Complex-Valued Convolutional Neural Network and Its Application in Polarimetric SAR Image Classification

Zhimian Zhang, Haipeng Wang, *Senior Member, IEEE*, Feng Xu, *Senior Member, IEEE*, and Ya-Qiu Jin, *Fellow, IEEE*

***Abstract*— Following the great success of deep convolutional neural networks (CNNs) in computer vision, this paper proposes a complex-valued CNN (CV-CNN) specifically for synthetic aperture radar (SAR) image interpretation. It utilizes both amplitude and phase information of complex SAR imagery. All elements of CNN including input–output layer, convolution layer, activation function, and pooling layer are extended to the complex domain. Moreover, a complex backpropagation algorithm based on stochastic gradient descent is derived for CV-CNN training. The proposed CV-CNN is then tested on the typical polarimetric SAR image classification task which classifies each pixel into known terrain types via supervised training. Experiments with the benchmark data sets of Flevoland and Oberpfaffenhofen show that the classification error can be further reduced if employing CV-CNN instead of conventional real-valued CNN with the same degrees of freedom. The performance of CV-CNN is comparable to that of existing state-of-the-art methods in terms of overall classification accuracy.**

***Index Terms*— Complex-valued convolutional neural network (CV-CNN), deep learning, synthetic aperture radar (SAR), terrain classification.**

I. Introduction

DEEP learning has demonstrated superior performance in the pattern recognition field, e.g., image classification [1], object detection [2], and natural language processing [3]. Commonly used deep learning algorithms include deep belief network (DBN), convolutional neural network (CNN), and recurrent neural network, where CNN is primarily applied in computer vision. In the computer vision field, conventional image classification algorithms consist of two stages, i.e., feature extraction and feature classification. The feature extractor is often handcrafted with statistical or physical characteristics to make a good representation of different types of targets. However, such paradigm has been revolutionized by CNN which automatically learns hierarchical representations from the data, and it demonstrated superior performances over conventional approaches by a significant margin [1], [4], [5]. It has since achieved a series of breakthroughs in tasks such as image classification. On the 2012 ImageNet Large-Scale Visual Recognition Challenge, Krizhevsky *et al.* [1] employed a deep CNN to achieve the test error rate of 15.3%, which was considerably better than that of the previous state of the art. In 2014, Szegedy *et al.* [6] proposed the GoogLeNet, a deep network with 22 layers, and reduced the top-5 error to 6.67%. In 2015, He *et al.* [4] proposed the residual networks with 152 layers and achieved a 3.57% ensemble error.

In parallel to the huge success of CNN in computer vision area, one might wonder whether such powerful technology could be adapted to synthetic aperture radar (SAR) imagery interpretation and potentially bring huge advancements in applications such as terrain surface classification, parameter inversion, and target recognition. However, only a handful of studies has recently emerged in the field of deep learning applications in SAR imagery interpretation. For example, Chen *et al.* [7] adopted a deep convolutional network for SAR target identification and achieved a record-breaking accuracy of over 99% for MSTAR 10-class data sets. Hou *et al.* [8] proposed a new polarimetric SAR (POLSAR) image classification method based on multilayer autoencoders and superpixels. Zhou *et al.* [9] applied convolutional network to POLSAR image classification and achieved a 92% accuracy on the Flevoland benchmark data. Nevertheless, these studies directly applied real-valued neural networks to the amplitude of SAR image (usually converted to dB scale), while neglecting phase information.

Phase information is unique to SAR image, and it is a crucial component in many SAR applications, with the most prominent one being SAR interferometry. It is in the authors' opinion that phase information has to be taken into account for many other SAR information retrieval applications such as target classification and recognition. In particular, for POLSAR data, the phase of off-diagonal elements of the covariance/coherency matrix is useful in identifying different types of scatterers. In order to fully explore the rich information embedded in complex-valued (CV) SAR images, complex version of deep neural networks has to be developed. Hirose *et al.* [10], [11] have pioneered in the application of CV-CNNs to various areas, including land-surface classification and motion control in combination with reinforcement learning. Recently, it has further extended to quaternion neural networks [12] for phase-sensitive POLSAR classification.

Manuscript received July 12, 2016; revised December 10, 2016, April 12, 2017, and July 30, 2017; accepted August 15, 2017. Date of publication September 18, 2017; date of current version November 22, 2017. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0502700 and in part by the National Natural Science Foundation of China under Grant 61571132, Grant 61571134, and Grant 61471127. *(Corresponding authors: Haipeng Wang; Feng Xu.)*

The authors are with the Key Laboratory for Information Science of Electromagnetic Waves (MoE), Fudan University, Shanghai 200433, China (e-mail: hpwang@fudan.edu.cn; fengxu@fudan.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2743222

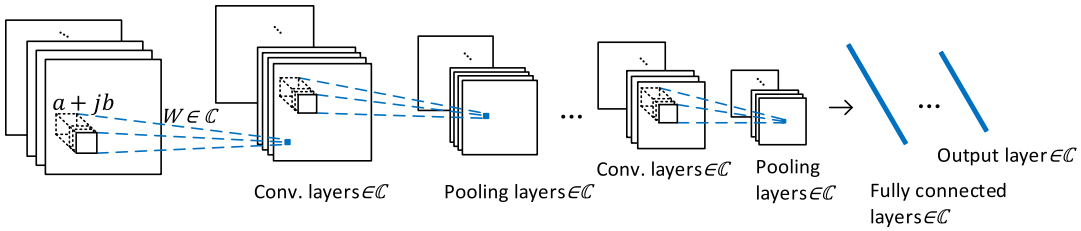


Fig. 1. Overview of the CV-CNN architecture.

The primary contribution of this paper is a deep CV-CNN. Both the feedforward neutral network model with CV data stream and the corresponding complex backpropagation training algorithm are developed. It provides the theoretical basis for further developments of SAR-oriented deep learning algorithms.

The second contribution of this paper is an experimental study of the newly developed CV-CNN in a typical application of POLSAR image classification. It aims to classify image pixels into different classes, representing known terrain types, such as forest, grass, water, urban area, and sand. It falls perfectly within conventional artificial intelligent tasks. With the rapid development of both spaceborne and airborne POLSAR systems, processing tasks of the daily retrieved massive POLSAR images certainly demands an automatic or semiautomatic intelligent algorithm. Hence, the development of such CV-CNN would partly address some of these tasks.

Like what has occurred to computer vision tasks, this paper demonstrates that significant improvement can be achieved in the POLSAR image classification by using the proposed SAR-oriented CV-CNN. On the contrary, such great improvement would not have been possible without the ad.

A brief review is given here of several conventional methods for the POLSAR image classification. They can be generally categorized into three major types. One is based on statistical characteristics of the scattering coherent matrix. For single-look complex (SLC) POLSAR data, Kong *et al.* [13] derived a maximum likelihood (ML) classifier based on the complex Gaussian distribution. For multilook data represented in covariance or coherency matrices, Lee *et al.* [14] derived a distance measure based on the complex Wishart distribution assuming that each frequency is statistically independent. Another alternative approach is based on the inherent characteristics of polarimetric scattering mechanisms, which has the additional virtue by providing some information of class-type identification. A hierarchical classification algorithm was proposed by Freeman *et al.* [15] based on scattering characteristics of different farmlands in different frequencies. Kouskoulas *et al.* [16] estimated the probability density function of principal scattering parameters of different terrains in multifrequencies with the maximum entropy method, and then the Bayesian hierarchical classifier is used to classify short vegetation. The third category combines polarimetric scattering characteristics and statistical properties. Lee *et al.* [17] proposed an approach used the H/α target decomposition [18] and the ML classifier based on the complex Wishart distribution.

This paper first presents a CV-CNN, which not only takes complex data as input, but also propagates the phase information through all layers. It is necessary because the entire hierarchic representation of SAR image has to be constructed using CV data. The proposed CV-CNN extends the entire network including both data and parameters into the complex domain, which means that all mathematical operations are extended under complex analysis theory. It also includes a complex backpropagation algorithm for network training. The performance of CV-CNN when applied to POLSAR classification is evaluated with several benchmark data sets and compared with conventional real-valued counterparts with the same configurations. Significant improvements in terms of classification accuracy are achieved.

The remainder of this paper is organized as follows. Section II presents a detailed theory for CV-CNN. Section III introduces the application of the proposed algorithm for SAR classification including the POLSAR data adaption and the network configuration. Experiments and discussions on different data sets are described in Section IV. Section V concludes this paper.

II. Theory for Complex-Valued Convolutional Neural Network

This section introduces the general CV-CNN framework which is extended from the conventional real-valued CNN (RV-CNN), along with the derivation of a complex backpropagation algorithm based on stochastic gradient descent (SGD).

A. Architecture of CV-CNN

As shown in Fig. 1, the architecture of a CV-CNN can be regarded as a variant of the deep neural networks. It is designed to take advantage of the 2-D structure of the input image. In image classification, one or more 2-D matrices or channels are treated as the input. The hidden units of convolutional and pooling layers are also organized as a set of 2-D matrices which are called feature maps. Just like RV-CNN, the CV-CNN also include an input layer, several alternations of convolutional and pooling layers, fully connected layers, and a final classifier layer. Among the multiple layers, the lower and higher layers are used to learn the low-level and high-level feature representations, respectively [19]. The input layer usually has a size of width by height by depth, where depth indicates the number of channels of the input image. In the case of SAR image classification, the complex multichannel SAR image could be directly fed into the CV-CNN.

A typical process of feature extraction in CNN is a convolution layer with nonlinear activation followed by a pooling layer. Convolution layers perform convolution with multiple learnable filters in parallel. Both the input and output could be multichannel 2-D matrices. The convolutional results are

fed into nonlinear activation functions, such as sigmoid or the recently introduced rectified linear unit [20] function, to generate feature maps. Then, a pooling function is used to downsample/locally accumulate the feature maps in order to reduce spatial dimension. These represent the typical characteristics of CNN, i.e., local connectivity, weight sharing, pooling, and cascaded layers [21]. For CV-CNN, all the elements of the networks, including filters, activation functions, and pooling, should be fully CV.

The details of each element in its CV version are presented in the following.

1) Convolution: Each hidden unit, which is CV, in the convolutional layers is connected to local patches in the feature maps, also CV, of the previous layer through a set of complex weight matrices known as kernels or filter banks. The units in a local patch are convolved by the weight matrix, and then passed through the nonlinearity activation function. By convolution, it means that the units in a feature map are forced to share the same filer bank. Different feature maps in a layer use different filter banks, which corresponds to different channels. Each filter bank detects specific regional features from the input matrices, so each feature map represents a unique feature at different positions of the previous layer. One of the advantages by using local connectivity and weight sharing is that the number of free parameters to be learned is significantly reduced, which helps to avoid the over-fitting problem and also reduces the memory requirements. The usage of this convolutional architecture is supported by the fact that image data are often spatially correlated and the interesting features should be translational invariant [21]. In other words, if a pattern appears in one part of the image, it could appear anywhere. This is the reason why the units at different locations sharing the same weights and detecting the same pattern in different parts of the image.

In the convolutional layer, the complex output feature maps $O_i^{(l+1)} \in \mathbb{C}^{W_2 \times H_2 \times I}$ are computed by the convolution between all the previous layer's input feature maps $O_k^{(l)} \in \mathbb{C}^{W_1 \times H_1 \times K}$ and a bank of filters $w_{ik}^{(l+1)} \in \mathbb{C}^{F \times F \times K \times I}$, and then add a bias $b_i^{(l+1)} \in \mathbb{C}^I$, where C denotes the complex domain and the superscript is its dimension. This convolution is calculated by

$$\begin{aligned} O_i^{(l+1)} &= f\left(\Re\left(V_i^{(l+1)}\right)\right) + jf\left(\Im\left(V_i^{(l+1)}\right)\right) \\ &= \frac{1}{1+e^{-\Re\left(V_i^{(l+1)}\right)}} + j\frac{1}{1+e^{-\Im\left(V_i^{(l+1)}\right)}} \end{aligned} \tag{1}$$

$$\begin{aligned} V_i^{(l+1)} &= \sum_{k=1}^{K} w_{ik}^{(l+1)} * O_k^{(l)} + b_i^{(l+1)} \\ &= \sum_{k=1}^{K} \left(\Re\left(w_{ik}^{(l+1)}\right) \cdot \Re\left(O_k^{(l)}\right) - \Im\left(w_{ik}^{(l+1)}\right) \cdot \Im\left(O_k^{(l)}\right)\right) \\ &\quad + j\sum_{k=1}^{K} \left(\Re\left(w_{ik}^{(l+1)}\right) \cdot \Im\left(O_k^{(l)}\right)\right. \\ &\quad \left. + \Im\left(w_{ik}^{(l+1)}\right) \cdot \Re\left(O_k^{(l)}\right)\right) + b_i^{(l+1)} \end{aligned} \tag{2}$$

where $j = \sqrt{-1}$ is the imaginary unit. Character $*$ is the convolution operation. $\Re$ and $\Im$ denote the real and imaginary parts of a complex number. $O_k^{(l)}$ is the unit of the kth input feature map in layer l, and $V_i^{(l+1)}$ denotes the weighted sum of inputs to the ith output feature map in layer $l+1$. $f(\cdot)$ denotes a nonlinear function. Note that the sigmoid function is used in this paper.

The hyperparameters of a convolutional layer include number of feature maps I, filter size $F \times F \times K$, stride S, and zero-padding size P. The stride means the intervals of each moving when the filters contact to the input feature maps. Due to the convolution operation, the valid output feature map is smaller than that of the input. By the zero-padding process, the size shrinking with depth can be well compensated with proper P, which helps to make an arbitrary deep convolutional network. If the input is composed of K feature maps with size $W_1 \times H_1$, and the output will be with I feature maps of size $W_2 \times H_2$, where $W_2 = (W_1 - F + 2P)/S + 1$ and $H_2 = (H_1 - F + 2P)/S + 1$. Recent research indicates that using small filter sizes (e.g., 3×3 or 5×5) with a stride of 1 usually produces a better performance [22]; therefore, the filter size and stride of this paper are 3×3 and 1, respectively. The parameters to be learned in the convolutional layer include $F \times F \times K \times I$ weights and I biases. The aim of the learning procedure is to find sets of filters w that can extract discriminative features for image classification [23].

2) Pooling: Pooling layer is to merge semantically similar features that are detected by the convolutional layer. The pooling function calculates a summary statistic over a local patch independently for each feature map in the convolutional layer. Therefore, the number of feature map in pooling and convolutional layers is equal. Maximum and average of a rectangular neighborhood are the two mostly used pooling functions. In other words, pooling layers can be regarded as subsampling layers [24]. Apart from reducing the dimension of feature, pooling also helps to make the representation invariant to small shifts and distortions of the input. Invariance to local translation can be a very useful property, when the feature itself is considered rather than its location.

A straightforward extension of average pooling from real to complex can be defined as

$$O_i^{(l+1)}(x, y) = \underset{u,v=0,\ldots,g-1}{\text{ave}} O_i^{(l)}(x \cdot s + u, y \cdot s + v) \tag{3}$$

where g denotes pooling size and s is the stride. $O_i^{(l+1)}(x, y)$ is the unit of the ith input feature map at position (x, y). However, for max pooling, extension from real to complex is not readily available. A natural way is to simply take the amplitude maximum. In this paper, we use the CV average pooling as defined in (3).

Pooling size of 2×2 or 3×3 with a stride of 2 is most frequently used. It is known that with increasing pooling size or stride, more information is abandoned and leading to a poorer performance.

3) Fully Connected Layer: In the top layers, usually one or more fully connected layers are used in our CV-CNN. That is, each neuron in the fully connected layers is connected to all neurons in the preceding layer, which can be seen

as a special case of convolution layer. The number of fully connected layers and each layer's neurons is not fixed. Usually, lower layers tend to have fewer neurons, while higher layers have more neurons. The output can be written as

$$O_i^{(l+1)} = f\left(\Re\left(V_i^{(l+1)}\right)\right) + jf\left(\Im\left(V_i^{(l+1)}\right)\right) \tag{4}$$

$$V_i^{(l+1)} = \sum_{k=1}^{K} w_{ik}^{(l+1)} \cdot O_k^{(l)} + b_i^{(l+1)} \tag{5}$$

where K denotes the number of neurons in lth fully connected layers.

4) Output Layer: After multihierarchical feature extraction stages, the final output layer which is a $1 * c$ vector consisting of complex value acts as the classifier to predict the classification of the input sample, while c is the number of classes. In CV-CNN, the label is organized as one-hot encoding vector, where the ON value is instead by $(1 + 1 * j)$. The length of the vector is the number of classes, and the location of $(1 + 1 * j)$ in the vector is the actual class number of the input and the other location is set to zero. After calculating the distance between each element in the output vector and $(1 + 1 * j)$, the location number of the element with the smallest distance is the required category. Given input samples and their labels, the parameters of CV-CNN are learned in an end-to-end supervised way by minimizing a loss function on training data.

In RV-CNN, the output layer is usually a softmax classifier predicting the probability distribution over different classes. Then, the entire network is learned by minimizing the log-loss function [7]. If the softmax is applied to CV-CNN, the result is not a probability due to its CV input. Therefore, the final output is the classifier, and the least-squares loss function is adopted in CV-CNN.

B. Complex Backpropagation for CV-CNN

Supervised training of a CNN is to optimize the weights/biases so that the network output matches with the desired output or the given labels of the training data. After the multiple feature extraction stages, there still exists error between the output and the target. The error can be described as a loss function E, for exampling using the classic least-squared error in CV-CNN. Therefore, the network parameters are trained by SGD and minimizing the loss function in backpropagation [25]. SGD refers to estimating the gradient using only a subset of training samples, i.e., 100-sample minibatch [26]. Therefore, the estimated gradient is actually sampled from a small number of cases, which is in fact a stochastic sampling process. By computing the error gradient with respect to parameters $(\partial E/\partial w)$, the updating rule is $w \leftarrow w - \eta\ (\partial E/\partial w)$, where η is the learning rate.

The training sample can be expressed as $\{X[n], T[n]\}_{n=1}^{N}$ where $X[n]$ and $T[n]$ denote the nth input data and the label, respectively. Both the input and label are complex value. Hence, the total classification error is given by

$$E = \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}[(\Re(T_k[n]) - \Re(O_k[n]))^2 + (\Im(T_k[n]) - \Im(O_k[n]))^2]. \tag{6}$$

The minimum of above loss function is searched by iteratively adjusting the weights according to

$$\begin{aligned} w_{ik}^{(l+1)}[t+1] &= w_{ik}^{(l+1)}[t] + \Delta w_{ik}^{(l+1)}[t] \\ &= w_{ik}^{(l+1)}[t] - \eta\frac{\partial E[t]}{\partial w_{ik}^{(l+1)}[t]} \end{aligned} \tag{7}$$

$$\begin{aligned} b_i^{(l+1)}[t+1] &= b_i^{(l+1)}[t] + \Delta b_i^{(l+1)}[t] \\ &= b_i^{(l+1)}[t] - \eta\frac{\partial E[t]}{\partial b_i^{(l+1)}[t]}. \end{aligned} \tag{8}$$

The derivatives of complex functions are obtained according to the complex chain rule [27]. The key point is computing the error gradient of weights

$$\begin{aligned} &\frac{\partial E}{\partial w_{ik}^{(l+1)}} \\ &= \frac{\partial E}{\partial \Re\left(w_{ik}^{(l+1)}\right)} + \frac{\partial E}{\partial \Im\left(w_{ik}^{(l+1)}\right)} \\ &= \left(\frac{\partial E}{\partial \Re\left(V_i^{(l+1)}\right)}\frac{\partial \Re\left(V_i^{(l+1)}\right)}{\partial \Re\left(w_{ik}^{(l+1)}\right)} + \frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)}\frac{\partial \Im(V_i^{(l+1)})}{\partial \Re(w_{ik}^{(l+1)})}\right) \\ &\quad + j\left(\frac{\partial E}{\partial \Re\left(V_i^{(l+1)}\right)}\frac{\partial \Re\left(V_i^{(l+1)}\right)}{\partial \Im\left(w_{ik}^{(l+1)}\right)} + \frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)}\frac{\partial \Im\left(V_i^{(l+1)}\right)}{\partial \Im\left(w_{ik}^{(l+1)}\right)}\right). \end{aligned} \tag{9}$$

In order to simplify the expression, an intermediate quantity called "error term" is defined

$$\delta_i^{(l+1)} = -\frac{\partial E}{\partial \Re\left(V_i^{(l+1)}\right)} - j\frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)}, \quad \delta_i^{(l+1)} \in \mathbb{C}. \tag{10}$$

With (1), (2), (6), and (10), (9) can be simplified as

$$\frac{\partial E}{\partial w_{ik}^{(l+1)}} = -\delta_i^{(l+1)}\overline{O_i^{(l)}} \tag{11}$$

where $\bar{(\cdot)}$ denotes taking the complex conjugate.

By iteratively reducing the above error, the parameters are updated until the error reaches a minimum. To show the detail of error term in each layer, the derivation is given in the following sections.

1) Error Term of Fully Connected Layers: For the error term of fully connected layers, assume that the $l + 1$ layer is output layer; then the previous layer l is called hidden layer, and error $\delta_i^{(l+1)}$ can be expanded as

$$\begin{aligned} &\delta_i^{(l+1)} \\ &= -\frac{\partial E}{\partial \Re\left(V_i^{(l+1)}\right)} - j\frac{\partial E}{\partial \Im\left(V_i^{(l+1)}\right)} \\ &= -\left(\frac{\partial E}{\partial \Re\left(O_i^{(l+1)}\right)}\frac{\partial \Re\left(O_i^{(l+1)}\right)}{\partial \Re\left(V_i^{(l+1)}\right)} + \frac{\partial E}{\partial \Im\left(O_i^{(l+1)}\right)}\frac{\partial \Im\left(O_i^{(l+1)}\right)}{\partial \Re\left(V_i^{(l+1)}\right)}\right) \\ &\quad - j\left(\frac{\partial E}{\partial \Re\left(O_i^{(l+1)}\right)}\frac{\partial \Re\left(O_i^{(l+1)}\right)}{\partial \Im\left(V_i^{(l+1)}\right)} + \frac{\partial E}{\partial \Im\left(O_i^{(l+1)}\right)}\frac{\partial \Im\left(O_i^{(l+1)}\right)}{\partial \Im\left(V_i^{(l+1)}\right)}\right). \end{aligned} \tag{12}$$

According to (4) and (5), the second and third terms in (12) are zero. Taking (6) into account, the other two terms can be

calculated out

$$\delta_i^{(l+1)} = \Re\big(T_i^{(l+1)} - O_i^{(l+1)}\big)\Re\big(O_i^{(l+1)}\big)\big(1-\Re\big(O_i^{(l+1)}\big)\big) + j\big(\Im\big(T_i^{(l+1)} - O_i^{(l+1)}\big)\Im\big(O_i^{(l+1)}\big)\big(1-\Im\big(O_i^{(l+1)}\big)\big)\big). \tag{13}$$

For the hidden layer's error term $\delta_k^{(l)}$, it can be unfolded similarly. Fully connection means that one unit in the hidden layer $O_k^{(l)}$ connects with all the units in the output layer $O_i^{(l+1)}$; thus, $O_k^{(l)}$ is affected by the error through all the units of output layer. Based on chain rulers, the real and imaginary parts of the error are deduced as

$$\frac{\partial E}{\partial \Re\big(O_k^{(l)}\big)} = -\sum_{i=1}^{I}\big[\Re\big(\delta_i^{(l+1)}\big)\Re\big(w_{ik}^{(l+1)}\big) + \Im\big(\delta_i^{(l+1)}\big)\Im\big(w_{ik}^{(l+1)}\big)\big] \tag{14}$$

$$\frac{\partial E}{\partial \Im\big(O_k^{(l)}\big)} = -\sum_{i=1}^{I}\big[\Im\big(\delta_i^{(l+1)}\big)\Re\big(w_{ik}^{(l+1)}\big) - \Re\big(\delta_i^{(l+1)}\big)\Im\big(w_{ik}^{(l+1)}\big)\big]. \tag{15}$$

Considering (12), $\delta_k^{(l)}$ will be

$$\begin{aligned}\delta_k^{(l)} &= \big(1-\Re\big(O_k^{(l)}\big)\big)\Re\big(O_k^{(l)}\big)\\ &\quad\times\sum_{i=1}^{I}\big[\Re\big(\delta_i^{(l+1)}\big)\Re\big(w_{ik}^{(l+1)}\big)+\Im\big(\delta_i^{(l+1)}\big)\Im\big(w_{ik}^{(l+1)}\big)\big]\\ &\quad+j\big(1-\Im\big(O_k^{(l)}\big)\big)\Im\big(O_k^{(l)}\big)\\ &\quad\times\sum_{i=1}^{I}\big[\Im\big(\delta_i^{(l+1)}\big)\Re\big(w_{ik}^{(l+1)}\big)-\Re\big(\delta_i^{(l+1)}\big)\Im\big(w_{ik}^{(l+1)}\big)\big].\end{aligned} \tag{16}$$

The error term of hidden layers is in line with that of output layers. Furthermore, it can be summarized that the lower layers' error term is the product of the higher layers' error, weights, and the derivative of nonlinear function to lower layer's input [28]. For real-valued network, it can be simply formulated as $\delta_k^{(l)} = (w^{(l+1)})^T\delta_i^{(l+1)} f'(V^{(l)})$.

2) Error Term of Convolutional Layers: If lth layer is the convolutional layer, its error term $\delta_k^{(l)}$ is related to the error term of the pooling layer $\delta_i^{(l+1)}$ and the pooling factor β. The weights at a pooling layer map are all set equally to β, which is a constant value computed by $\beta = (1/g\cdot g)$, which is inversely proportional to the square of pooling size g. Characters k and i denote the feature map in $l, l+1$ layers, respectively. Because of the subsampling, the size of $l+1$ layer is smaller than that of l layer. To keep the same size, it needs to up-sample the $\delta_i^{(l+1)}$ by copying each pixel g times in two directions, denoted as $up(\delta_i^{(l+1)})$ in the following. Similar as the derivation (16) of the hidden layer in fully connected layers, the error term becomes

$$\begin{aligned}\Re\big(\delta_k^{(l)}\big) &= \beta_i^{l+1}\cdot\big[\Re\big(up\big(\delta_i^{(l+1)}\big)\big)+\Im\big(up\big(\delta_i^{(l+1)}\big)\big)\big]\\ &\quad\times\big(\Re\big(O_k^{(l)}\big)\big(1-\Re\big(O_k^{(l)}\big)\big)\big)\end{aligned} \tag{17}$$

$$\begin{aligned}\Im\big(\delta_k^{(l)}\big) &= \beta_i^{l+1}\cdot\big[\Im\big(up\big(\delta_i^{(l+1)}\big)\big)-\Re\big(up\big(\delta_i^{(l+1)}\big)\big)\big]\\ &\quad\times\big(\Im\big(O_k^{(l)}\big)\big(1-\Im\big(O_k^{(l)}\big)\big)\big)\end{aligned} \tag{18}$$

$$\delta_k^{(l)} = \Re\big(\delta_k^{(l)}\big) + j\Im\big(\delta_k^{(l)}\big). \tag{19}$$

3) Error Term of Pooling Layers: There are no parameters to be learned in the pooling layer. However, it still needs to compute its error term in order to obtain the lower layer's error term during backpropagation. The size of feature maps in the pooling layers is smaller than that of the convolutional layers. To keep them the same size, it needs to pad $F-1$ times zeros in each dimension of $\delta_i^{(l+1)}$. According to (16), the error term in pooling layers is

$$\Re\big(\delta_k^{(l)}\big) = \sum_i\big[\Re\big(w_{ik}^{(l+1)}\big)*\Re\big(\delta_i^{l+1}\big)+\Im\big(w_{ik}^{(l+1)}\big)*\Im\big(\delta_i^{l+1}\big)\big] \tag{20}$$

$$\Im\big(\delta_k^{(l)}\big) = \sum_i\big[\Re\big(w_{ik}^{(l+1)}\big)*\Im\big(\delta_i^{l+1}\big)-\Im\big(w_{ik}^{(l+1)}\big)*\Re\big(\delta_i^{l+1}\big)\big]. \tag{21}$$

Like (19), $\delta_k^{(l)}$ is the sum of the real and imaginary parts.

After computing the error term of each layer, the partial derivative of loss function with respect to weights and bias is computed as (7) and (8) and (11).

Neurons in fully connected layers are formed in 1-D array, and its parameters are updated by

$$w_{ik}^{(l+1)}[t+1] = w_{ik}^{(l+1)}[t] + \eta\delta_i^{(l+1)}\overline{O_i^{(l)}} \tag{22}$$

$$b_i^{(l+1)}[t+1] = b_i^{(l+1)}[t] + \eta\delta_i^{(l+1)}. \tag{23}$$

In other layers, neurons are formed in 2-D array

$$\begin{aligned}&w_{ik}^{(l+1)}[t+1]\\ &= w_{ik}^{(l+1)}[t] + \eta\delta_k^{(l+1)} * \overline{O_k^{(l-1)}}\\ &= w_{ik}^{(l+1)}[t] + \eta\sum_{x,y}\delta_k^{(l)}(x,y)\cdot\overline{O_k^{(l-1)}}(x-u,y-v)\end{aligned} \tag{24}$$

$$\begin{aligned}&b_i^{(l+1)}[t+1]\\ &= b_i^{(l+1)}[t] + \eta\delta_i^{(l+1)} = b_i^{(l+1)}[t] + \eta\sum_{x,y}\delta_k^{(l+1)}(x,y).\end{aligned} \tag{25}$$

The parameters are adjusted iteratively until the precision between the target and the network output meets the requirement.

III. Application to SAR Image Classification

In this section, we present the detailed implementation of the proposed CV-CNN for SAR/POLSAR image classification. SAR/POLSAR data preparation is first introduced, and the coherency matrix is adopted as the input features, followed by the detail configuration of the CV-CNN.

A. POLSAR Data Adaption

POLSAR images are acquired by the polarimetric radar system, and each resolution cell of the basic SLC format is expressed by a 2×2 complex scattering matrix

$$S = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \tag{26}$$

where H, V denotes the orthogonal horizontal/vertical polarization bases. S_{pq} is the backscattering coefficient of p and q which are the polarization scattered and incident field, respectively. In monostatic case, S_{HV} and S_{VH} are often assumed identical under the reciprocity theorem. Thus, the matrix can be reduced to a 3-D scattering vector $\mathbf{k}$.

Using the Pauli decomposition [29], $\mathbf{k}$ can be expressed as

$$\mathbf{k} = \frac{1}{\sqrt{2}}[S_{HH} + S_{VV} \;\; S_{HH} - S_{VV} \;\; 2S_{HV}]^T. \tag{27}$$

The coherency matrix of POLSAR data for the multilook case is obtained as

$$\mathbf{T} = \frac{1}{L}\sum_{i=1}^{L} \mathbf{k}_i \mathbf{k}_i^H = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \tag{28}$$

where L is the number of looks. The superscript T denotes the transpose and H denotes the conjugate transpose. Apparently, the coherency matrix $\mathbf{T}$ is a Hermitian matrix whose diagonal elements are real number, while off-diagonal elements are complex. Hence, the upper triangular $\{T_{11}, T_{12}, T_{13}, T_{22}, T_{23}, T_{33}\}$ of $\mathbf{T}$ matrix were adopted as the input data.

For pixel-by-pixel classification, each pixel is fully represented by a local patch defined by a neighborhood window of size $m_1 \times m_2$. It thus captures not only the polarimetric characteristics but also the spatial imaging pattern surrounding the center pixel, and the CV-CNN is designed to extract these features. For a large POLSAR image, applying the sliding window of $m_1 \times m_2$ would generate a significant amount of patches which can serve as the training and validation samples. The input layer to the CV-CNN is now a tensor of size $m_1 \times m_2 \times 6$, where 6 is the number of channels extracted from $\mathbf{T}$ matrix.

In order to gain a higher classification accuracy, preprocessing of the data is necessary. The average T_{ave} and the standard deviation T_{std} of each channel are first calculated from the training data, and each channel of all data is normalized. Taking T_{11} as an example of the real channels

$$T_{11\_\text{ave}} = \frac{\sum_{i=1}^{n} T_{11}(i)}{n} \tag{29}$$

$$T_{11\_\text{std}} = \sqrt{\frac{\sum_{i=1}^{n} (T_{11}(i) - T_{11\_\text{ave}})^2}{n}} \tag{30}$$

and the normalized data are

$$T'_{11} = \frac{T_{11} - T_{11\_\text{ave}}}{T_{11\_\text{std}}}. \tag{31}$$

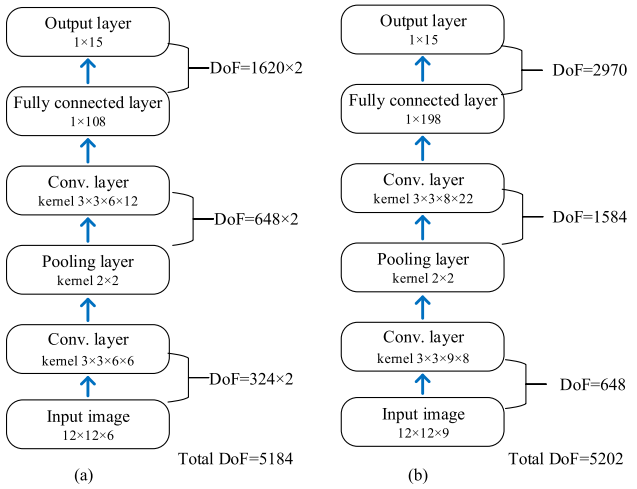


Fig. 2. Overall architecture of (a) CV-CNN and (b) RV-CNN for SAR image classification.

Taking T_{12} as an example of the complex channels

$$T_{12\_\text{ave}} = \frac{\sum_{i=1}^{n} T_{12}(i)}{n} \tag{32}$$

$$T_{12\_\text{std}} = \sqrt{\frac{\sum_{i=1}^{n} (T_{12}(i) - T_{12\_\text{ave}})\overline{(T_{12}(i) - T_{12\_\text{ave}})}}{n}} \tag{33}$$

and the final normalized data are

$$T'_{12} = \frac{T_{12} - T_{12\_\text{ave}}}{T_{12\_\text{std}}}. \tag{34}$$

In CV-CNN, the three real input channels were treated in the same way as the other three complex channels. Because of the property of convolution, the feature maps will be complex, as well as the weights of all layers.

B. Configuration of CV-CNN

Fig. 2(a) presents the architecture of the proposed CV-CNN. Besides the input and output layers, it consists of two convolutional layers, one pooling layer and one fully connected layer.

The input layer has a size of $12 \times 12 \times 6$, which means that the size of a local patch is 12×12 and the number of channels is 6. The size of feature maps will be shrunk due to convolution and pooling operations. In order to have certain depth of network, it needs zero padding on each dimension of the input layer if the size of input data is less than 12×12. The input image is filtered by a bank of six convolution filters of size $3 \times 3 \times 6$ with stride 1 in the first convolutional layer, resulting in six feature maps of size 10×10. Followed by an average pooling layer, with a pooling size of 2×2 and stride of 1, its size then becomes 5×5. The filter size of the second convolutional layer is $3 \times 3 \times 6 \times 12$, producing 12 feature maps with size 3×3. Then, reshaping the 3-D feature maps to a 1-D vector contains 108 neurons as the fully connected layer. Followed by the output layer containing c neurons which are CV, c is equal to the classification classes. c is 15 in the Flevoland data set.

Conventionally, the degree of freedom (DoF) of CV-CNN is twice that of a counterpart RV-CNN. In order to make a

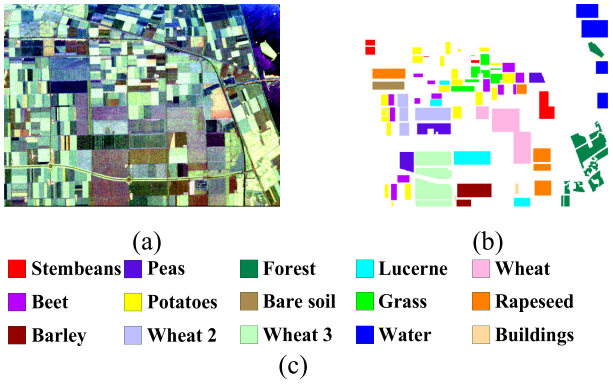


Fig. 3. Flevoland data set. (a) Pauli RGB of POLSAR. (b) Ground truth of (a). (c) Legend of the ground truth.

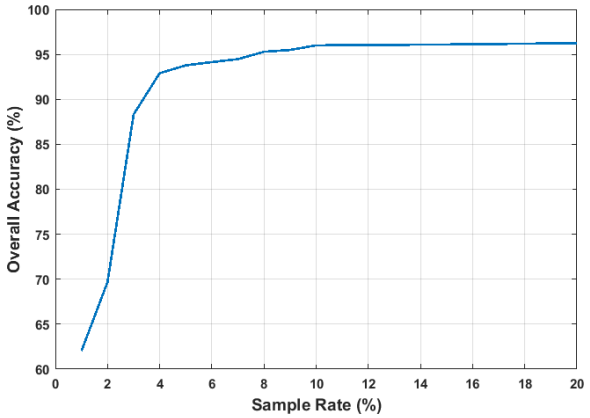


Fig. 4. Flevoland OA given a different sample rate.

fair comparison, a RV-CNN architecture as shown in Fig. 2(b) is designed to have the same DoF of the proposed CV-CNN. The 9-D input data consist of three real channels and the real and imaginary parts of the other three complex channels in the matrix $\mathbf{T}$, which has equivalent information content with CV-CNN.

IV. Experimental Results and Discussion

In this section, the performance of the proposed algorithm is demonstrated and analyzed with two POLSAR data sets. Airborne SAR (AIRSAR) data of an agriculture area over Flevoland in The Netherlands and electronically steered array radar (ESAR) data over Oberpfaffenhofen in Germany are analyzed in the following experiments. The overall accuracy (OA) and the confusion matrix are used to evaluate the performance of CV-CNN. Moreover, a RV-CNN with almost the same calculation freedom is used to compare with the performance of CV-CNN.

A. Experiment on Flevoland Data Set

The first experiment was carried out on the data set over Flevoland, which is an agriculture area of The Netherlands. It is a subset of an L-band, full POLSAR image, acquired by the NASA/Jet Propulsion Laboratory AIRSAR platform in 1989 during MAESTRO-1 Campaign [30], and is widely used as a benchmark data for POLSAR data classification study. An RGB image formed by the intensities with the Pauli decomposition is shown in Fig. 3(a), and its size is 1024×750 pixels. The ground truth is shown in Fig. 3(b) [31]. There are in total 15 identified classes including stembeans, peas, forest, lucerne, three types wheat, beet, potatoes, bare soil, grass, rapeseed, barley, water, and a small amount of buildings. Fig. 3(c) is the legend of the ground truth. Note that for small regions such as buildings, a zero-padding strategy is employed. For example, a 12×12 window centered at a building pixel near the boundary will be zero padded for those pixels outside the region.

In order to evaluate its influence and choose an optimal sampling rate, we conduct sensitivity analysis on varying sampling rate, and the results are shown in Fig. 4.

As can be seen from Fig. 4, when the sampling rate is 1%, the OA is about 62%, and the accuracy increasing quickly to 93% when the sampling rate approaching 4%, which indicates the effectiveness and efficiency of the proposed CV-CNN. When the sampling rate is greater than 10%, the accuracy becomes stable at 96%. Therefore, 10% sampling rate is enough for the Flevoland data set. In the experiment, the samples are separated into two portions, i.e., 9% for training and 1% for validation, respectively. All pixels that have corresponding ground-truth map are used as testing data.

Apparently, classification accuracy increases as training sample rate increases and approaches a saturation rate at certainly level. Our experiences suggest that the saturation level of sampling rate is specific to the data, task, and network structure. When the distribution of the same class is homogeneous and there is a large difference between classes, even a low training sample rate may achieve a good performance. For the classification of small number of classes, a lower sample rate is generally acceptable. On the other hand, the more the complex network structure, the more the needs of training data. There has several published literature discussing the issue of sampling rate in SAR image classification. Hou *et al.* [8] selected about 5% training samples per class. Guo *et al.* [32] used 10% training sample rate. Jiao and Fang [33] adopted 5% training sampling rate. Hence, we suggest a general range of 5%–10% for sample rate in real applications.

The architecture of the networks is shown in Fig. 2(a). Hyperparameters were chosen as follows. The learning rate η is 0.5, and the batchsize is 100 with 50 training epochs. Fig. 5(a) shows the final classification for the whole data set, and Fig. 5(b) shows the results of the areas with the ground truth. Fig. 5(c) and (d) shows the results of the comparing RV-CNN.

As shown in Fig. 5, the classification results are in good agreement with the ground truth at the first glance. To evaluate the performance of the proposed method, the comparison is made between Figs. 3(b) and 5(b). Most pixels are correctly classified, especially lucerne, water, and bare soil.

Table I(a) lists the classification accuracy on the whole ground truth of each class of two models, and Table I(b) lists the corresponding confusion matrix between each class of CV-CNN.

The OA of CV-CNN and RV-CNN are 96.2% and 95.3%, respectively. Note that a considerable improvement of factor of 1.2 in terms of classification error rate has achieved by extending CNN from real to complex.

TABLE I

(a) CLASSIFICATION ACCURACY OF THE WHOLE GROUND TRUTH (%). (b) CONFUSION MATRIX OF THE WHOLE GROUND TRUTH (%) FOR CV-CNN

(a)

Class	CV-CNN	RV-CNN
Stem beans	98.8	97.5
Peas	98.7	97.4
Forest	96.8	96.0
Lucerne	98.1	94.5
Wheat	95.0	93.5
Beet	97.6	97.8
Potatoes	96.7	95.6
Bare soil	98.8	99.9
Grass	90.0	94.3
Rapeseed	92.0	92.1
Barley	94.5	86.2
Wheat2	94.2	97.2
Wheat3	96.6	95.6
Water	99.4	98.5
Buildings	83.2	80.0
Overall Accuracy	96.2	95.3
Overall Error	3.8	4.7
OA with Postprocessing	97.7	97.2

(b)

%	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	98.8	0	0.3	0.1	0	0.4	0.2	0	0.2	0	0	0	0	0	0
2	0	98.7	0	0	0.7	0.4	0.1	0	0	0	0	0	0	0	0
3	0.2	0	96.8	0	0	0.2	2.8	0	0	0	0	0	0	0	0
4	0.1	0	0	98.1	0	0.3	0	0	1.2	0	0.3	0	0	0	0
5	0	0.2	0	0	95.0	0	0	0	0	1.8	0	2.7	0.3	0	0
6	0	0.6	0	0	0	97.6	1.6	0	0.2	0	0	0	0	0	0
7	0	0	1.6	0	0	1.7	96.7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	98.8	0	0.3	0	0	0	0.9	0
9	0	0	0	7.4	0	0.6	0	0	90.0	2.0	0	0	0	0	0
10	0	0.2	0	0	3.6	0.2	0	0	0.3	92.0	0	3.0	0.7	0	0
11	0.1	0	0	0.5	0	0.9	0	0.1	3.9	0	94.5	0	0	0	0
12	0	0	0	0	1.8	0	0	0	0.2	3.3	0	94.2	0.5	0	0
13	0	0	0	0.4	0.9	0	0	0	0.8	1.0	0	0.3	96.6	0	0
14	0	0	0	0	0	0	0	0.2	0	0	0.4	0	0	99.4	0
15	16.4	0	0	0	0	0.4	0	0	0	0	0	0	0	0	83.2

*Note: Each row in the confusion matrix denotes the actual target class, and each column represents the class predicted by the network. 1-15 are corresponding to the types shown in Fig.3. (c): 1 Stem beans; 2 Peas; 3 Forest; 4 Lucerne; 5 Wheat; 6 Beet; 7 Potatoes; 8 Bare soil; 9 Grass; 10 Rapeseed; 11 Barley; 12 Wheat2; 13 Wheat3; 14 Water; 15 Buildings

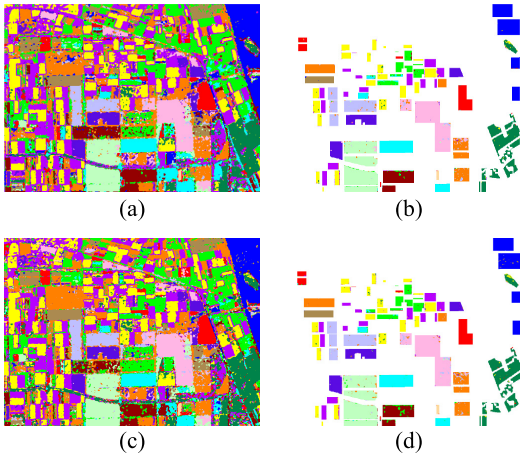
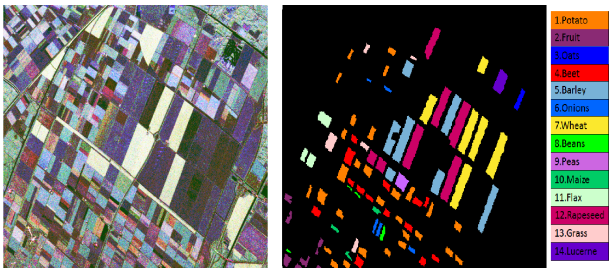


Fig. 5. Classification results of the proposed algorithm on the first data set. (a) Result of whole map classification. (b) Result overlaid with the ground-truth map. (c) and (d) Results of the RV-CNN.

From Table I(a), it can be seen that the majority classes have a correct rate higher than 95%, and all higher than 90% except buildings. The accuracy values of rapeseed, grass, and buildings are lower. Table I(b) shows that 6.6% of the rapeseed were misclassified into wheat and wheat2. From the original POLSAR data, it is found that the pixel value of rapeseed is close to that of wheat and wheat2. The misclassification between two similar types is expected. Less training samples lead to the poor classification of buildings.

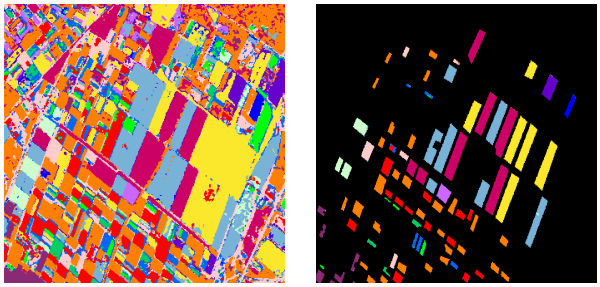
The reason why CV-CNN obtained a better performance may be because CV-CNN utilized the phase information, which is a crucial characteristic in SAR images. To verify this point, the histogram of different species was investigated, and the results show that the phase distribution varies obviously with species, but stable for the same species in a different area. Therefore, phase information can be regarded as an effective feature for the POLSAR image classification.

Another experiment is carried out to investigate the influence of network structure with one more pooling layer. The kernel size of the second convolutional layer was reduced from 3×3 to 2×2, and one more pooling layer was added. The overall test accuracy 90% is lower than that of the previous experimental result 96.2%. The reason is that the pooling operation reduces the size of the feature map, and some information is then lost, which leads to a poorer performance.



(a) (b)

Fig. 6. Flevoland dataset 2. (a) Pauli RGB of POLSAR. (b) Ground-truth map and legend of the ground truth.



(a) (b)

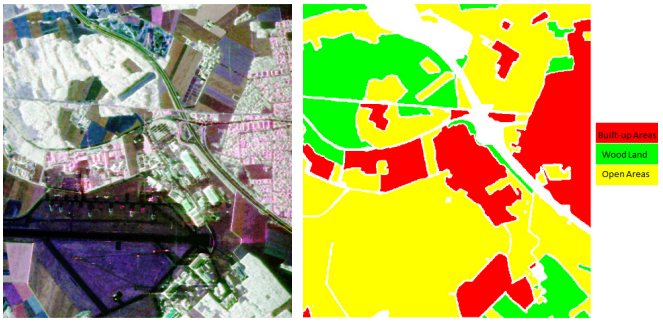
Fig. 7. Classification results of the proposed algorithm on the first data set. (a) Result of whole map classification. (b) Result overlaid with the ground-truth map.

In some recently published studies, postprocessing techniques are further applied to filter out some isolated spots in the classification map, which can naturally enhance the OA. For example, Liu *et al.* [34] obtained a state-of-the-art 97.57% accuracy on the Flevoland classification based on the Wishart DBN and local spatial information as postprocessing. Guo *et al.* [32] obtained 95.32% accuracy with similar approaches. We further applied majority vote postprocessing to the final classification map; it shows that the OA can be improved further to 97.7% for CV-CNN.

B. Performance Evaluation on Benchmark Data Set

In order to evaluate the performance of the proposed CV-CNN against some state-of-the-art classifiers [35], [36], the benchmark data set of an L-band AIRSAR data obtained in 1991 over Flevoland, The Netherlands is used for cross-comparison. The selected benchmark AIRSAR image is shown in Fig. 6(a), with a size of 1020 × 1024. The ground-truth map is shown in Fig. 6(b) which is adapted from [37]. According to the ground truth, there are 14 classes. The legend of the ground truth is given along with the ground-truth map. A 12 × 12 sliding window is chosen to generate the samples from the original area within the ground truth. About 850 samples per class were chosen randomly as training samples, and the residual labeled data are used for testing.

The architecture of the networks is the same as that of the previous experiment. Hyperparameters were chosen as follows. The learning rate η is 0.8, and the batch size is 100 with 60 training epochs. Fig. 7(a) shows the final classification for the whole data set, and Fig. 7(b) shows the results of the areas with the ground truth.



(a) (b)

Fig. 8. Oberpfaffenhofen data set. (a) Pauli RGB of POLSAR. (b) Ground-truth map and legend.

As shown in Fig. 7, the classification results are in good agreement with the ground truth. To evaluate the performance of the proposed method, the comparison is made between Figs. 6(b) and 7(b). Most pixels are correctly classified, especially potato, barley, wheat, peas, and rapeseed. Table II lists the accuracy of classification of the entire ground-truth area. From Table II, it can be seen that the majority classes have a correct rate close to 100%, and all classes are higher than 90%. While the RV-CNN is worse in onions and beans. This particular case shows that CV-CNN could improve error rate by a significant factor over RV-CNN.

In Table II, it also gives the result of Gao *et al.* [35] using only L-band and using all P-, L-, and C-band, respectively. They employed the Wishart mixture model as classifier. The sixth column is the result of Liu *et al.* [36] using the combination of all C-, L-, and P-bands. The method that they proposed is based on tensor representation and multilinear subspace learning. When compared with the single L-band performance, the CV-CNN performs much better than the method proposed by Gao *et al.* [35] Although using only the L-band, the CV-CNN matches with the performance of Liu *et al.* [36] achieved by using all three bands.

C. Experiment on Oberpfaffenhofen Data Set

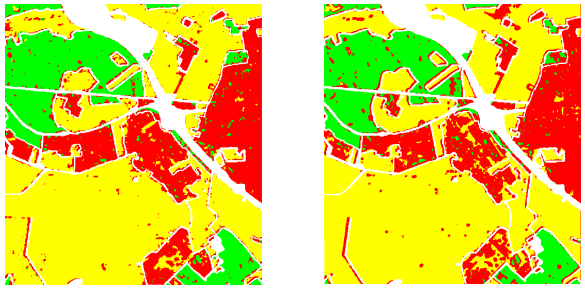
Fig. 8(a) shows the ESAR L-band, multilook data over Oberpfaffenhofen, Germany [30]. The size of the image is 1300×1200 pixels. The ground truth is shown in Fig. 8(b) [38], which is acquired manually according to the corresponding optical remote-sensing imagery in Google Earth. There are three basic classes: built-up areas, wood land, and open areas, and the legend is given in Fig. 8(b). The void areas are the unknown class or the boundaries between classes, which are neglected in both training and validation samples. Each sample size is 12 × 12 pixels. Because of the large size and small number of classes, the sampling rate is 1% in this experiment. Thus, the training samples and the validation samples are 0.9% and 0.1%, respectively.

Parameters are set as following: the learning rate η is 0.8, and the batch size is 100 with 60 training epochs. Fig. 9(a) shows the classification overlaid with the ground-truth map. Fig. 9(b) shows the results of RV-CNN.

Comparing Fig. 9 with the ground truth, the CV-CNN performs well in the majority areas. It can be seen that

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY OVER THE WHOLE GROUND-TRUTH AREA (%)

Class	CV-CNN Using L band	RV-CNN Using L band	Gao *et al.* [35] Using L band	Gao *et al.* [35] Using PLC band	Liu *et al.* [36] Using PLC band
1Potato	99.8	98.1	89.4	99.7	99.6
2Fruit	98.3	96.2	91.7	100.0	99.7
3Oats	98.9	96.2	92.0	98.2	99.7
4Beet	96.1	96.4	82.1	92.5	95.9
5Barley	99.6	99.6	84.2	97.3	99.4
6Onions	93.2	70.8	65.0	94.8	93.4
7Wheat	99.9	99.0	91.5	99.9	99.5
8Beans	90.7	84.6	86.8	99.2	99.7
9Peas	99.0	94.4	95.6	99.9	100
10Maize	98.2	90.2	77.2	93.3	89
11Flax	97.0	92.9	96.9	98.1	97.3
12Rapeseed	99.8	98.8	90.1	99.9	99.6
13Grass	96.6	92.2	86.7	97.7	98.2
14Lucerne	99.2	90.9	94.8	99.8	99.9
Overall Accuracy	99.0	97.3	87.4	97.9	98.9
Overall Error	1	2.7	12.6	2.1	1.1
Sample rate	10%	10%	/	/	5.2%



(a) (b)

Fig. 9. Classification results of the Oberpfaffenhofen data set. (a) Result overlaid with the ground-truth map. (b) Corresponding results of the RV-CNN.

TABLE III
(a) CLASSIFICATION ACCURACY OF THE WHOLE IMAGE (%).
(b) CONFUSION MATRIX OF THE WHOLE IMAGE (%) FOR CV-CNN

(a)

Class	CV-CNN	RV-CNN
Built-up areas	91.3	85.7
Wood land	92.2	85.2
Open areas	94.6	93.4
Overall Accuracy	93.4	89.9
Overall Error	6.6	10.1

(b)

Class	Built-up areas	Wood land	Open areas
Built-up areas	91.3	3.2	5.5
Wood land	7.2	92.2	0.6
Open areas	5.2	0.2	94.6

(*Note: Each row in the confusion matrix denotes the actual target class, and each column represents the class predicted by the network.)

this data set is more complicated than the first one, since there are similarities between the built-up areas and the wood land in some extent for Fig. 8(a). By comparing the final classification results in Fig. 9(a) and (b), the proposed CV-CNN matches much better with the ground truth. In this case, an improvement factor of 1.5 in terms of error rate is achieved by extending RV-CNN to CV-CNN.

A detail comparison is presented in Table III. Table III(a) lists the classification accuracy on the whole image of each class of two models, and Table III(b) lists the corresponding confusion matrix between each class of CV-CNN. The proposed CV-CNN is successful for classification with the OA of 93.4%, while the accuracy of the RV-CNN is 89.9%. Again, we observe a significant improvement of a factor of nearly 1.5 in the error rate.

V. CONCLUSION

CNN has achieved great success in computer vision areas. This study aims to apply CNN to SAR image processing. In order to take advantage of phase information, we present the CV-CNN which extends RV-CNN to complex domain. Both neurons and weights are represented by complex numbers. A complex version of backpropagation is also proposed, which can effectively train deep CV-CNN with massive complex SAR images. A novel POLSAR image classification scheme based on CV-CNN is then proposed and evaluated with real SAR data. Results show that a significant improvement in terms of classification accuracy can be achieved by converting RV-CNN to CV-CNN. It opens up a wide range of applications for CV-CNN in SAR image interpretation.

ACKNOWLEDGMENT

The source code of this paper is available at: https://github.com/fudanxu/CV-CNN.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[3] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.
[4] K. He, X. Zhang, S. Ren, and J. Sun. (Feb. 2015). "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification." [Online]. Available: https://arxiv.org/abs/1502.01852
[5] 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.
[6] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[7] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Jun. 2016.
[8] B. Hou, H. D. Kou, and L. C. 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.
[9] 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.
[10] A. Hirose, Ed., *Complex-Valued Neural Networks: Advances and Applications*. Hoboken, NJ, USA: Wiley, Apr. 2013.
[11] A. Hirose, Y. Asano, and T. Hamano, "Developmental learning with behavioral mode tuning by carrier-frequency modulation in coherent neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1532–1543, Nov. 2006.
[12] F. Shang and A. Hirose, "Quaternion neural-network-based PolSAR land classification in poincare-sphere-parameter space," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5693–5703, Sep. 2014.
[13] J. A. Kong, A. S. Swartz, H. A. Yueh, L. M. Novak, and R. T. Shin, "Identification of terrain cover using the optimum polarimetric classifier," *J. Electromagn. Waves Appl.*, vol. 2, no. 2, pp. 171–194, Jan. 1988.
[14] J.-S. Lee, M. R. Grunes, and R. Kwok, "Classification of multi-look polarimetric SAR imagery based on complex Wishart distribution," *Int. J. Remote Sens.*, vol. 15, no. 11, pp. 2299–2311, Jul. 1994.
[15] A. Freeman, J. Villasenor, J. D. Klein, P. Hoogeboom, and J. Groot, "On the use of multi-frequency and polarimetric radar backscatter features for classification of agricultural crops," *Int. J. Remote Sens.*, vol. 15, no. 9, pp. 1799–1812, 1994.
[16] Y. Kouskoulas, F. T. Ulaby, and L. E. Pierce, "The Bayesian hierarchical classifier (BHC) and its application to short vegetation using multifrequency polarimetric SAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 2, pp. 469–477, Feb. 2004.
[17] J.-S. Lee, M. R. Grunes, T. L. Ainsworth, L.-J. Du, D. L. Schuler, and S. R. Cloude, "Unsupervised classification using polarimetric decomposition and the complex Wishart classifier," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 5, pp. 2249–2258, Sep. 1999.
[18] 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.
[19] V. John, K. Yoneda, Z. Liu, and S. Mita, "Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching," *IEEE Trans. Comput. Imag.*, vol. 1, no. 3, pp. 159–173, Sep. 2015.
[20] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 315–323.
[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[22] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, Sep. 2014, pp. 818–833.
[23] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Proc. ICARCV*, Dec. 2014, pp. 844–848.
[24] M. M. Lau, K. H. Lim, and A. A. Gopalai, "Malaysia traffic sign recognition with convolutional neural network," in *Proc. Int. Conf. Digit. Signal Process.*, 2015, pp. 1006–1010.
[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
[26] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Berlin, German: Springer, 2012, pp. 437–478.
[27] R. Hänsch and O. Hellwich, "Classification of polarimetric SAR data by complex valued neural networks," in *Proc. ISPRS Hannover Workshop, High-Resolution Earth Imag. Geospatial Inf.*, vol. XXXVII. Hannover, Germany, 2009.
[28] J. Bouvrie, "Notes on convolutional neural networks," Center Biol. Comput. Learn., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 2006, pp. 38–44.
[29] S. R. Cloude and E. Pottier, "A review of target decomposition theorems in radar polarimetry," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 2, pp. 498–518, Mar. 1996.
[30] *Earth Online*. [Online]. Available: http://envisat.esa.int/POLSARpro/datasets.html2
[31] P. Yu, A. K. 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, Apr. 2012.
[32] Y. Guo, S. Wang, C. Gao, D. Shi, D. Zhang, and B. Hou, "Wishart RBM based DBN for polarimetric synthetic radar data classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1841–1844.
[33] 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.
[34] 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.
[35] W. Gao, J. Yang, and W. T. Ma, "Land cover classification for polarimetric SAR images based on mixture models," *Remote Sens.*, vol. 6, no. 5, pp. 3770–3790, Apr. 2014.
[36] C. Liu, J. Yin, J. Yang, and W. Gao, "Classification of multi-frequency polarimetric SAR images based on multi-linear subspace learning of tensor objects," *Remote Sens.*, vol. 7, no. 7, pp. 9253–9268, Jul. 2014.
[37] 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.
[38] 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.

Zhimian Zhang received the B.E. degree in electronic engineering from Anhui University, Hefei, China, in 2015. She is currently pursuing the M.S. degree with the Key Laboratory of Information Science of Electromagnetic Waves, Fudan University, Shanghai, China.

Her research interests include the application of machine learning and deep learning algorithms to synthetic aperture radar.

Haipeng Wang (S'03–M'07–SM'16) received the B.E. and M.S. degrees in mechanical and electronic engineering from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree in environmental systems engineering from the Kochi University of Technology, Kochi, Japan, in 2006.

He was a Visiting Researcher with the Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan, in 2008. He is currently an Associate Professor with the Key Laboratory of Electromagnetic Wave Information Science (MoE), Department of Communication Science and Engineering, School of Information Science and Engineering, Fudan University, Shanghai, China. His research interests include signal processing, synthetic aperture radar (SAR) image processing and analysis, speckle statistics, and applications to forestry and oceanography, machine learning and its applications to SAR images.

Dr. Wang has been a member of the Technical Program Committee of the IEEE Geoscience and Remote Sensing Symposium since 2011. He was a recipient of the Dean Prize of the School of Information Science and Engineering, Fudan University, in 2009.

Feng Xu (S'06–M'08–SM'14) received the B.E. degree (Hons.) in information engineering from Southeast University, Nanjing, China, in 2003, and the Ph.D. degree (Hons.) in electronic engineering from Fudan University, Shanghai, China, in 2008.

From 2008 to 2010, he was a Post-Doctoral Fellow with the NOAA Center for Satellite Application and Research, Camp Springs, MD, USA. From 2010 to 2013, he was with Intelligent Automation Inc., Rockville, MD, USA, while partly involved as a Research Scientist with the NASA Goddard Space Flight Center, Greenbelt, MD, USA. In 2012, he was selected into the China's Global Experts Recruitment Program, and subsequently returned to Fudan University in 2013, where he is currently a Professor with the School of Information Science and Technology and the Vice Director of the Key Laboratory for Information Science of Electromagnetic Waves (MoE). He has authored more than 40 papers in peer-reviewed journals, co-authored two books, and holds two patents, among many conference papers. His research interests include fast electromagnetic modeling for complicated target and environments, intelligent interpretation of synthetic aperture radar (SAR) images, inverse scattering tomography, and SAR remote-sensing applications in earth observation.

Dr. Xu was awarded the second-class National Nature Science Award of China, among other honors, in 2011. He was a recipient of the Early Career Award of the IEEE Geoscience and Remote Sensing Society in 2014 and the SUMMA Graduate Fellowship in the advanced electromagnetics area in 2007. He currently serves as the Associate Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. He is the Founding Chair of the IEEE GRSS Shanghai Chapter.

Ya-Qiu Jin (SM'89–F'04) received the B.E. degree in electrical engineering and computer science from Peking University, Beijing, China, in 1970, and the M.S., E.E., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1982, 1983, and 1985, respectively.

In 1985, he joined the Atmospheric Environmental Research, Inc., Cambridge, MA, USA, as a Research Scientist. From 1986 to 1987, he was a Research Associate Fellow with the City University of New York, New York, NY, USA. In 1993, he joined the University of York, York, U.K., as a Visiting Professor, sponsored by the U.K. Royal Society. He is currently the Te-Pin Professor and the Director of the Key Laboratory for Information Science of Electromagnetic Waves (MoE), Fudan University, Shanghai, China. He has authored more than 740 papers in refereed journals and conference proceedings and 14 books, including the *Polarimetric Scattering and SAR Information Retrieval* (Wiley and IEEE, 2013), the *Theory and Approach of Information Retrievals From Electromagnetic Scattering and Remote Sensing* (Springer, 2005), and the *Electromagnetic Scattering Modeling for Quantitative Remote Sensing* (World Scientific, 1994). His research interests include electromagnetic scattering and radiative transfer in complex natural media, microwave satellite-borne remote sensing, as well as theoretical modeling, information retrieval and applications in earth terrain and planetary surfaces, and computational electromagnetics.

Dr. Jin received the IEEE GRSS Distinguished Achievement Award in 2015, the IEEE GRSS Education Award in 2010, the China National Science Prize in 1993 and 2011, the Shanghai Sci/Tech Gong-Cheng Award in 2015, and the first-grade MoE Science Prizes in 1992, 1996, and 2009, among many other prizes. He was also awarded by the Senior Research Associateship in NOAA/NESDIS by the USA National Research Council in 1996. From 2005 to 2012, he was an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, a member of the IEEE GRSS AdCom, and the Chair of the IEEE Fellow Evaluation of GRSS from 2009 to 2011. He was a Co-Chair of TPC for IGARSS2011 in Vancouver, Canada, and will be the Co-General Chair for IGARSS2016 in Beijing, China. He is the Academician of the Chinese Academy of Sciences, and a fellow of the TWAS (the Developing Countries Academy of Sciences), the International Academy of Astronautics, and the Electromagnetic Academy. He is an IEEE GRSS Distinguished Speaker and an Associate Editor of the IEEE ACCESS.