Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification

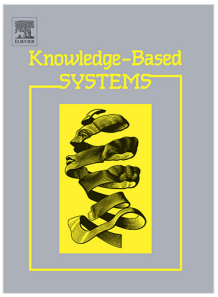
Shang, Ronghua; He, Jianghai; Wang, Jiaming; Xu, Kaiming; Jiao, Licheng; Stolkin, Rustam

DOI:
10.1016/j.knosys.2020.105542

Document Version
Peer reviewed version

Citation for published version (Harvard):
Shang, R, He, J, Wang, J, Xu, K, Jiao, L & Stolkin, R 2020, 'Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification', *Knowledge-Based Systems*, vol. 194, 105542. https://doi.org/10.1016/j.knosys.2020.105542

Journal Pre-proof



Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification

Ronghua Shang, Jianghai He, Jiaming Wang, Kaiming Xu,
Licheng Jiao, Rustam Stolkin

PII: S0950-7051(20)30041-1
DOI: https://doi.org/10.1016/j.knosys.2020.105542
Reference: KNOSYS 105542

To appear in: *Knowledge-Based Systems*

Received date : 31 August 2019
Revised date : 17 January 2020
Accepted date : 19 January 2020

Please cite this article as: R. Shang, J. He, J. Wang et al., Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.105542.

Dense Connection and Depthwise Separable Convolution Based CNN for Polarimetric SAR Image Classification

Ronghua Shang, *Member, IEEE*, Jianghai He, Jiaming Wang, Kaiming Xu, Licheng Jiao, *Fellow, IEEE*, Rustam Stolkin, *Member, IEEE*

***Abstract*—Convolution neural networks (CNN) have achieved great success in natural image processing where large amounts of training data are available. However, for the polarimetric synthetic aperture radar (PolSAR) image classification problem, the number of labeled training samples is typically limited. To improve the performance of CNN on limited training data, we propose a new network, the densely connected and depthwise separable convolutional neural network (DSNet). According to characteristics of PolSAR data, DSNet uses depthwise separable convolution to replace standard convolution, to independently extract features over each channel in PolSAR images. DSNet also introduces dense connections to directly connect non-adjacent layers. With the depthwise separable convolution and dense connections, DSNet can avoid extracting redundant features, reuse the hierarchical feature maps of PolSAR images and reduce the number of training parameters. Compared with normal CNN, DSNet is more lightweight and its training parameters decrease to less than 1/9. We compare DSNet against several popular algorithms on three different data sets, and show that DSNet achieves better results while using less training samples.**

***Index Terms*—DSNet, polarimetric SAR image classification, convolutional neural networks, depthwise separable convolution, dense connection.**

I. Introduction

PolSAR is a now mature and widely used technology, and has become one of the most important tools for earth observation. PolSAR is not affected by the weather and can work during both daytime and nighttime conditions. PolSAR obtains rich characteristics of objects on earth surface by using different polarimetric scattering modes. PolSAR is widely applied in agriculture, military reconnaissance, water area detection and resource exploration [1-3]. At present, the mainstream of PolSAR classification algorithms can be broadly divided into three categories.

The first category are conventional algorithms, which mainly consider the characteristics of PolSAR images. Some conventional algorithms are based on the scattering mechanism of PolSAR data, such as Cameron decomposition [4], Freeman decomposition [5], entropy/alpha (H-alpha) decomposition [6], Pauli decomposition [7] and so on. These algorithms are good at extracting the polarimetric scattering information of PolSAR images and have strong physical interpretability. Some other conventional algorithms are based on statistical distributions, mainly including Wishart distribution [8-9], K-distribution [10], G-distribution [11-12]. These methods predominantly describe PolSAR images by building various prior distributions.

The second category is machine learning algorithms, such as Support Vector Machine (SVM) [13-14], AdaBoost [15], Markov Random Field (MRF) [16-17] and so on. These algorithms are convenient to use, can be applied to different data sets, and have strong generality. However, such algorithms are often limited by their feature extraction and representation ability. It often fails to obtain some particularly satisfactory results when dealing with classification problem of complex scene image.

With the rapid development of deep learning algorithms, some deep learning methods have been proposed for PolSAR image classification. Hou *et al*. proposed a method which combines multi-layer autoencoders and superpixels to efficiently learn the features of PolSAR data [18]. Liu *et al*. used Wishart–Bernoulli restricted boltzmann machines to build a Wishart Deep Belief Network (W-DBN), and post-processed the classification result maps by local spatial information [19]. Combining Wishart distribution with deep learning, Jiao *et al*. proposed the Wishart Deep Stacking Network (W-DSN), which is formed by stacking several Wishart Networks [20]. Zhou *et al*. designed a four-layer CNN using a 6-D real feature vector as input to classify PolSAR images [21]. Zhang *et al*. designed a Complex-valued Convolution Neural Network (CV-CNN), and utilized both

This work was partially supported by the National Natural Science Foundation of China, under Grants 61773304, 61371201 and 61772399, and the Key Laboratory Fund under Grant 61421010402. Ronghua Shang, Jiaming Wang, Kaiming Xu and Licheng Jiao are with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an, Shanxi Province 710071, China. Rustam Stolkin is Director of the Extreme Robotics Lab, University of Birmingham, UK. (emails: rhshang@mail.xidian.edu.cn, tjnuwjm@gmail.com, 1735045518@qq.com, lchjiao@mail.xidian.edu.cn, R.Stolkin@cs.bham.ac.uk).

phase and amplitude information of PolSAR images to achieve a very good classification result [22].

Although the deep learning algorithms have stronger fitting ability than conventional algorithms and earlier forms of machine learning algorithms, deep learning models may be prone to overfitting, especially when using insufficient training data. Because of the limitation caused by the PolSAR imaging mechanism, the data obtained under different imaging conditions tend to be very different. This means that researchers often have to generate training and testing data sets from the labeled data in only a single PolSAR image. Therefore, the number of training samples may be very limited, which is not conducive to training a deep CNN.

In natural image datasets [23], large numbers of labeled training samples are often available, enabling a variety of CNN models to achieve excellent results [24], e.g. Alexnet [25], VggNet [26], GoogleNet [27]. Gao, Peng, et al. use learning reinforced attentional representation[28], and siamese attentional keypoint network in visual tracking[29]. Yu, Zeng, et al. add cross-layer neurons in convolutional networks for image recognition[30]. Later research proposed improved models, such as the depthwise separatable convolution to replace the standard convolution. The depthwise separable convolution is inspired by grouped convolution and inception modules. Representative networks include Xception networks [31] and MobileNet [32]. These networks cut down redundant training parameters and reduce the complexity of the model, without significantly reducing performance, and sometimes even with improved performance.

Researchers optimize models by changing the connection of networks. ResNet [33] introduced residual connection to alleviate the vanishing gradient and obtain a deeper model. DenseNet [34] proposed to use dense connection, of which each layer in a dense block is connected with all preceding layers. Dense connection can strengthen the networks' features and gradients propagation and reuse feature map information.

Inspired by Xception and DenseNet, considering the characteristics of PolSAR images, we propose a densely connected and depthwise separable convolutional neural network, DSNet, to enhance performance on PolSAR image classification problems involving limited training data sets. DSNet has the following key characteristics:

(1) All regular convolutions in DSNet are replaced by depthwise separable convolutions. The channel dimension of natural images is made up of R, G and B. In contrast, in this paper we work with PolSAR images, in which the channel dimension is made up of the PolSAR coherency matrix. The PolSAR coherency matrix contains the phase information and amplitude information of the image. Therefore, the spatial correlations of data are much larger than cross-channel correlations. If we used conventional convolution methods to extract features, we would need to extract fairly independent channel features, and highly correlated spatial features, simultaneously, and the extracted space-cross-channel features would be redundant. By using the depthwise separable convolution, the spatial correlations and cross-channel correlations can be extracted separately, and the extracted features are more efficient.

(2) Dense connection is used in DSNet. Dense connection has proven its effectiveness in the use of features on natural image datasets. Dense connection is conducive to repeatedly using features and improving the data information flow between layers. Because the feature maps are repeatedly used, the computation and training parameters of the network are decreased. In PolSAR data, training samples are very valuable, and important information can be greatly preserved by dense connections. In contrast to DenseNet [34], in DSNet we have extended dense connection to the pooling layer and can concatenate feature maps with different sizes.

(3) Both depthwise separable convolution and dense connections can substantially reduce the number of training parameters. Under the effects of these two operations, the network parameters and the complexity of the model are greatly reduced, while its generalization ability is enhanced. The parameter number of DSNet is much less than that of a standard CNN, which alleviates the risk of overfitting on small data sets.

To concluded, to achieve better performance on PolSAR image processing, CNN is considered. However, the number of labeled training samples is typically limited. So we use the dense connection technique and replace the common convolution with the depthwise separable convolution.

In this paper we compare DSNet and a standard CNN on several data sets. Our experimental results show that, under the same sampling rate, the performance of DSNet is better than that of a standard CNN in all three datasets.

The remaining parts of this paper are as follows. Section II details the specific architecture and method of DSNet. Section III presents experiments on several datasets, and analysis of the results. Section IV provides a summary and concluding remarks.

II. The Structure and Method of DSNet

This section presents details of the key technology and methods of DSNet, and how it is applied to the PolSAR image classification problem. The preparation of raw PolSAR data will be explained. The configuration of the DSNet network, training methods and parameter settings is given.

A. Decomposition of PolSAR data

Unlike conventional RGB images, each pixel in a full PolSAR image can be represented by a 2 × 2 complex scattering matrix *S*, as shown in equation (1):

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

where S_{pq} represent backscattering coefficients under different polarimetric combinations. p is the polarimetric mode of incident wave and q is the polarimetric mode of scattered wave. h and v represent the direction of electromagnetic wave; h is the horizontal direction, and v is the vertical direction. In monostatic radar case, according to the reciprocity theorem, $S_{hv}=S_{vh.}$, which means S is a symmetric matrix.

For obtaining the coherency matrix of PolSAR images, the scattering matrix S is vectorized, and the results after vectorization can be written as equation (2):

$$S = a\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + b\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} + c\frac{1}{\sqrt{2}}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \tag{2}$$

where a, b, c are the values of three different scattering components.

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

According to formula (3), the coherency matrix T of PolSAR images can be written as formula (4).

$$T = [a,b,c]^{\mathrm{T}}[a^*,b^*,c^*] = \begin{bmatrix}T_{11} & T_{12}^* & T_{13}^*\\ T_{12}^* & T_{22} & T_{23}^*\\ T_{13}^* & T_{23}^* & T_{33}\end{bmatrix} = \begin{bmatrix}|a|^2 & ab^* & ac^*\\ a^*b & |b|^2 & bc^*\\ a^*c & b^*c & |c|^2\end{bmatrix} \tag{4}$$

It can be seen from formula (4) that matrix T is a Hermitian matrix, whose diagonal value is real and whose off-diagonal value is complex. And "*" represents conjugate operation. Since matrix T is a symmetric matrix, we only need to extract a part data of T, $\{T_{11},T_{12}^*,T_{13}^*,T_{22},T_{23}^*,T_{33}\}$,which contains all information of T. $T_{12}^*,T_{13}^*,T_{23}^*$ are complex numbers. We extract the imaginary parts of $T_{12}^*,T_{13}^*,T_{23}^*$ and convert them into real values. Then we obtain a 9-dimensional real-valued vector T_v:

$$T_v = \{T_{11},T_{22},T_{33},re(T_{12}^*),re(T_{13}^*),re(T_{23}^*),im(T_{12}^*),im(T_{13}^*),im(T_{23}^*)\} \tag{5}$$

where $re(\cdot)$ and $im(\cdot)$ represent the real and imaginary parts of a complex number. Through the above operations, each pixel is transformed from a 2 × 2 matrix S into a 9-dimensional real-valued vector. For each training sample pixel p_x, a $N_u \times N_u$ neighborhood window centered at p_x is generated as the input feature map. It contains local spatial polarimetric information surrounding p_x.

To ensure the stability of the network, channel-wise normalization is performed on each pixel. The normalization equation can be written as (6):

$$T_v[j] = \frac{T_v[j] - T_{v-avg}[j]}{T_{v-std}[j]} \tag{6}$$

where $j \in (0, 1, \ldots, 9)$ and it represents 9 channels of T_v. $T_{v-avg}[j]$ and $T_{v-std}[j]$ are the average and standard deviation of jth channel of all training data respectively.

B. Network structure and dense connection

Our proposed DSNet consists of one input layer, one output layer, three depthwise separable convolutional layers, one max pooling layer and one fully connected layer. We introduce dense connection into DSNet on the basis of CNN. Dense connection can shorten the distance between the input layer and the output layer, which makes gradients and feature information propagate more fluently. Furthermore, when only a small amount of training data is available, dense connection also plays the role of a regularizing model, which decreases the risk of overfitting. The overall structure of DSNet is shown in Fig. 1.

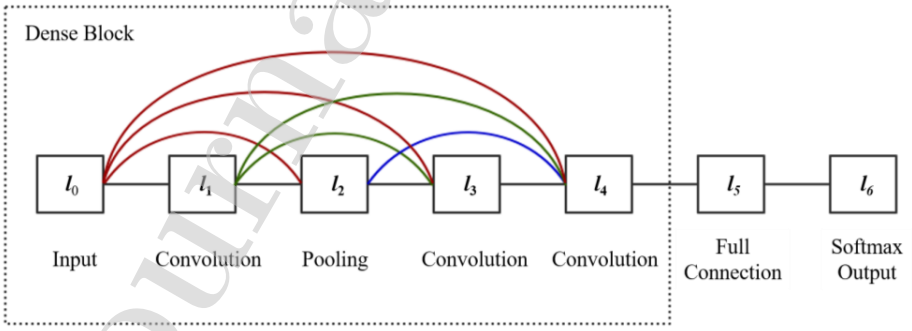


Fig. 1. The structure of DSNet. l_i represents the ith layer in the network.

As can be seen from Fig. 1, the top L (L=5) layers of the network form a dense block. The lth layer in the dense block obtains feature maps from all its preceding layers and pass its feature maps to all its subsequent L-l layers. Therefore, in the entire dense block, there are L×(L-1)/2 connections, instead of L connections as using traditional connections.

Input of one layer is a concatenation of the outputs from all previous layers. Assuming that x_l represents the input feature maps of the lth layer, and y_l represents the output feature maps of the lth layer, x_l can be calculated by formula (7):

$$x_l = Concat(y_0, y_1, \cdots, y_{l-1}) \tag{7}$$

where *Concat* $(\cdot)$ represents the concatenation of tensors along channel axis. In order to quickly reduce the size of the feature maps, valid padding is used in the convolutional layer. This means that the size of output feature maps in different layers is different. However, in DenseNet, each feature map should

have the same size because the concatenation is operated on the channel dimension. This is viable only when the size of feature maps does not change [34]. To solve this problem, we use the bilinear interpolation method to scale feature maps to the same size. With the resizing operation, DSNet extends dense connection to arbitrary-type layers instead of just convolutional layers, which can further enhance feature propagation and reduce the number of parameters compared with DenseNet.

Fig. 2 explains the concatenation operation between different-sized feature maps. Feature map F_1 and feature map F_3 have different size ($a_1 \times a_1$ and $a_3 \times a_3$). To connect F_1 and F_3, F_1 is first resized to $a_3 \times a_3$ (see F_1^*) and then undergoes a channel-wise concatenation with F_3.

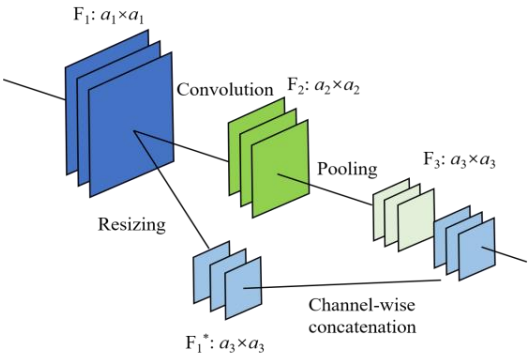


Fig. 2. Concatenation operation between different-sized feature maps.

We can obtain new output feature maps y_l' after the resizing operation, where $y_l' = Resize(y_l)$. The equation (7) can be rewritten as equation (8):

$$x_l = Concat(y_0', y_1', \ldots, y_{l-1}') \tag{8}$$

The bilinear interpolation method solves the concatenation problem between different size feature maps. Although it also loses some information, most information is still preserved. Consequently, our dense connection reuses the same feature maps multiple times and reduces the number of training parameters.

C. Depthwise separable convolution

The standard convolution operation extracts features from all three dimensions of each image, including two spatial dimensions (width and height) and one channel dimension. Therefore, a convolutional kernel needs to describe spatial correlations and cross-channel correlations simultaneously. This is written as:

$$\mathrm{Conv}(W,x)_{(i,j)} = \sum_{m,n,k}^{M,N,K} W_{(m,n,k)} \cdot x_{(i+m,j+n,k)} \tag{9}$$

where W is the weight matrix of convolutional kernels and is trainable. x is the input feature map of the convolutional layer, and (i, j) is the coordinate point of output feature maps. m, n and k are the 3 dimensions of the convolutional kernel. Depthwise separable convolution has been proven to be successful in neural image classification, it can avoid extracting some redundant features and considerably reduce the required paraments. In DSNet, the channel dimension of input data comprises the 9-dimensional vector T_v in formula (5). The channels have strong independence with each other. Depthwise separable convolution is more suitable than the common convolution to extract features in PolSAR images. In contrast to standard convolution, depthwise separable convolution divided the entire feature extraction into two simpler steps (a depthwise convolution and a pointwise convolution) [32]. The overall process is shown in Fig. 3:

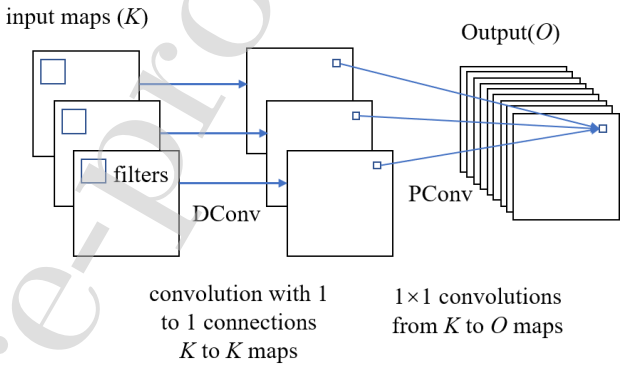


Fig. 3. Depthwise separable convolution.

The first step of depthwise separable convolution is a depthwise convolution. A single filter is applied to each input channel so that each channel can output one feature map. After depthwise convolution, the number of the channels does not change. The depthwise convolution can be written as formula (10):

$$\mathrm{DConv}(W,x)_{(i,j)} = \sum_{m,n}^{M,N} W_{(m,n)} \cdot x_{(i+m,j+n)} \tag{10}$$

In the second step, a 1×1 convolution (called a pointwise convolution) is applied to combine the outputs of the depthwise convolution. Pointwise convolution is used to perform the extraction of spatial features. This doesn't change the spatial size of feature maps but can change the channel number. Pointwise convolution operation is shown in formula (11):

$$\mathrm{PConv}(W,x)_{(i,j)} = \sum_{k}^{K} W_k \cdot x_{(i,j)} \tag{11}$$

For example, consider the case where we have two 3×3×3 standard convolutional kernels to do convolution with a 5×5×3 input map, to output a 3×3×2 feature map. If we use depthwise separable convolution, firstly, three 3×3×1 convolutional kernels apply convolution to each single channel of the input map generating three 3×3 feature maps. Secondly, two 1×1×3 convolution kernels are used to do convolution with the 3×3×3 feature map and get a 3×3×2 output map.

With formulas (10) and (11), the overall process of depthwise separable convolution can be written as:

$$\mathrm{SConv}(W_p, W_d, x)_{(i,j)} = \mathrm{PConv}(W_p, x)_{(i,j)}\,(W_p, \mathrm{DConv}(W_d, x)_{(i,j)}) \tag{12}$$

Compared with the standard convolution, the parameters of depthwise separable convolution are greatly reduced. If the number of output channel is *o*, for standard convolutional layers, according to the formula (9), the total required parameters are $m \times n \times k \times o$. While for depthwise separable convolutional layer, according to the formula (12), the total required parameters are $m \times n \times k + k \times o$. The ratio of these is:

$$\frac{m \times n \times k + k \times o}{m \times n \times k \times o} = \frac{1}{o} + \frac{1}{m \times n} \tag{13}$$

There are three depthwise separable convolutional layers in DSNet and the parameter settings are shown in TABLE I.

TABLE I
PARAMETER SETTINGS OF DSNET.

Num.	Type / Stride	Filter Shape	Input Size
0	Input	Input 15×15×9	15×15×9
1	DConv / s1	6×6×9	15×15×9
	PConv/s1	1×1×9×27	15×15×9
2	Max Pool / s2	Pool 2×2	10×10×[9+27]
3	DConv / s1	3×3×72	5×5×[9+27+36]
	PConv/s1	1×1×72×144	5×5×72
4	DConv / s1	3×3×216	3×3×[9+27+36+144]
	PConv/s1	1×1×216×216	1×1×216
5	dropout-FC / s1	216×15	1×1×216
6	Softmax / s1	Classifier	1×1×15

(*Table I: "[o_0+ o_1 +... +o_i]" in the "Input size" column represents concatenating the 0th, 1th, ..., *i*th layers' output feature map copies.)

With depthwise separable convolution and dense connection, parameters are used very efficiently. The time complexity of the standard convolution is $O\sim(M^2*K^2*C_{in}*C_{out})$, while that of the depthwise separable convolution is $O\sim(M^2*K^2*C_{in}+M^2*C_{in}*C_{out})$. M stands for the size of feature map, K means the size of kernel, C_{in} is the number of input channels and C_{out} is the number of output channels. The total number of parameters in DSNet is about 2.0M when using 32-bit float variables. In contrast, a CNN with similar structure has about 18.8M parameters.

D. Activation function and pooling operation

The convolutional layer can only achieve linear transformations. It is necessary to introduce activation for nonlinear transformations. The commonly used activation functions include sigmoid function, tanh function, rectified linear unit (ReLU) function etc. We choose the sigmoid function as the activation function. Sigmoid is a continuous and strictly monotonic function, whose output value is between (0, 1). The formula can be written as:

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{14}$$

The sigmoid function is only used after pointwise convolution, and there is no activation function after depthwise convolution.

A pooling operation can quickly decrease the dimension of feature maps, helping to reduce the number of layers that are needed in neural networks, and can introduce some nonlinear changes. The commonly used pooling operations include average pooling and max pooling. Max pooling is used in this paper. It can output the max value in a given local area and its pooling size is 2 × 2 with a stride of 1 [35-36].

E. Fully connected layer and dropout

Dropout [37] is a technique which can prevent overfitting and improve the generalization ability of neural networks. Dropout randomly deactivates some units during the training process, however the weight of these units is still retained for testing. A fully connected layer with dropout operation can be written as:

$$r \sim \mathrm{Bernoulli}(p_r)$$
$$\mathrm{FC}(W, x)_{(i,j)} = \sum_k^K W_k \cdot (r \cdot x_{(i,j)}) + B \tag{15}$$

where *r* is an independent Bernoulli variable. Its value is 1 with probability of p_r and 0 with probability of $1\text{-}p_r$. *B* is the trainable bias. The raw input *x* multiplies with *r* and the result of multiplying is input to fully connected layer. After fully connected layer, a softmax classifier is added to the network and it outputs the final probability of each class.

F. The training of DSNet

With the output of softmax classifier and ground truth labels, the cross-entropy loss can be calculated as our objective function. The objective function is always nonconvex making it hard to find the global optimum. Therefore, we use the back-propagation algorithm (BP) [38] to calculate the required partial derivative of training parameters, and use Adam [39] optimizer to update parameters. The Adam optimizer requires less memory and can adaptively change its learning rates. In the Adam optimizer, exponential decay rates are $\rho 1 = 0.9$ and $\rho 2 = 0.999$. The initialization values of *W* and *B* in depthwise separable convolutional layers and fully connected layers are important. They can affect the training performance of networks. We initialise *B* as 0. We use uniform initialization to randomly assign the values of *W*, as recommended by Glorot *et al.* [40]. The formula is shown in (16):

$$W \sim U(-\sqrt{\frac{6}{n_k + n_o}}, \sqrt{\frac{6}{n_k + n_o}}) \tag{16}$$

where $U(\cdot)$ represents the uniform distribution. n_k indicates the input number of each layer and n_o indicates the output number of each layer. The training learning rate is set to 0.001, batch size value is set to 128 and dropout value is 0.5.

III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we evaluate the performance of DSNet in three different real PolSAR data sets. Data set 1 and data set 2 were collected by Airborne SAR (AIRSAR). The scene covers agricultural area in Flevoland, Netherlands. Data set 3 was collected by the Electronically Steered Array Radar (ESAR)

covering the Oberpfaffenhofen area in Germanyt. These three data sets are often used as benchmark data sets to assess the performance of PolSAR classification algorithms. DSNet is compared with several machine learning and deep learning methods including SVM [14], Wishart-DBN [19], and CV-CNN [22]. To demonstrate the effectiveness of the novel structure of DSNet, Network A is based on a conventional CNN, Network B is based on DenseNet (using dense connection), Network C is based on Xception (using depthwise separable convolutions) and Network D (uses dense connection and depthwise separable convolution but doesn't extend dense connection to pooling layers) are all tested. A, B, C and D all share similar architectures with DSNet except for the key differences of DSNet that are described in Section II above. Overall accuracy (OA), average accuracy (AA) and Kappa (Ka) coefficient of different methods are quantitatively compared. In all experiments, the size of neighborhood window N_u is set to 15, namely, the input size is 15 × 15 × 9.

A. Evaluation and analysis on Flevoland dataset 1

The first dataset was obtained by the AIRSAR platform of NASA Jet Propulsion Laboratory on August 16, 1989. It covers the L-band four-look polarimetric sense of farmland in Flevoland, Netherlands, and its resolution is 6.6 × 12.10m. The pseudo color image can be generated by Pauli decomposition, as shown in Fig. 4(a). The ground truth map is illustrated in Fig. 4(b), as used in [41]. In Fig. 4(b), different colors represent different categories and black represents unlabeled image regions. There are 15 identified classes including peas, stem beans, three different kinds of wheat, lucerne, beet, rape seed, barley, potatoes, grass, bare soil, forest, water, and a small area of buildings.

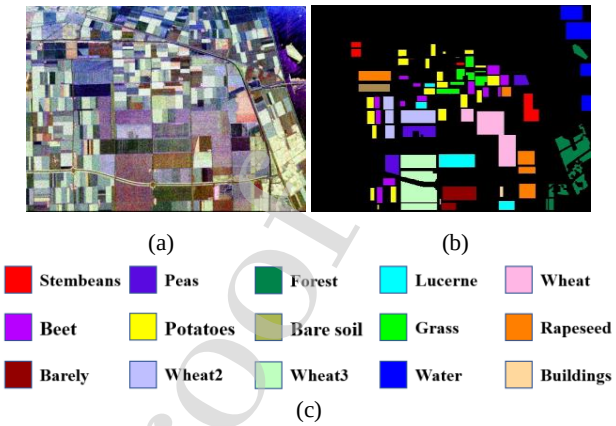


Fig. 4. Flevoland data set 1: (a) Pauli pseudo color image. (b) Ground truth map. (c) Legend of different classes.

Different PolSAR images are obtained for different scenes with different configuration parameters, such as polarimetric mode, resolution, electromagnetic wave band, etc. Therefore, most PolSAR classification algorithms use both training data and testing data derived from labeled samples in the same PolSAR image. The commonly used sampling rate is 5%-10%. Namely, 5%-10% of labeled samples are used as training sets, and the remaining 90%-95% labeled data are used as testing sets. It can be seen that the training sets are very limited in PolSAR classification problems. In our experiments, in order to conveniently compare with other algorithms, we measured the classification results of DSNet under a 1% sampling rate (very low sampling rate) and a 5% sampling rate (used in much of the literature). The lower sampling rate provides a difficult challenge for PolSAR classification algorithms. All algorithms are executed under the same conditions.

TABLE II
CLASSIFICATION RESULTS OF FLEVOLAND DATA SET 1.

Class	SVM	W-DBN	CV-CNN	Network A	Network B	Network C	Network D	DSNet (1%)	DSNet (5%)
Stembeas	0.9112	0.9882	0.9861	0.9892	0.9920	0.9813	0.9880	0.9749	0.9969
Peas	0.9153	0.9875	0.9827	0.9922	0.9887	0.9836	0.9891	0.9928	0.9973
Forest	0.9440	0.9915	0.9833	0.9778	0.9832	0.9867	0.9875	0.9945	0.9964
Lucerne	0.9426	0.9847	0.9804	0.9518	0.9647	0.9548	0.9645	0.9883	0.9969
Wheat	0.9499	0.9738	0.9703	0.9719	0.9817	0.9781	0.9749	0.9884	0.9959
Beet	0.9406	0.9577	0.9881	0.9678	0.9723	0.9633	0.9806	0.9827	0.9871
Potatoes	0.3913	0.9779	0.9808	0.9765	0.9801	0.9778	0.9812	0.9774	0.9959
Bare	0.4977	0.9984	1.0000	0.9734	0.9925	0.9256	1.0000	0.9971	0.9932
Grass	0.8556	0.9245	0.9357	0.9324	0.9442	0.8794	0.9379	0.9239	0.9848
Rapeseed	0.8204	0.9328	0.9441	0.9102	0.9392	0.9646	0.9420	0.9507	0.9835
Barely	0.9466	0.9575	0.9789	0.9655	0.9593	0.9691	0.9635	0.9788	0.9930
Wheat2	0.9378	0.9763	0.9656	0.9394	0.9341	0.9649	0.9840	0.9696	0.9854
Wheat3	0.9550	0.9930	0.9914	0.9892	0.9924	0.9939	0.9927	0.9942	0.9966
Water	0.7385	0.9999	0.9979	0.9987	0.9994	0.9985	0.9990	0.9977	0.9974
Buildings	0.5000	0.8508	0.8193	0.8550	0.8214	0.8067	0.8277	0.9832	0.9853
Sample rate	5%	5%	10%	1%	1%	1%	1%	1%	5%
AA	0.8164	0.9663	0.9670	0.9594	0.9630	0.9552	0.9676	**0.9796**	**0.9924**
OA	0.8582	0.9759	0.9775	0.9683	0.9742	0.9728	0.9781	**0.9816**	**0.9934**
Ka	0.8454	0.9238	0.9754	0.9654	0.9719	0.9703	0.9761	**0.9799**	**0.9928**

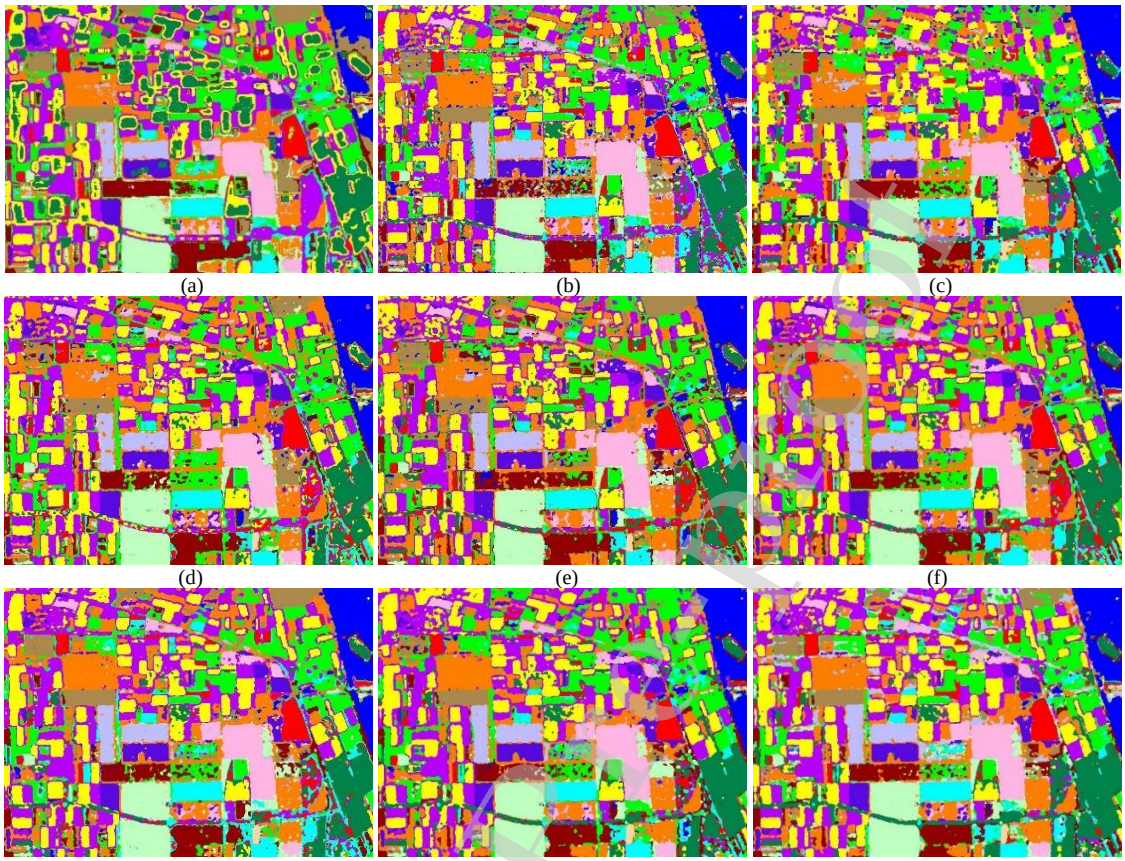


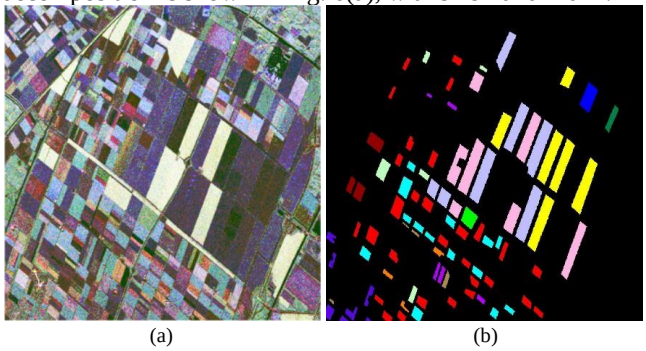
Fig. 5. Classification result maps of Flevoland data set 1: (a) SVM (b) W-DBN (c) CV-CNN (d) Network A (e) Network B (f) Network C (g) Network D (h) DSNet (1% sampling rate) (i) DSNet (5% sampling rate)

We have compared DSNet with several algorithms and the final results are shown in TABLE II, and segmentation result maps are shown in Fig. 5. The sampling rates of SVM, W-DBN and CV-CNN are 5%, 5% and 10% respectively.

Under 1% sampling rate, the values of AA, OA and Kappa of DSNet are all approximately equal to 0.98. Compared with above three algorithms, DSNet only uses 1/5 or 1/10 training samples but still generates better results. DSNet is also compared with Networks A, B, C and D under 1% sampling rate. DSNet's OA, AA and Kappa coefficient are all higher than B and C. This suggests that the combination of depthwise separation convolution and dense connection makes more difference in improving a networks' performance than either dense connection or depthwise separation convolution alone. Compared with DSNet and Network D, OA, AA and Kappa of DSNet are respectively 1.2 %, 0.35 % and 0.42 % higher, which demonstrates the effectiveness of extending dense connection to the arbitrary-typed (including pooling) layers. Some algorithms introduce post-processing methods to improve their classification accuracy. For example, W-DBN uses post-processing methods based on local information, and CV-CNN post-processes its classification maps by majority voting. DSNet can be seen as an end-to-end system and does not use any post-processing techniques. There are two main advantages to this approach. On the one hand, post-processing methods are specific to data. It is hard to design a post-processing method that is generalizable enough to give good results on all data sets. On the other hand, from TABLE II, it can be seen that under 5% sampling rate, DSNet has achieved more than 99% OA, and post-processing helps little to improve the final results.

B. Evaluation and analysis in Flevoland data set 2

The second Flevoland data was obtained by AIRSAR in 1991 under L-band. It is a fully polarimetric image of agricultural area in Flevoland. The pseudo color map by Pauli decomposition is shown in Fig. 6(a), with size 1020×1024.



(c)

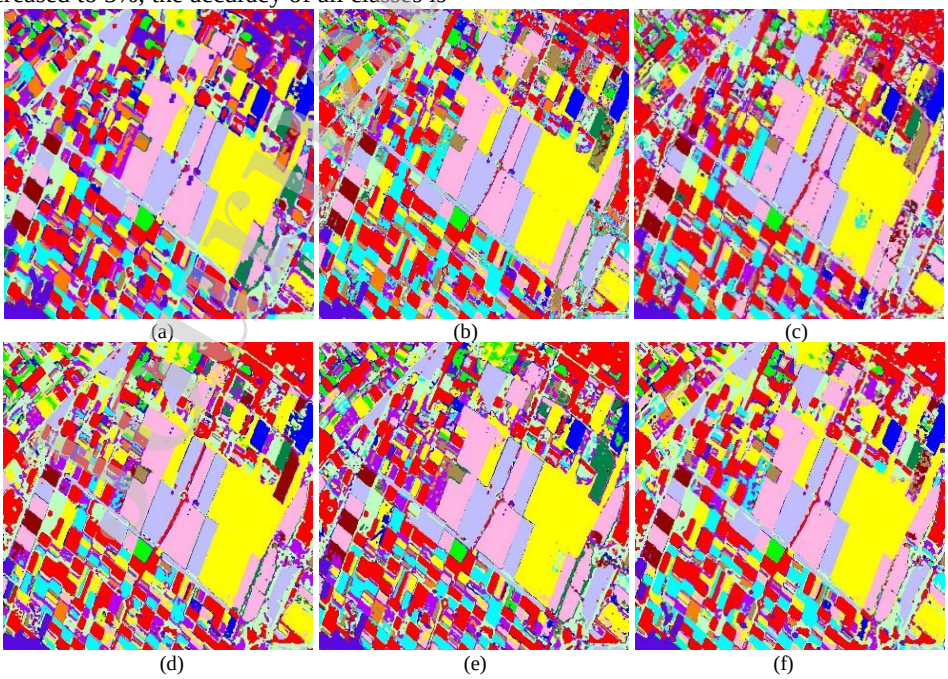
Fig. 6. Flevoland data set 2: (a) Pauli pseudo color image. (b) Ground truth map. (c) Legend of different classes.

It can be seen from the ground truth map that there are 14 categories in the data set, including potatoes, fruit, oats, beet, barely, onions, wheats, beans, peas, maize, flax, rapeseed, grass, lucerne. The AA, OA, and Kappa coefficient of different algorithms are shown in TABLE III.

TABLE III
CLASSIFICATION RESULTS OF FLEVOLAND DATA SET 2.

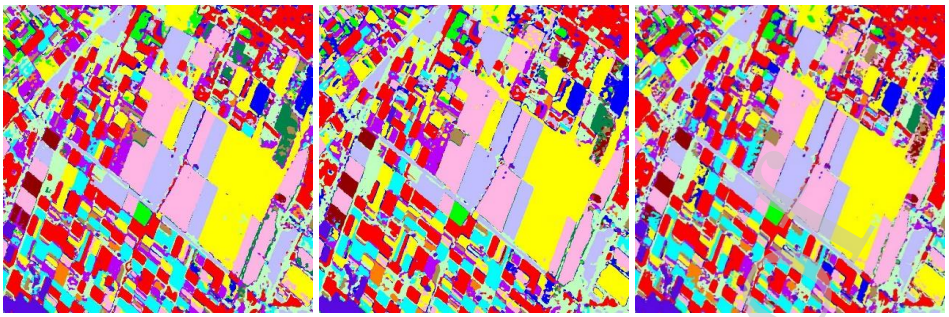
Class	SVM	W-DBN	CV-CNN	Network A	Network B	Network C	Network D	DSNet (1%)	DSNet (5%)
Potato	0.9969	0.9972	0.9974	0.9998	0.9967	0.9990	0.9972	0.9962	0.9998
Fruit	1.0000	1.0000	0.9828	0.9919	0.9899	0.9765	1.0000	1.0000	1.0000
Oats	1.0000	0.9813	0.9885	0.9964	0.9677	0.9871	0.9921	0.9821	0.9770
Beet	0.8964	0.9933	0.9606	0.8978	0.9417	0.9322	0.9528	0.9805	0.9936
Barely	0.9537	0.9969	0.9961	0.9914	0.9969	0.9959	0.9943	0.9941	0.9991
Onions	0.7277	0.6174	0.9319	0.9009	0.8000	0.8920	0.9512	0.9188	0.9803
Wheats	0.9988	0.9970	0.9990	0.9990	0.9988	0.9943	0.9942	0.9987	0.9995
Beans	0.7200	0.9510	0.9067	0.8688	0.9233	0.9529	0.8928	0.9344	0.9871
Peas	1.0000	0.9954	0.9903	0.9991	1.0000	0.9500	0.9731	0.9944	1.0000
Maize	0.9620	0.9558	0.9814	0.6868	0.8791	0.8085	0.8798	0.9109	0.9698
Flax	0.9986	0.9895	0.9695	0.9872	0.9886	0.9847	0.9984	0.9935	0.9970
Rapeseed	0.9990	0.9968	0.9981	0.9988	0.9985	0.9965	0.9976	0.9998	0.9999
Grass	0.8352	0.8694	0.9660	0.9841	0.9512	0.9441	0.9715	0.9810	0.9955
Lucerne	0.9885	0.9356	0.9922	0.9268	0.9593	0.9739	0.9488	0.9817	0.9834
Sample rate	5%	5%	10%	1%	1%	1%	1%	1%	5%
AA	0.9341	0.9483	0.9758	0.9449	0.9566	0.9563	0.9674	**0.9762**	**0.9916**
OA	0.9700	0.9843	0.9902	0.9814	0.9854	0.9835	0.9878	**0.9923**	**0.9976**
Ka	0.9647	0.9748	0.9884	0.9781	0.9828	0.9806	0.9856	**0.9910**	**0.9972**

With 1% sampling rates, DSNet's OA and Kappa coefficient are above 0.99. All single class accuracy rates are higher than 0.98 except onions, beans and maize. Notably, the accuracy rates of fruit are up to 100%. The final segmentation maps are shown in Fig. 7. For DSNet, the result under only 1% sampling rates for training data is still good. When the sampling rate is increased to 5%, the accuracy of all classes is increased, and several classes achieve the best results. While compared with 1% sampling rate, 5% sampling brings relatively little improvement. This suggests that more training samples does not bring great performance improvements, and 5% sampling rate reaches the saturation state, beyond-which additional labeled training samples are not useful. An appropriate choice of sampling rate is important.



(a) (b) (c)

(d) (e) (f)



(g) (h) (i)

Fig. 7. Classification result maps of Flevoland data set 2: (a) SVM (b) W-DBN (c) CV-CNN (d) Network A (e) Network B (f) Network C (g) Network D (h) DSNet (1% sampling rate) (i) DSNet (5% sampling rate)

We next analyze the influence of sampling rates on DSNet's OA. The relationship between OA and sampling rates is shown in Fig. 8.

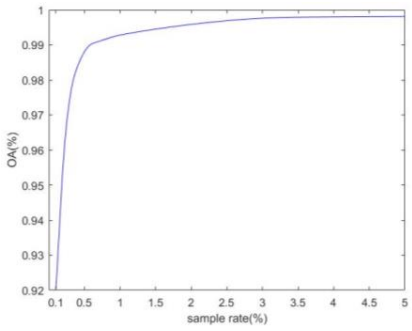
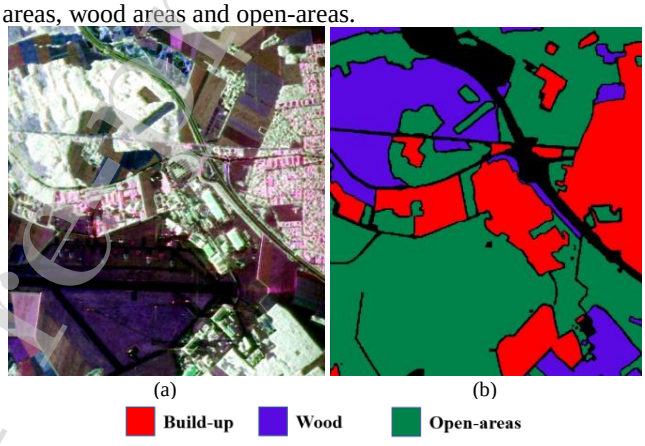


Fig. 8. OA under different sampling rates

From Fig. 8, it can be seen that DSNet has reached 92% OA when using 0.1% sampling rate. And OA rapidly increases to 99% when the sampling rate approaches 0.6%. When the sampling rate grows from 0.6% to 3%, OA more gradually rises to 99.7%. When the sampling rate is higher than 3%, OA does not show appreciable further improvement. Therefore, 0.6%-3% sampling rate is suitable in this data set. Compared with other algorithms, fewer labeled samples are needed to train DSNet to a high accuracy. This highlights DSNet's efficiency in the use of training data.

C. Evaluation and analysis in Oberpfaffenhofenin data set

In this experiment, the PolSAR image is L-band and collected by ESAR in the Oberpfaffenhofenin area of Germany. The pseudo color map generated by Pauli decomposition is shown in Fig. 9(a), and its size is 1300 × 1200, which is bigger than the previous two Flevoland data sets. The ground truth map is described in Fig. 9(b), which is gleaned from [43]. The data set totally contains 3 different types of land, including built-up areas, wood areas and open-areas.



(a) (b)

(c)

Fig. 9. Oberpfaffenhofenin data set: (a) Pauli pseudo color image. (b) Ground truth map. (c) Legend of different classes.

There are only three classes in the Oberpfaffenhofenin data set, which is much less than the previous two data sets. However, this image has more complex terrain. As can be seen in Fig. 9(a), some areas belonging to different classes are very similar and easy to confuse. This causes severe difficulties to accurately classify every pixel in the data set. The test results are shown in TABLE IV.

The performance of all the compared algorithms decreases compared with the previous two Flevoland data sets. However, DSNet still achieves the best result. At 1% sampling rate, the OA of DSNet is 2.39%, 2.20%, 1.13%, 1.69%, 0.75% higher than CV-CNN, A, B, C and D respectively. These results suggest that DSNet is better able to deal with complex data sets. Classification maps are shown in Fig. 10.

TABLE IV
CLASSIFICATION RESULTS OF OBERPFAFFENHOFENIN DATA SET.

Class	SVM	W-DBN	CV-CNN	Network A	Network B	Network C	Network D	DSNet (1%)	DSNet (5%)
Build-up	0.7928	0.8688	0.8667	0.8944	0.8396	0.8110	0.8869	0.9278	0.9730
Wood	0.9058	0.8955	0.9280	0.9731	0.9654	0.9691	0.9492	0.9557	0.9880
Open-areas	0.9694	0.9860	0.9645	0.9838	0.9868	0.9882	0.9780	0.9708	0.9919
Sample rate	5%	5%	1%	1%	1%	1%	1%	1%	5%
AA	0.8893	0.9168	0.9197	0.9235	0.9306	0.9228	0.9380	**0.9514**	**0.9843**
OA	0.9134	0.9397	0.9334	0.9353	0.9460	0.9404	0.9498	**0.9573**	**0.9864**
Ka	0.8504	0.8804	0.8861	0.8895	0.9067	0.8969	0.9139	**0.9272**	**0.9769**

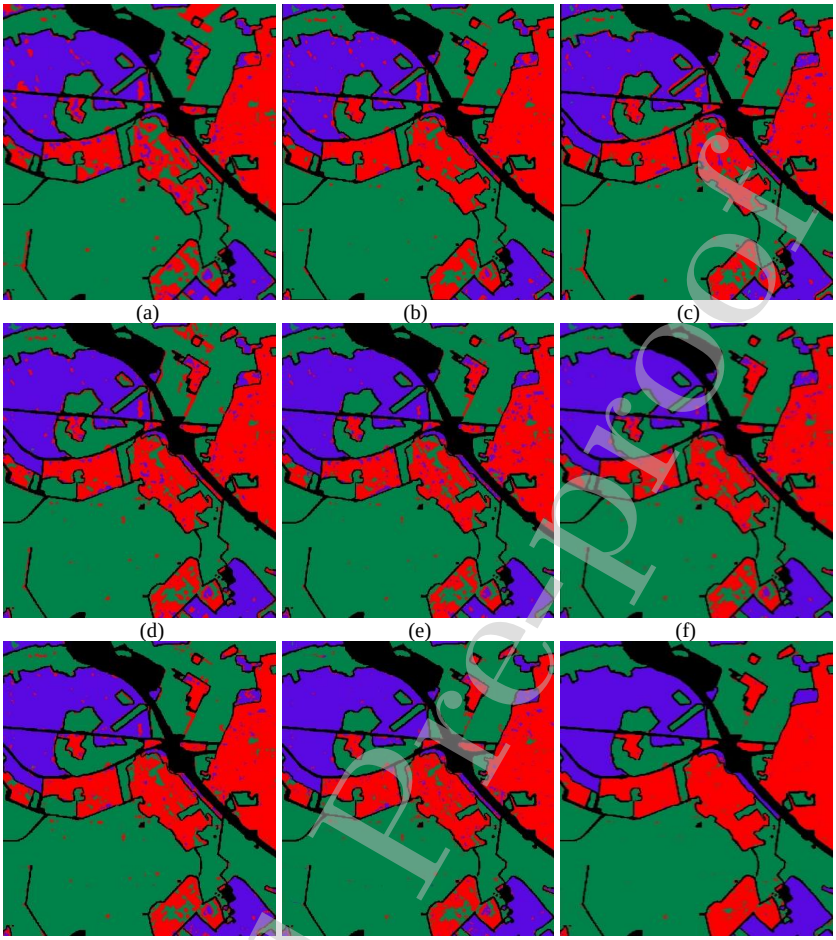


Fig. 10. Classification result maps with mask of Oberpfaffenhofenin data set (3 classes): (a) SVM (b) W-DBN (c) CV-CNN (d) Network A (e) Network B (f) Network C (g) Network D (h) DSNet (1% sampling rate) (i) DSNet (5% sampling rate)

Under 5% sampling rate, there are too many noise points in the classification maps of SVM (Fig. 10(a)) and W-DBN (Fig. 10(b)). While the result of DSNet (Fig. 10(i)) has less noise points and it is very smooth. It is clear that the result of DSNet is closer to the ground truth map.

D. The influence of ground truth maps

In TABLE IV, DSNet gets very high accuracy and its final result map is very close to the ground truth map. Unfortunately, manually labeled "ground truth" is not always believable, due to speckle noise and other factors [45]. The ground truth map and the results in Fig. 9. may seem to be unnaturally rough. To get a more accurate result, we use another ground truth map created by [44], shown in Fig. 11.

This ground truth map contains 5 different types of land, including woodland, farmland, suburban, road and other objects. This is finer-grained classification than the ground truth map used in Section C. The test results are listed in TABLE V, and the result maps are shown in Fig. 12.

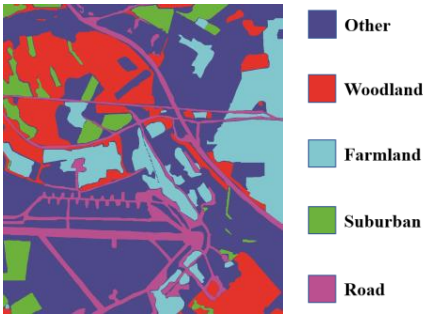


Fig. 11. Oberpfaffenhofenin data set (5 classes): (a) Ground truth map. (b) Legend of different classes.

In TABLE V, the OA of all algorithms decreases significantly, but DSNet still achieves the best result. Its AA, OA and Ka reach 81.75%, 85.19% and 78.37% respectively. The accuracy of road of DSNet (1% sampling rate) reaches 75.01%, which is obviously higher than other algorithms.

TABLE V
CLASSIFICATION RESULTS OF OBERPFAFFENHOFENIN DATA SET.

Class	SVM	W-DBN	CV-CNN	Network A	Network B	Network C	Network D	DSNet (1%)	DSNet (5%)
Other	0.9234	0.8944	0.9488	0.9229	0.9267	0.8889	0.9118	0.8955	0.9442
Woodland	0.7763	0.8545	0.8222	0.8290	0.8479	0.8401	0.8628	0.8709	0.9190
Farmland	0.6776	0.8224	0.8400	0.7843	0.8099	0.8024	0.7601	0.8062	0.9123
Suburban	0.1621	0.5879	0.6975	0.5706	0.5772	0.6919	0.7442	0.7649	0.8851
Road	0.1593	0.5959	0.4452	0.5381	0.5142	0.6747	0.6772	0.7501	0.8226
Sample rate	5%	5%	1%	1%	1%	1%	1%	1%	5%
AA	0.5397	0.7510	0.7501	0.7290	0.7352	0.7796	0.7912	**0.8175**	**0.8966**
OA	0.7220	0.8219	0.8352	0.8177	0.8245	0.8294	0.8414	**0.8519**	**0.9169**
Kappa	0.5619	0.7173	0.7265	0.7249	0.7348	0.7493	0.7657	**0.7837**	**0.8784**

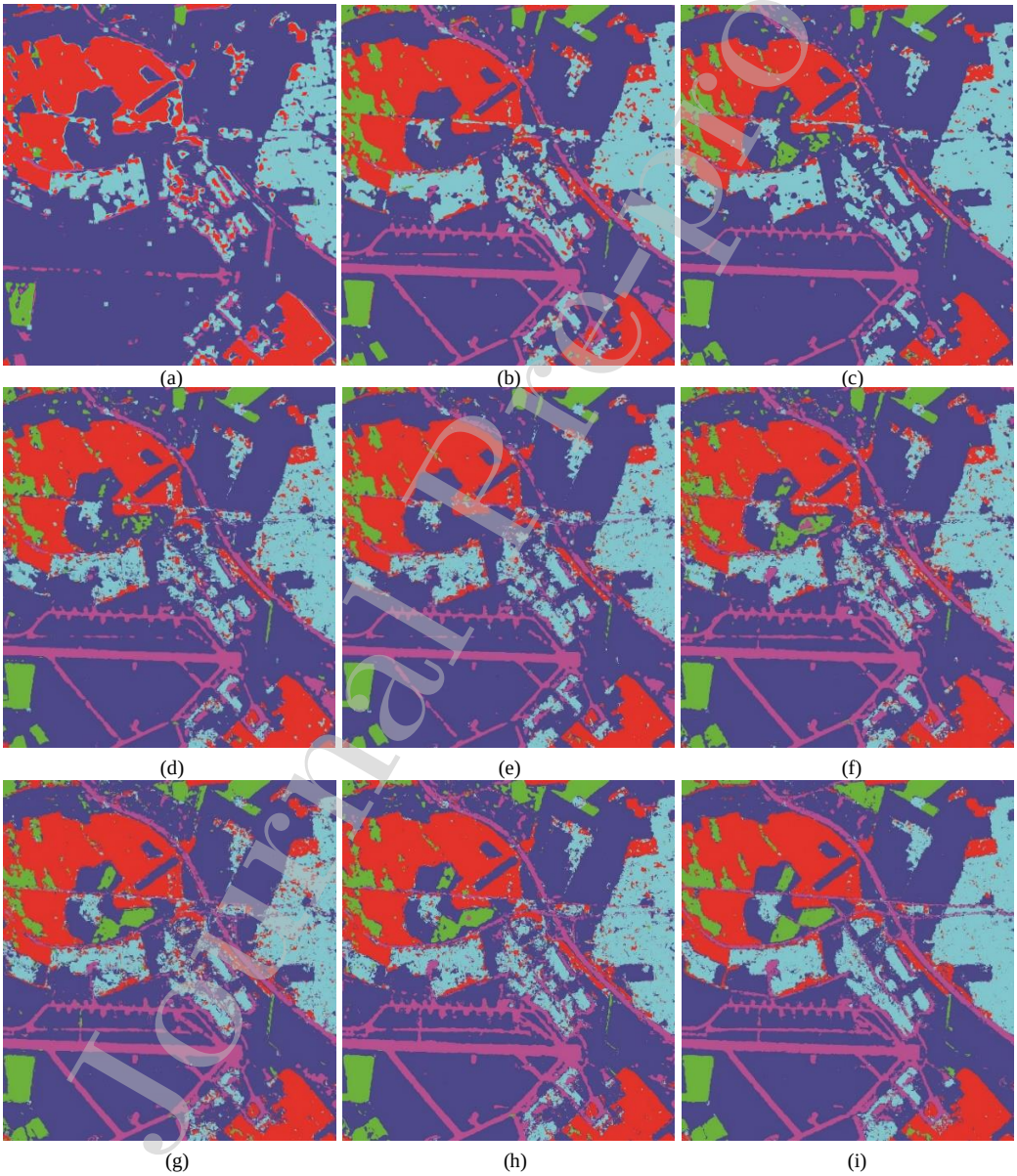


Fig. 12. Classification result maps of Oberpfaffenhofenin data set: (a) SVM (b) W-DBN (c) CV-CNN (d) Network A (e) Network B (f) Network C (g) Network D (h) DSNet (1% sampling rate) (i) DSNet (5% sampling rate)

In Fig. 12, a lot of pixels of road is misclassified in (a), (b), (c), (d) and (e). While (f), (g) and (h) have more accurate segmentation in road class. TABLE IV and TABLE V suggest that DSNet outperforms the other algorithms in different

ground truth maps. Clearly it is very important that ground truth maps are sufficiently reliable. Low-quality labeled data will have undesirable impact on the performance of classification algorithms. One solution is to select a smaller amount of more credible data to train the network. Therefore, the networks which require less training data will have great advantages.

E. Time Consumption Analysis

We have compared the computation time of four different networks, including network A, B, C mentioned above and DSNet. We have measured the forward propagation and back propagation time that each network required for each batch (1000 samples per batch). Our experiments were executed in Tensorflow 1.10.0 environment using Intel core i7 2.60 GHz CPU under single thread conditions. The overall computation time is listed in Table VI.

TABLE VI
TIME CONSUMPTION ANALYSIS

Network	Foreword calculation Time /s	Back propagation Time/s
A	0.3965	1.3235
B	0.3072	1.2528
C	0.2629	0.5850
DSNet	**0.1993**	**0.5035**

It can be seen that DSNet has significantly lower forward calculation time and back propagation time than the other three algorithms. The combination of dense connection and depthwise separable convolution can save a lot of computing resources and be run more efficiently compared with independent dense connection or depthwise separable convolution.

F. data augmentation and up-sampling

In this section, we use data augmentation and up-sampling to further improve the performance of DSNet. Data augmentation is a useful method that applies some reasonable transforms to the training data to generate additional training samples. It can enhance the performance of a network and alleviate the problem of overfitting. Our data augmentation rotates images by 90, 180 or 270 degrees randomly. To handle imbalanced classification problems, we use up-sampling (resample or copy instances from the minority class to match the number of samples of the majority class). In this way, each class number is roughly equal in every training epoch. The result is shown in Table VII.

Data1, Data2 and Data3 represent Flevoland data set 1, 2 and Oberpfaffenhofenin data set (3 classes) respectively. Data augmentation and up-sampling can slightly improve the performance of DSNet in AA, OA and Kappa. It is therefore recommended to use both of them.

TABLE VII
RESULTS UNDER DIFFERENT TRAINING CONDITION

Training condition		Raw	Data augmentation	Up-sampling	Data augmentation and Up-sampling
Data1	AA	0.9796	0.9732	0.9828	**0.9838**
	OA	0.9816	0.9823	0.9819	**0.9828**
	Kp	0.9799	0.9807	0.9802	**0.9812**
Data2	AA	0.9762	0.9759	0.9904	**0.9943**
	OA	0.9923	0.9926	0.9930	**0.9940**
	Kp	0.9910	0.9912	0.9917	**0.9929**
Data3	AA	0.9514	0.9614	0.9587	**0.9634**
	OA	0.9573	0.9675	0.9601	**0.9681**
	Kp	0.9272	0.9444	0.9321	**0.9455**

G. The Architecture of DSNet

The architecture of DSNet is based on the sub-network (from C3 layer to F6 layer) of LeNet [35], which includes a convolutional layer with 5 × 5 filters (C3), a max-pooling layer (stride is 2) with 2 × 2 filters (S4), another convolutional layer with 5 × 5 filters (C5) and a fully connected output layer with o' dimensions (F6), where o' is the class number.

TABLE VIII
THE ARCHITECTURES OF DIFFERENT OF NETWORKS.

Network Structure		
N1	N2	N3
DpCov(9, 27)@6*6+Max-pooling		
DpCov(72, 216)@5*5	DpCov(72, 144)@3*3	DpCov(72, 216)@3*3
	DpCov(216, 216)@3*3	Max-pooling
FC		

"DpCov(y,z)" represents the depthwise separable convolution with y depthwise convolutional filters and z pointwise convolutional filters.

The size of input feature maps of LeNet's C3 is 14 × 14, while in this paper we have used 15 × 15. We use 6 × 6 filters to replace the first 5 × 5 filters to adapt this difference. We have designed 3 different networks (N1, N2 and N3), and their architectures are listed in TABLE VIII.

N1 is similar to LeNet's sub-network. N2 adopts the structure of VggNet [26] to use two 3 × 3 convolutional filters to replace the second 5 × 5 filter, which can reduce some free parameters. And N3 uses a 3 × 3 convolutional filter and a max-pooling layer with 3 × 3 filters and 1 stride to replace the second 5 × 5 filter, which can reduce more parameters on the basis of N2. Final results of these networks are shown in TABLE IX.

TABLE IX
CLASSIFICATION RESULTS IN DIFFERENT DATA SETS.

network		N1	N2	N3
Data1	AA	0.9653	**0.9796**	0.9545
	OA	0.9773	**0.9816**	0.9739
	Ka	0.9752	**0.9799**	0.9715
Data2	AA	0.9325	**0.9762**	0.9440
	OA	0.9814	**0.9923**	0.9860
	Ka	0.9781	**0.9910**	0.9835
Data3	AA	0.9421	**0.9514**	0.9458
	OA	0.9525	**0.9573**	0.9544
	Ka	0.9185	**0.9272**	0.9219

The results show that N2 provides better performance than N1 and N3 in all three datasets.

IV. Conclusions

CNN have achieved great success in image classification problems. To overcome the limited training data available in PolSAR imaging problems, we designed a novel CNN, called DSNet. DSNet uses depthwise separable convolution to replace the standard convolution operation, which can extract the features of PolSAR images more efficiently and avoids obtaining redundant features. DSNet also introduces dense connections into networks to reuse feature maps and strengthen information transmission. Due to the improved structure, DSNet has fewer parameters than regular CNNs. DSNet was tested on three real PolSAR data sets and compared against several commonly used algorithms, such as SVM, W-DBN, CV-CNN. Experimental results show that DSNet achieves better results than conventional CNN and other algorithms. Its AA, OA and Kappa coefficients achieve the best scores for three different benchmark data sets. The structure of DSNet can also be conveniently applied to other neural networks, such as complex-valued neural networks, where it may achieve additional useful results. Finally, CNN is a conditional probability model. It therefore cannot handle those classification problems where the training dataset has distributions that are inconsistent with those of the test dataset. If training dataset and test dataset are not taken from one single PolSAR image (e.g. training dataset and test dataset may have different wavelength or look angle), then CNN methods are unlikely to perform well.

References

[1] P. Formont, F. Pascal, G. Vasile, J.-P. Ovarlez, and L. Ferro-Famil, "Statistical classification for heterogeneous polarimetric SAR images," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 398-407, Jun. 2011.

[2] J. S. Lee, M. R. Grunes, E. Pottier, and L. F. Famil, "Unsupervised terrain classification preserving polarimetric scattering characteristics," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 4, pp. 722-731, Apr. 2004.

[3] H. Liu, Z. Wang, F. Shang, S. Yang, S. Gou, and L. Jiao, "Semi-supervised tensorial locally linear embedding for feature extraction using PolSAR data," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1476–1490, Dec. 2018.

[4] W. L. Cameron and L. K. Leung, "Feature motivated polarization scattering matrix decomposition," in *Proc. IEEE Int. Radar Conf.*, May 1990, pp. 549-557.

[5] A. Freeman and S. L. Durden, "A three-component scattering model for polarimetric SAR data," *IEEE Trans. Geosci. Remote Sens.*, vol.36, no.3, pp. 963-973, May 1998.

[6] S. R. Cloude and E. Pottier, "An entropy based classification scheme for land applications of polarimetric SAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 1, pp. 68-78, Jan. 1997.

[7] 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 *Proc. SPIE Radar Polarimetry*, 1993, pp. 72-85.

[8] G. P. S. Junior, A. C Frery, S. Sandri, H. Bustince, et al. "Optical images-based edge detection in Synthetic Aperture Radar images," *Knowledge-Based Systems.*, 87, pp. 38-46, 2015.

[9] J. S. Lee, K. W. Hoppel, S. A. Mango, and A. R. Miller, "Intensity and phase statistics of multilook polarimetric and interferometric SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 5, pp. 1017-1028, Sep. 1994.

[10] S. H. Yueh, J. A. Kong, J. K. Jao, R. T. Shin, and L. M. Novak, "K-distribution and polarimetric terrain radar clutter," *J. Electromagn. Waves Appl.*, vol. 3, no. 8, pp. 747-768, 1989.

[11] C. C. Freitas, A. C. Frery, and A. H. Correia. "The polarimetric G distribution for SAR data analysis," *Environmetrics*, vol. 16, no. 1, pp. 13-31, Feb. 2005.

[12] A. C. Frery, H.-J. Müller, C. C. F. Yanasse, S. J. S. Sant'Anna, "A model for extremely heterogeneous clutter," *IEEE Trans. Geosci. Remote Sensing*, vol. 35, pp. 648-659, May 1997.

[13] R. Díaz-Morales, and Á. Navia-Vázquez. "LIBIRWLS: A parallel IRWLS library for full and budgeted SVMs," *Knowledge-Based Systems*, 136, pp. 183-186, 2017.

[14] H. Wang, J. Gu, and S. Wang. "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Systems*, 136, pp. 130-139, 2017.

[15] X. She, J. Yang, and W. Zhang, "The boosting algorithm with application to polarimetric SAR image classification," in *Proc. 1st APSAR Conf.*, 2007, pp. 779-783.

[16] F. D. N. Neto, C. de Souza Baptista, and C. E. Campelo. "Combining Markov model and prediction by partial matching compression technique for route and destination prediction," *Knowledge-Based Systems*, 154, pp. 81-92, 2018.

[17] A. P. Doulgeris, "An automatic U-distribution and Markov random field segmentation algorithm for PolSAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1819-1827, Apr. 2015.

[18] B. Hou, H. Kou and L. Jiao, "Classification of polarimetric SAR images using multilayer autoencoders and superpixels," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 3072-3081, Jul. 2016.

[19] F. Liu, L. Jiao, B. Hou and S. Yang, "POL-SAR image classification based on Wishart DBN and local spatial information," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3292-3308, Jun. 2016.

[20] L. Jiao and F. Liu, "Wishart deep stacking network for fast POLSAR image classification," *IEEE Trans. Image Process*, vol. 25, no. 7, pp. 3273-3286, Jul. 2016.

[21] Z. Zhang, H. Wang, F. Xu and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177-7188, Dec. 2017.

[22] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin, "Polarimetric SAR image classification using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1935-1939, Dec. 2016.

[23] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[24] H. Liu, B Lang, M Liu, H Yan. "CNN and RNN based payload classification methods for attack detection," *Knowledge-Based Systems*, 163, pp. 332-341, 2019.

[25] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. 25*, Lake Tahoe, Nevada, USA, 2012, pp. 1106-1114.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations*, 2015.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*., Boston, MA, 2015, pp. 1-9.

[28] Peng Gao, Qiquan Zhang, Fei Wang, Liyi Xiao, Hamido Fujita, Yan Zhang, "Learning reinforced attentional representation for end-to-end visual tracking" Information Sciences, in-press https://doi.org/10.1016/j.ins.2019.12.084.

[29] Peng Gao, Ruyue Yuan, Fei Wang, Liyi Xiao, Hamido Fujita, Yan Zhang, "Siamese attentional keypoint network for high performance visual tracking" Knowledge-Based Systems, in-press https://doi.org/10.1016/j.knosys.2019.105448.

[30] Yu, Zeng, Tianrui Li, Guang chun Luo, Hamido Fujita, NingYu,, Yi Pan "Convolutional networks with cross-layer neurons for image recognition." Information Sciences 433 (2018): 241-254.

[31] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*., Honolulu, HI, 2017, pp. 1800-1807.

[32] A. G. Howard *et al*. (2017). "MobileNets: Efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.org/abs/1704.04861

[33] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, 2016, pp. 770-778.

[34] G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Honolulu, HI, 2017, pp. 2261-2269.

[35] V. Nair, G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proc. Int'l Conf. Machine Learning*, 2010.

[36] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition", *Proc. IEEE Int. Conf. Signal Image Process. Appl.*, pp. 342-347, 2011.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I Sutskever and R Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929-1958, 2014.

[38] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[39] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, 2015.

[40] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist*., 2011, pp. 315-323.

[41] P. Yu, A. Qin, and D. A. Clausi, "Unsupervised polarimetric sar image segmentation and classification using region growing with edge penalty," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1302-1317, 2012.

[42] D. H. Hoekman and M. A. M. Vissers, "A new polarimetric classification approach evaluated for agricultural crops," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 12, pp. 2881-2889, Dec. 2003.

[43] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 907-924, Feb. 2013.

[44] H. Liu, Y. Wang, S. Yang, S. Wang, J. Feng, and L. Jiao, "Large polarimetric SAR data semi-supervised classification with spatial-anchor graph," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 4, pp. 1439-1458, Apr. 2016.

[45] B. Hou, Q. Wu, Z. Wen and L. Jiao, "Robust Semisupervised Classification for PolSAR Image With Noisy Labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6440-6455, Nov. 2017.

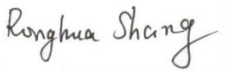
AUTHOR DECLARATION TEMPLATE

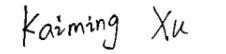
We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work. [OR] We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from (rhshang@mail.xidian.edu.cn) Signed by all authors as follows:





CRediT Author Statement

Ronghua Shang:Conceptualization, Software, Resources, Writing - Review & Editing, Project administration. **Jianghai He**: Methodology, Formal analysis, Investigation, Writing - Review & Editing, Visualization. **Jiaming Wang**: Methodology, Validation, Formal analysis, Writing - Original Draft. **Kaiming Xu**: Validation, Investigation, Data Curation. **Licheng Jiao**: Resources, Supervision, Project administration, Funding acquisition. **Rustam Stolkin**: Writing - Review & Editing