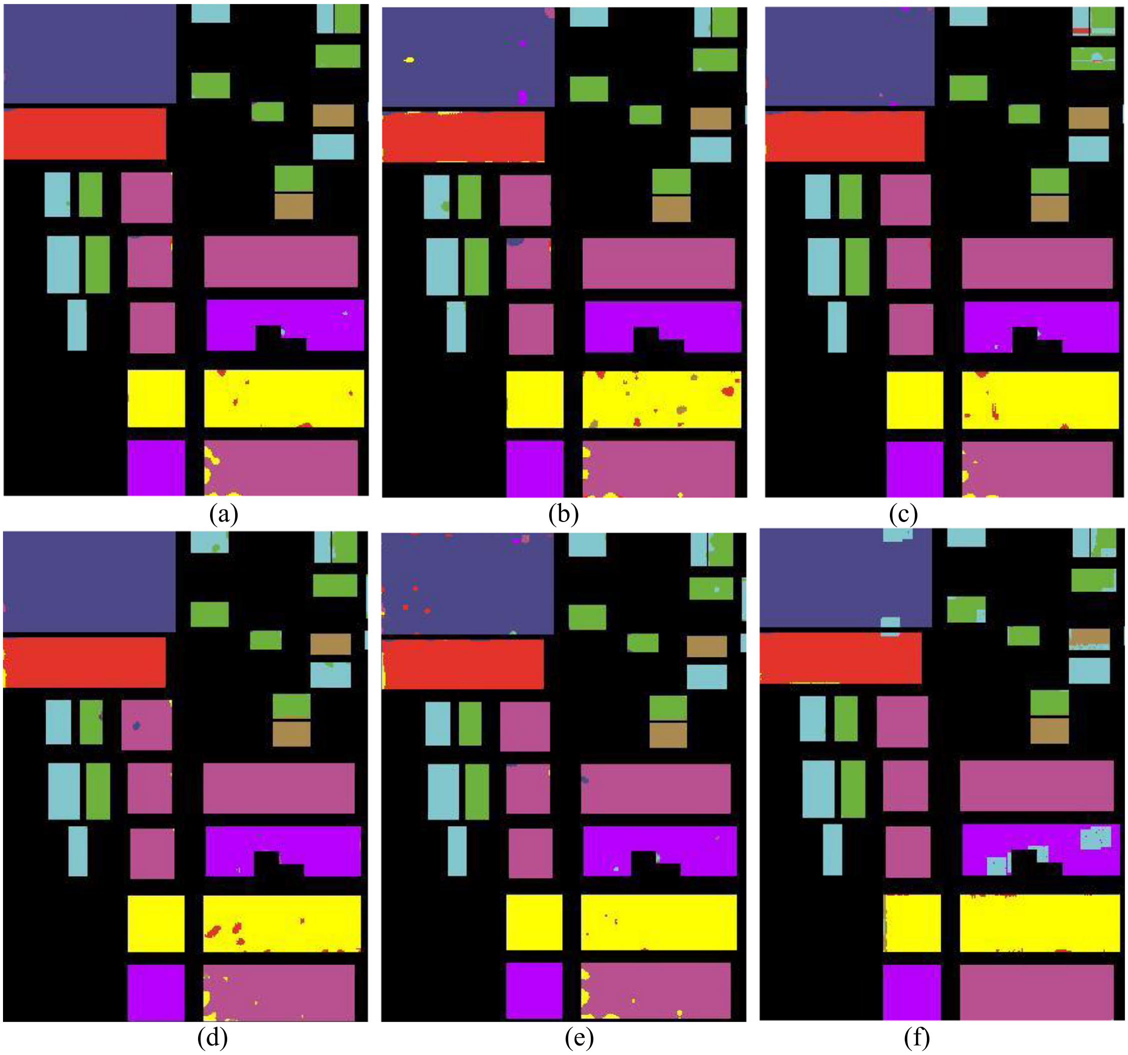


Fig. 8. Classification result maps of Flevoland PolSAR data set: (a) SF-CNN (b) CNN (c) CV-CNN (d) W-DBN (e) SVM.

Table 5
Classification results of Xi'an PolSAR data set.

Class	SF-CNN	VMW	CNN	CV-CNN	W-DBN	SVM
Urban	0.9212	0.9362	0.9154	0.8977	0.9034	0.9559
Grass	0.7333	0.7158	0.7014	0.7358	0.7282	0.6075
Water	0.9368	0.9137	0.9324	0.9368	0.9129	0.9061
Crop	0.9059	0.8519	0.9090	0.8394	0.5814	0.7033
AA	**0.8743**	0.8544	0.8646	0.8524	0.7814	0.7932
OA	**0.8347**	0.8257	0.8174	0.8257	0.8107	0.7746
Kappa	0.7551	**0.8102**	0.7318	0.7405	0.7165	0.6724

they are 90.90% and 83.94%. Obviously, standard CNN achieves better results than CV-CNN in crop classification, but it performs poorly in grass classification. On the contrary, CV-CNN performs well in grass but worse in crop. However, neither CNN nor CV-CNN can achieve good results in both crop and grass classification. The accuracy of SF-CNN in grass and crop is 77.55% and 90.90% respectively, which both are the best among the algorithms. It shows that SF-CNN has the best performance to distinguish grass and crop. SF-CNN has strong ability to distinguish similar terrain objects, which also proves the effectiveness of the special structure of SF-CNN.

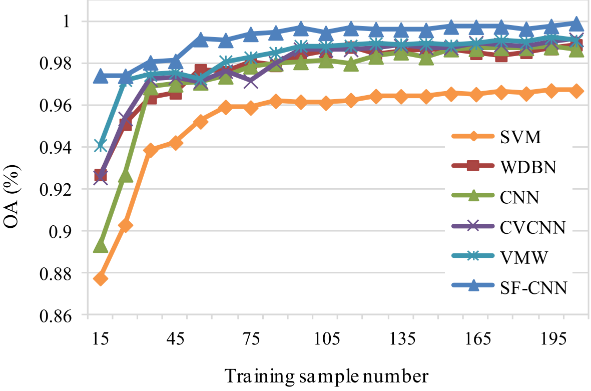


Fig. 9. The influence of training sample number on the overall accuracy of SF-CNN.

3.4. Evaluation and analysis in Flevoland PolSAR data set

The third experimental PolSAR data are captured in the farmland areas of Flevoland, Netherlands. During the MAESTRO-1

Table 6
Training sample number and total sample number of Flevoland PolSAR data set.

Number	Rapeseed	Bare soil	Potato	Beet	Wheat	Pea	Barley	Lucerne
Training set number	100	100	100	100	100	100	100	100
Testing set number	11 800	5100	4194	4150	14 500	6245	7550	1044
Total number	11 900	5200	4294	4250	14 600	6345	7650	1144

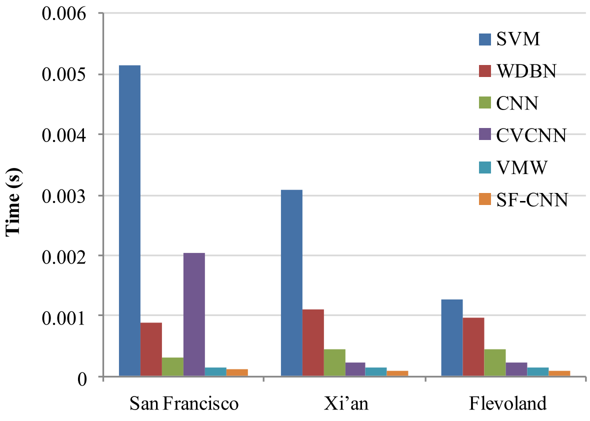


Fig. 10. Time of each algorithm to process one sample.

Table 7
Classification results of Flevoland PolSAR data set.

Class	SF-CNN	VMW	CNN	CV-CNN	W-DBN	SVM
Rapeseed	0.9988	0.9952	0.9842	0.9988	0.9817	0.9663
Bare soil	0.9972	0.9843	0.9703	0.9899	0.981	0.9813
Potato	0.9950	0.9859	0.9816	0.9766	0.9950	0.9990
Beet	0.9922	0.9540	0.9959	0.9745	0.9723	0.9157
Wheat	0.9946	0.9933	0.9836	0.9866	0.9829	0.9999
Pea	0.9962	0.9967	0.9980	0.9935	0.9935	0.8752
Barley	0.9950	0.9855	0.9618	0.9758	0.9970	0.9732
Lucerne	0.9993	1.0000	1.0000	0.9965	0.9965	0.8733
AA	**0.9961**	0.9868	0.9844	0.9865	0.9875	0.9480
OA	**0.9958**	0.9886	0.9822	0.9872	0.9861	0.9635
Kappa	**0.9950**	0.9863	0.9786	0.9846	0.9833	0.9561

project in 1989, NASA Jet Propulsion Laboratory used the AIRSAR platform to collect the data under L-band, fully polarized conditions. The shooting resolution is 6.6 × 12.10 m. The PolSAR pseudo-color image of the data set is shown in Fig. 7(a), and there are many different kinds of crops in the figure. The characteristics of each crop are quite obvious and easy to distinguish. The size of the image is smaller than the former two data sets, it is 300 × 400. Its ground truth map is shown in Fig. 7(b) and the label legends are shown in Fig. 7(c). Each color represents a category, and black color represents the unlabeled category. As can be seen from the figure, this data set contains 8 categories, including rapeseed, bare soil, potato, beet, wheat, peas, barley and lucerne.

The numbers of training samples and total samples for each class of Flevoland PolSAR data set are shown in Table 6.

In order to better reflect the robustness of SF-CNN algorithm, cross validation is used to obtain the classification results. The specific method is to randomly select 100 different samples in each class as the training set and the remaining samples as the testing set. The average of the ten classification results is shown as the final result. Other algorithms do not use cross validation. The final results are shown in Table 7.

As can be seen from Table 7, SF-CNN achieves the best results compared with other algorithms, with AA, OA and Kappa reaching 99.61%, 99.58% and 0.995 respectively. SF-CNN also achieves more than 99% accuracy in each class. The final results of different algorithms are shown in Fig. 8.

As can be seen from Fig. 8, compared with other algorithms, SF-CNN only makes classification errors at the edges of each pixel blocks. The mistakes mainly occurred when distinguishing wheat and barley. SF-CNN, CNN and CV-CNN are easy to confuse wheat with barley. All of the three CNN based algorithms achieve relatively good result on wheat category, and their wheat accuracy is all higher than 98%. When classing barley, SF-CNN's barley accuracy also keeps higher than 99%. While for CNN and CV-CNN, their barley accuracy dropped to 96.18% and 97.58%. The best algorithm for wheat classification is SVM, but the disadvantage of SVM is also very clear. It has only about 87% accuracy in pea and Lucerne categories, which is far lower than other deep learning algorithms. The possible reason maybe be that the fitting ability of SVM algorithm is not strong enough, which lead to an imprecise classification result. Generally speaking, the performance of SF-CNN is better than others.

In order to analyze the influence of the training sample number on the network, we tested the overall accuracy of all algorithms under different training sample numbers as shown in Fig. 9.

As can be seen from Fig. 9, when the training sample number is between 15 and 60, OA rises rapidly. Especially when using 65 training samples, the OA of SF-CNN has been up to 99.1%. It can be seen that SF-CNN is very efficient in the use of samples. After the training sample number more than 80, the OA of SF-CNN is above 99% and increases slowly. Therefore, it can be drawn some conclusions. When the training sample number is relatively small, providing more samples is conducive to the rapid improvement of the network. When the number of samples is sufficient, the network has no obvious improvement by continually increasing the training sample number. Because the work of labeling data is labor-intensive, a reasonable number of training samples is very important.

3.5. Time execution needed to process from certain number of samples

This section will analyze the running time of each algorithm. Fig. 10 shows the comparison result of the time required for each algorithm to process one sample.

The average time required for processing a certain number of samples in the running process of 6 algorithms is tested. As can be seen from Fig. 10, the maximum time required by all algorithms to process a single sample is in the order of 10^{-3}, and most of them are in the order of 10^{-4}. With the increase of the size of the datasets, the time required for SVM algorithm and CV-CNN algorithm increases significantly. The time of W-DBN algorithm, CNN algorithm, VMW algorithm and SF-CNN algorithm needed to process a single sample in the process of data scale increasing is more stable. And we can see that SF-CNN algorithm needs the least time to process a single sample in three kinds of datasets.

3.6. Analysis of threshold α

A threshold α is set in Eq. (12) to increase the distance between features of different classes until the distance is greater than α. This loss calculation method limits the Euclidean distance of different classes of features to be between $(0, \alpha)$. When the distance exceeds α, that is, the features of different classes are

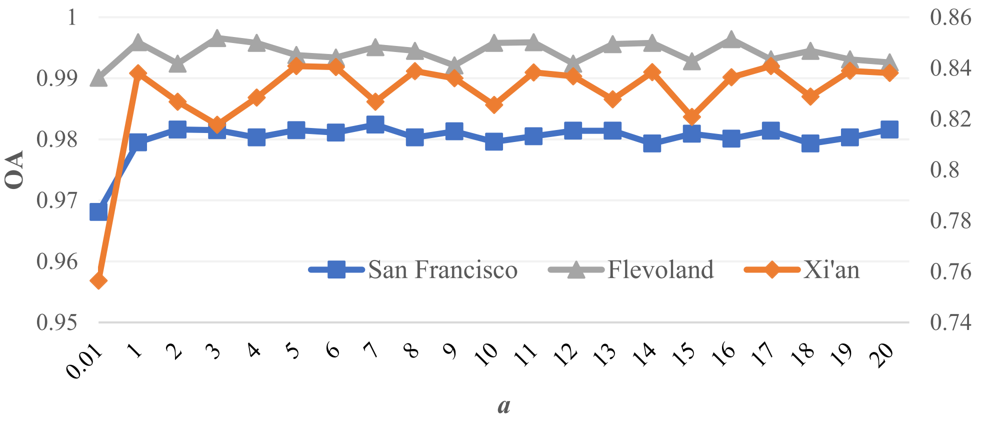


Fig. 11. Classification accuracy OA obtained on three datasets with different α values.

Table 8
Comparison results of two structures running on San Francisco dataset for 10 times.

San Francisco	SF-CNN_ave	Trinal-branch_ave	SF-CNN_max	Trinal-branch_max
Ocean	0.9991	**0.9996**	0.9997	**1.0000**
Forest	0.9615	**0.9696**	0.9643	**0.9644**
Low-density urban	0.9658	**0.9683**	0.9692	**0.9755**
High-density urban	**0.9581**	0.9561	0.9641	**0.9653**
Grassland	0.9773	**0.9782**	0.9801	**0.9831**
AA	0.9724	**0.9743**	**0.9733**	0.9729
OA	0.9810	**0.9827**	0.9822	**0.9823**
Kappa	**0.9715**	0.9711	0.9733	**0.9735**
Time (s)	**461.37**	643.42	**480.52**	679.89

Table 9
Comparison results of two structures running on Xi'an dataset for 10 times.

Xi'an	SF-CNN_ave	Trinal-branch_ave	SF-CNN_max	Trinal-branch_max
Urban	**0.9212**	0.9104	**0.9313**	0.9212
Grass	0.7333	**0.7422**	0.7654	**0.7728**
Water	0.9368	**0.9415**	0.9452	**0.9544**
Crop	0.9059	**0.9185**	0.9204	**0.9245**
AA	0.8743	**0.8782**	0.8798	**0.8932**
OA	0.8347	**0.8352**	0.8461	**0.85**
Kappa	**0.7551**	0.7536	0.7704	**0.7754**
Time (s)	**479.67**	616.28	**487.21**	625.87

far away, the loss is very low. Fig. 11 shows the results of classification accuracy OA obtained on three datasets with different α values.

In order to better show the results, the double ordinates are used in Fig. 11. It can be seen from Fig. 11 that on the three datasets, α varies from 1 to 20, which has no significant effect on the accuracy of classification results. However, when $\alpha = 0.01$, that is, when the value of α is small to a certain extent, the OA value is low. Therefore, the penalty range of Euclidean distance is not so important when the value of α is small to a certain extent. So the α can be selected randomly in the range of more than 1 and less than 20. The value selected in this paper is 5.

3.7. The results of trinal-branch CNN

A dual-branch parallel CNN structure is introduced in this paper, which can reduce the parameters and accept multiple groups of samples as input to realize sample expansion. This structure can be extended to three branches and more, but at the same time of improving the accuracy, it brings about the increase of calculation cost. In this section, the results of the trinal-branch CNN structure are compared with the original dual-branch structure. Triplet loss was used for loss calculation [41]. The triplet loss in paper is calculated as Eq. (13).

$$TLoss = \sum_{i}^{N} \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 + \beta \right] \tag{13}$$

where (x^a, x^p, x^n) represents a set of data of anchor, positive and negative. β is a constant, which is assigned as 1 in this experiment. Through learning, the distance between classes is greater than that within classes.

Tables 8–10 shows the maximum, average results and time required for the two structures to run 10 times on three datasets.

It can be seen from Tables 8–10 the classification accuracy of trinal-branch CNN network on most classes is improved to some extent, but the gap is not particularly obvious. At the same time, due to the increase of calculation cost, the running time of the algorithm also increases. Therefore, the establishment of multi-branch network should be selected according to different requirements.

4. Conclusion

In order to overcome the limitation of high-quality labeled data in PolSAR data, we propose a spatial feature-based CNN model (SF-CNN). The network can accept multiple samples as input at the same time, which can greatly expand the original

Table 10
Comparison results of two structures running on Flevoland dataset for 10 times.

Flevoland	SF-CNN_ave	Trinal-branch_ave	SF-CNN_max	Trinal-branch_max
Rapeseed	0.9988	**0.9996**	**1.0000**	**1.0000**
Bare soil	**0.9972**	0.9953	**0.9998**	0.9986
Potato	0.9950	**0.9953**	**0.9997**	**0.9997**
Beet	0.9922	**0.9937**	0.9969	**0.9972**
Wheat	0.9946	**0.9955**	0.9981	**0.9999**
Pea	0.9962	**0.9969**	**1.0000**	**1.0000**
Barley	**0.9950**	0.9934	**0.9988**	0.9986
Lucerne	**0.9993**	0.9992	**1.0000**	**1.0000**
AA	**0.9961**	**0.9961**	0.9976	**0.9993**
OA	**0.9958**	0.9943	0.9969	**0.9976**
Kappa	**0.9950**	0.9931	0.9963	**0.9971**
Time (s)	**474.62**	669.28	**483.36**	680.97

data set. SF-CNN adopts the dual-branch CNN structure, and the two CNN branches share training parameters. Each CNN is a full convolutional neural network, which removes the full-connected layers. This special structure can greatly reduce the parameters of network and decrease the risk of overfitting. When training, SF-CNN maximizes the distance of different-class feature centers and minimize the distance of same-class feature centers, which makes the features extracted by SF-CNN more discriminative. During the test process, SF-CNN used KNN to model the training sample features, which exploit more sample information. SF-CNN was tested on three PolSAR datasets and compared with other algorithms. Experimental results show that SF-CNN is better than standard CNN and other algorithms in indexes of AA, OA and Kappa. We can see that SF-CNN really has some advantages for its special structure. However, the SF-CNN uses the Euclidean distance. This kind of artificial measurement function is more or less defective. In the future work, we will try to use neural network to automatically learn the measurement function. When the feature centers of the two branches of CNN are calculated, the two feature centers are directly input into the neural network for learning. This automatic learning method would be better adapt to different data and learn more appropriate similarity measurement function. And the whole process can be "end-to-end", which will be very efficient.

CRediT authorship contribution statement

Ronghua Shang: Editing and revising the article, Supervision, Project administration. **Jiaming Wang:** Methodology, Investigation, Drafting the article. **Licheng Jiao:** Supervision, Project administration, Funding acquisition. **Xiaohui Yang:** Validation, Revising the article. **Yangyang Li:** Resources, Revising the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to express our sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Natural Science Foundation of China under Grants Nos. 62176200, 61773304, and 61871306, the Program for Cheung Kong Scholars and Innovative Research Team in University, China under Grant IRT1170, the National Key R&D Program of China and the Guangdong Provincial Key Laboratory, China under Grant No. 2020B121201001, the Open Research Projects of Zhejiang Lab, China under Grant 2021KG0AB03.

References

[1] M. Bielecka, S. Porzycka-Strzelczyk, J. Strzelczyk, SAR images analysis based on polarimetric signatures, Appl. Soft Comput. 23 (2014) 259–269.

[2] H. Liu, S. Yang, S. Gou, S. Liu, L. Jiao, Terrain classification based on spatial multi-attribute graph using polarimetric SAR data, Appl. Soft Comput. 68 (2018) 24–38.

[3] E. Pottier, Dr. J.R. Huynen's main contributions in the development of polarimetric radar techniques and how the 'radar targets phenomenological concept' becomes a theory, in: Radar Polarimetry, 1993, pp. 72–85.

[4] W.L. Cameron, L.K. Leung, Feature motivated polarization scattering matrix decomposition, in: IEEE International Conference on Radar, 1990, pp. 549–557.

[5] A. Freeman, S.L. Durden, A three-component scattering model for polarimetric SAR data, IEEE Trans. Geosci. Remote Sens. 36 (3) (1998) 963–973.

[6] S.R. Cloude, E. Pottier, An entropy based classification scheme for land applications of polarimetric SAR, IEEE Trans. Geosci. Remote Sens. 35 (1) (1997) 68–78.

[7] J.R. Huynen, Physical reality of radar targets, in: Proc. SPIE Radar Polarimetry, 1993, pp. 86–96.

[8] E. Krogager, New decomposition of the radar target scattering matrix, Electron. Lett. 26 (18) (1990) 1525–1527.

[9] J.S. Lee, M.R. Grunes, T.L. Ainsworth, L.J. Du, D.L. Schuler, S.R. Cloude, Unsupervised classification using polarimetric decomposition and the complex wishart classifier, IEEE Trans. Geosci. Remote Sens. 37 (5) (1999) 2249–2258.

[10] F. Liu, L. Jiao, B. Hou, S. Yang, POL-SAR image classification based on Wishart DBN and local spatial information, IEEE Trans. Geosci. Remote Sens. 54 (6) (2016) 3292–3308.

[11] L. Jiao, F. Liu, Wishart deep stacking network for fast POLSAR image classification, IEEE Trans. Image Process. 25 (7) (2016) 3273–3286.

[12] W. Xie, L. Jiao, B. Hou, W. Ma, J. Zhao, S. Zhang, F. Liu, POLSAR image classification via Wishart-AE model or Wishart-CAE model, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (8) (2017) 3604–3615.

[13] A.C. Frery, H.J. Muller, C.C.F. Yanasse, S.J.S. Sant'Anna, A model for extremely heterogeneous clutter, IEEE Trans. Geosci. Remote Sens. 35 (3) (1997) 648–659.

[14] S.H. Yueh, J.A. Kong, J.K. Jao, R.T. Shin, L.M. Novak, K-distribution and polarimetric terrain radar clutter, J. Electromagn. Waves Appl. 3 (8) (1989) 747–768.

[15] D. Zhang, L. Jiao, X. Bai, S. Wang, B. Hou, A robust semi-supervised SVM via ensemble learning, Appl. Soft Comput. 65 (2018) 632–643.

[16] C. Lardeux, P.L. Frison, C. Tison, J.C. Souyris, B. Stoll, B. Fruneau, J.P. Rudant, Support vector machine for multifrequency SAR polarimetric data classification, IEEE Trans. Geosci. Remote Sens. 47 (12) (2009) 4143–4152.

[17] C. He, M. Liu, Z.X. Liao, B. Shi, X.N. Liu, X. Xu, M.S. Liao, A learning-based target decomposition method using Kernel KSVD for polarimetric SAR image classification, EURASIP J. Adv. Signal Process. (1) (2012) 159.

[18] M. Jager, A. Reigber, O. Hellwich, Unsupervised classification of polarimetric SAR data using graph cut optimization, in: 2007 IEEE International Geoscience and Remote Sensing Symposium, 2007, pp. 2232–2235.

[19] Y. Wang, C. Han, F. Tupin, PolSAR data segmentation by combining tensor space cluster analysis and Markovian framework, IEEE Geosci. Remote Sens. Lett. 7 (1) (2010) 210–214.

[20] A.P. Doulgeris, An automatic U-distribution and Markov random field segmentation algorithm for PolSAR images, IEEE Trans. Geosci. Remote Sens. 53 (4) (2015) 1819–1827.

[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 770–778.

[24] G. Huang, Z. Liu, L.V.D. Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, 2017, pp. 2261–2269.

[25] B. Hou, H. Kou, L. Jiao, Classification of polarimetric SAR images using multilayer autoencoders and superpixels, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 9 (7) (2016) 3072–3081.

[26] Y. Zhou, H. Wang, F. Xu, Y.Q. Jin, Polarimetric SAR image classification using deep convolutional neural networks, IEEE Geosci. Remote Sens. Lett. 13 (12) (2016) 1935–1939.

[27] Z. Zhang, H. Wang, F. Xu, Y.Q. Jin, Complex-valued convolutional neural network and its application in polarimetric SAR image classification, IEEE Trans. Geosci. Remote Sens. 55 (12) (2017) 7177–7188.

[28] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[29] R. Shang, J. Wang, L. Jiao, R. Stolkin, B. Hou, Y. Li, SAR targets classification based on deep memory convolution neural networks and transfer parameters, IEEE J-STARS (99) (2018) 1–13.

[30] R. Shang, J. He, J. Wang, K. Xu, L. Jiao, Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification, Knowl.-Based Syst. (2020) 105542.

[31] F. Cakir, K. He, X. Xia, B. Kulis, S. Sclaroff, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1861-1870.

[32] J. Bromley, I. Guyon, Y. LeCun, E. Sickinger, R. Shah, Signature verification using a Siamese time delay neural network, in: 7th Annual Neural Information Processing Systems Conference, 1994, pp. 737–744.

[33] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4353–4361.

[34] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3630–3638.

[35] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4077–4087.

[36] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208.

[37] N.S. Altman, An introduction to kernel and nearest neighbor nonparametric regression, Am. Stat. 46 (3) (1992) 175–185.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[39] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Cogn. Model. 5 (3) (1988) 1.

[40] Q. Wu, B. Hou, Z. Wen, L. Jiao, Variational learning of mixture Wishart model for PolSAR image classification, IEEE Trans. Geosci. Remote Sens. 57 (1) (2019) 141–154.

[41] F. Schroff, D. Kalenichenko, J. Philbin, et al., FaceNet: A unified embedding for face recognition and clustering, Comput. Vis. Pattern Recognit. (2015) 815–823.