Automatic Design of CNNs via Differentiable Neural Architecture Search for PolSAR Image Classification

Hongwei Dong, Bin Zou, Senior Member, IEEE, Lamei Zhang, Senior Member, IEEE, and Siyu Zhang

Abstract—Convolutional neural networks (CNNs) have shown good performance in polarimetric synthetic aperture radar (PolSAR) image classification. Excellent hand-crafted CNN architectures incorporated the wisdom of human experts, which is an important reason for CNNs success. However, the design of architectures is a difficult problem, which needs a lot of professional knowledge as well as computational resources. Moreover, the manually designed architecture might be suboptimal, because it is only one of the thousands of unobserved but objective exists paths. Considering that the success of deep learning is largely due to its automation of the feature engineering process, how to design automatic architecture search methods to replace the hand-crafted ones is an interesting topic. In this article, the application of neural architecture search (NAS) in the PolSAR area is explored for the first time. Different from the utilization of existing methods, a PolSAR-tailored Differentiable Architecture Search (DARTS) method, called PDAS, is proposed in order to adapt NAS to the PolSAR classification. A PolSAR-tailored search space and an improved one-shot search strategy are equipped with the proposed method. By PDAS, the architecture (corresponds to the hyperparameter but not the topology) parameters can be optimized with high efficiency by a stochastic gradient descent (SGD) method. The optimized architecture parameters should be transformed into the corresponding architecture and retrained to achieve classification. In addition, a complex-valued PDAS (CVPDAS) is developed to fit the data form of PolSAR images so as to improve the performance. Experiments on three benchmark data sets show that the architectures obtained by searching have better classification performance than hand-crafted ones.

Index Terms—Automatic machine learning (AutoML), convolutional neural network (CNN), neural architecture search (NAS), polarimetric synthetic aperture radar (PolSAR) classification.

I. INTRODUCTION

POLARIMETRIC synthetic aperture radar (PolSAR) is an indispensable sensor in earth observation. It can provide abundant target information in all-weather and all-times, which is not available for other sensors. More and more attention has been paid to the development of PolSAR and the interpretation of PolSAR images. As the basis of PolSAR interpretation, PolSAR image classification has been a hot research field in recent years. For quite a long time in the past, features with clear physical scattering meanings are eager to be studied in the PolSAR classification area, i.e., target decomposition techniques. Many target decomposition-based feature extraction methods have been extensively studied [1]–[3]. Statistical-based feature extraction methods [4], [5] have also been widely used to realize better classification results. However, it is still difficult to construct complete features due to the complex imaging mechanism of PolSAR. Therefore, PolSAR image classification is still an open problem.

Deep learning [6] has made remarkable progress in natural language processing and computer vision. Its main idea is to automatically learn features through a neural network instead of hand-crafted feature filters. As the main carrier of deep learning techniques in the image processing field, convolutional neural networks (CNNs) [7] have aroused the interest of a large number of researchers since AlexNet and ImageNet Large-Scale Visual Recognition Challenge 2012 [8]. Given AlexNet’s great success, a large number of studies have been carried out to design powerful architectures so as to obtain better performance [9]. Novel architectures have greatly promoted the development of CNNs. It can be said that CNN’s current achievements are largely due to the efforts of experts on the issue of neural architecture design.

Researchers in remote sensing areas have begun to use CNNs to improve image classification due to the difficulty in designing features manually [10]–[13]. Geng et al. [13] first explored the application of deep neural networks in synthetic aperture radar (SAR) image classification. After that, many CNN-based classification methods have sprung up. They can be divided into two parts according to different research objectives: Target recognition and terrain classification. An all-convolutional architecture was proposed in [14] to achieve high-precision SAR image automatic target recognition (SAR-ATR). A Highway unit-based architecture was proposed in [15] to implement target recognition in the environment of limited training samples. They obtained the matched precision compared with the method in [14] under the premise of reducing 30% training data. Multiview SAR data were generated and a multinput CNN architecture was proposed in [16] to get the features from different views. High-order features extracted
by CNN were made into a feature dictionary, and the end-to-end training was carried out through feature metric and two-stage optimization in order to achieve small sample SAR-ATR [17]. Different from the task of target recognition, the purpose of terrain classification is to predict each pixel of an image. The general paradigm of PolSAR image terrain classification can be summarized as: The complex-valued PolSAR data are first converted into real-valued tensors. Well-designed CNNs are then used for classification. The architecture with two convolutional layers was proposed in [18] and achieved 92.46% accuracy on PolSAR Flevoland data. Transfer learning-based ensemble CNNs were used to realize scene classification on the data acquired from the P-band Chinese airborne PolSAR system [19]. In order to make up the shortages of slice-based classification methods, i.e., slow speed and lack of global information, fully convolutional networks [20] were introduced to achieve better full image classification [21], [22]. Some works studied new representations of PolSAR data, so as to improve the performance of CNN-based methods [23]–[25]. Considering the particularity of PolSAR data, complex-valued (CV) CNN (CV-CNN) architectures have also been studied to some extent [26]–[28]. In general, the recent research of CNN-based PolSAR image classification methods mainly focuses on building PolSAR-tailored network architectures, or designing architectures for the environment of limited training samples.

As can be seen from the works described above, manually designing CNNs architecture is the problem that researchers have been working on for the past few years. Although there are many experiences that can guide the design of network architectures, a new research direction is emerging, i.e., neural architecture search (NAS), to replace the classical hand-crafted architectures. NAS has many overlaps with automatic machine learning (AutoML) [29] and meta-learning [30]. Its superiority is reflected in two aspects: On the one hand, the success of deep learning is largely due to its automation of the feature engineering process. It can be expected that if this automated process is extended to the model design stage, it will be possible to further improve the performance. On the flip side, it is a costly task to design a tailored neural architecture for a certain problem. This is also the biggest difficulty in utilizing deep learning techniques to solve the problems in other fields, e.g., radar image processing. As known that network architectures are defined by the topology and hyperparameter, Zoph and Le [31] noticed that the topology and hyperparameter of a neural network could be typically specified by a variable-length string, and a recurrent network, called controller, was used to generate such strings. The controller was trained with reinforcement learning [32] to maximize the expected accuracy of the string-transformed architecture on the validation set. Although the architectures searched by evolutionary or reinforcement learning-based NAS methods [33], [34] exhibited strong performance, all candidates need to be trained repeatedly to look for the optimal architecture, which brings unaffordable computational difficulties and seriously restricts the development and application of NAS. In order to fundamentally seek a low-cost way, Liu et al. [35] relaxed the architecture representation and proposed a gradient-based search method, called Differentiable Architecture Search (DARTS). They built a continuous search space by a series of attention-based architecture parameters, thereby avoiding the expensive computational burden caused by the updates of controllers. The fly in the ointment is that the approximate solution of the bilevel optimization problem [36], [37] made DARTS unstable, so some improvements are being gradually explored [38], [39].

The objective of this article is to make full use of the ability of CNNs to improve PolSAR image terrain classification. It can be seen from the previous analysis that the obstacle to achieving this goal lies in the difficulty in designing PolSAR-tailored CNN architectures. This difficulty is partly reflected in the fact that the design of architectures requires considerable expertise in neural networks, which is lacking in the PolSAR area. However, the authors believe that with the efforts of researchers [25], [26], this will not be a stumbling block for future development. In fact, the fundamental problem to be solved is the unsatisfactory performance of hand-crafted architectures. Naturally, we consider seeking help from NAS to handle this problem, and a tailored and cost-effective NAS method is expected. Inspired by previous works and according to the characteristics of the PolSAR classification task, a PolSAR-tailored DARTS method, called PDAS, is proposed in this article to automatically seek high-performance CNN architectures for better performance. More specifically, a PolSAR-tailored search space is defined whose variables are the hyperparameters of convolutions, i.e., spatial size and output depth. The reason for such search space comes from the observation that hyperparameters have a significant impact on the performance of PolSAR slices-based classification. Continuous relaxation is applied to the proposed search space, so gradient-based one-shot architecture search can be implemented by PDAS to greatly save computational resources. Moreover, the search strategy of PDAS is improved to avoid the repeated operations as well as solve the bilevel optimization problems accurately [35], [38]. To the best of our knowledge, this is the first work which tries to look for the CNN architectures through automatic search instead of manual design, in the radar image processing area. Compared with the hand-crafted architectures, the proposed method has better interpretability and stronger performance. Furthermore, a complex-valued version of PDAS is proposed to fit the data format of PolSAR images. It is noteworthy that the proposed methods can be used to not only search for suitable PolSAR classification architectures from scratch but also improve the existing methods in a plug-and-play manner. The major contributions of this article can be summarized as follows.

1) The improvement of PolSAR classification with the help of NAS is explored for the first time.
2) A novel PolSAR-tailored DAS method, i.e., PDAS, and its complex-valued version are proposed.
3) Experimental results on three widely used PolSAR benchmark data sets demonstrate the validity of the proposed methods.

The rest of this article is organized as follows: The proposed method and its complex-valued version are introduced in Section II. Experimental results and discussion are presented
An extensively used activation function, i.e., rectified linear unit (ReLU), can be defined as

\[ y_j = \sum_{i=1}^{c_{in}} x_i \ast w_{i,j} + b_j \]  

(1)

where \( x_i, y_j \) denote the \( i \)th input and \( j \)th output feature maps, respectively. \( c_{in} \) denotes the number of input feature maps. \( w \) and \( b \) are the learnable parameters of the convolution. \( \ast \) represents the standard convolution operator. It can be seen that the convolution is a linear operation, so a nonlinear activation after it is needed to improve the model capability.

A. CNNs for PolSAR Classification

Powerful data fitting ability of CNNs has been valued with the continuous increase of available data. Convolution, nonlinear activation, pooling, and fully connected are the basic units of CNNs. The vanilla 2-D convolution operations can be defined as

\[ y_j = \sum_{i=1}^{c_{in}} x_i \ast w_{i,j} + b_j \]  

(1)

where \( x_i, y_j \) denote the \( i \)th input and \( j \)th output feature maps, respectively. \( c_{in} \) denotes the number of input feature maps. \( w \) and \( b \) are the learnable parameters of the convolution. \( \ast \) represents the standard convolution operator. It can be seen that the convolution is a linear operation, so a nonlinear activation after it is needed to improve the model capability.

By stacking the aforementioned operations, a CNN architecture can be obtained to extract hierarchical features. An extensively used activation function, i.e., rectified linear unit (ReLU), can be defined as

\[ \sigma_{ReLU}(x) = \max(0, x). \]  

(2)

As shown in Fig. 1, the aim of PolSAR image classification is to give a certain category to every pixel. First, small slices are cut around the central pixel according to the polarization coherence matrix of PolSAR data and the ground truth map. Each image slice is treated as a sample with supervisory information and the training and testing sets can be obtained from the labeled image slices. Finally, researchers need to design a suitable CNN architecture and implement training and testing on different sets.

The automation of feature engineering can be recognized as an important reason for CNN’s success. Accompanied by this is the transformation from hand-crafted features to handcrafted network architectures. This demand led to the birth of NAS [31], which aims to automate the process of architecture engineering and treats the design of suitable architectures as a search problem, like hyperparameter optimization [40].

B. Proposed PDAS

We propose a novel DARTS method, called PDAS, according to the characteristics of slice-based PolSAR classification. The first aim of PDAS is to search suitable architectures, i.e., the hyperparameters including the size of kernels and the depth of output for a particular topology. In this article, topology is fixed to a cascaded five-layer convolution network with two pooling layers. Another aim of PDAS is to search the architecture efficiently, so it draws lessons from one-shot NAS methods to a great extent. In PDAS, the architectures with different hyperparameter combinations are represented by a series of attention-based architecture parameters. Through stochastic gradient descent (SGD) on the training set, the architecture parameters can be efficiently optimized, and then the optimized architecture can be obtained. The proposal of PDAS can be divided into the following steps: definition of PolSAR-tailored search space, continuous relaxation of the search space, linear combination of convolutions, and improved one-shot architecture search. The first step reflects the particularity of PDAS, which is abstracted from the specific PolSAR classification problem. The last three steps are the improvements in the problem-solving strategy.

1) PolSAR-Tailored Search Space: The definition of the search space determines the number of architectures that can be represented in principle. In general, the search space is exponentially large or unbounded for a NAS method, which greatly limits the search speed. Considering that the size of the input PolSAR image slices is small (usually no more than \( 20 \times 20 \)), PolSAR classification used architectures usually have a shallower depth (less than ten layers and no more than three pooling) compared with optical image classification used (generally more than 100 layers). More layers means that the complexity of the topology increases exponentially, which is why the search of topologies is popular in optical image processing. However, the size of input limits the depth of PolSAR classification used architectures. Therefore, the following assumption can be made: for image slice-based PolSAR classification, the choice of hyperparameters is at least as important as the topologies to the performance of the architectures with similar model complexity.
In order to verify the assumption, numerical simulations are carried out on Flevoland PolSAR data. The experimental settings and details of the used data set are described in Section III. The involved architectures can be divided into two parts, i.e., similar complexity but different topology and same topology but different complexity. First, in order to study the effect of topology on the overall accuracy (OA) of PolSAR classification, five widely used architectures including LeNet [7], GoogLeNet [41], VGGNet [42], ResNet [9], and DenseNet [43] with similar floating point operations (FLOPs) $(1.4-2.0 \times 10^8)$ are constructed and trained with $4500$ training samples and $200$ epoch. The aforementioned architectures have two pooling layers and two fully connected layers, and the number of convolution layers they have varies from three to six. Then, four versions of LeNet are constructed and compared to study the effect of the hyperparameter. They have exactly the same topology, i.e., cascaded three convolutional layers, two pooling layers, and two fully connected layers. However, the model complexity of them is exponentially increased (the number of output feature maps of each layer is multiplied). As shown in Fig. 2, on the one side, architectures with different topologies but similar number of trainable weights have comparable performance. On the other side, changes in hyperparameters do have a great impact on model performance. It is worth noting that the LeNet-v3 in Fig. 2(b) is the architecture LeNet used in Fig. 2(a) whose output feature maps of the three convolution layers are $16$, $26$, and $32$, respectively. Compared with the importance of the two, the influence of hyperparameter is at least not less than that of topology. About $6\%$ of accuracy improvement can be obtained by changing hyperparameters, while changing the topology can only obtain less than $2\%$ in the experiments.

Based on the aforementioned analyses, we focus on looking for better combinations of hyperparameters under a specific topology in order to achieve a cost-effective architecture search. Without loss of generality, a commonly used CNN architecture [7]–[9] is chosen as the basis, which has three convolution layers, two pooling layers, and two fully connected layers. Hyperparameters of this topology constitute the search space of PDAS, i.e., the spatial size and output depth of the convolution kernels. An intuitive diagram of the search space is shown in Fig. 3.

The height and width of convolution kernels and the number of output channels are variables of the search space, which are noted as $S_h$, $S_w$, and $C$. In this way, the total number of potential architectures $N_{\text{total}}$ in the proposed search space can be obtained

$$N_{\text{total}} = \prod_{i=1}^{l} S_h^i \times S_w^i \times C^i$$

(3)

where $l$ means the number of trainable layers and $l = 5$ in this article. It is worth noting that using such a fixed topology is to verify the validity of PDAS as well as reduce the search complexity as much as possible. Searching within other kinds of architectures suitable for PolSAR classification is no doubt supported by PDAS.

2) Continuous Relaxation: After determining the appropriate search space, an efficient search strategy needs to be designed to implement an efficient architecture search. Based on this requirement, continuous relaxation proposed by DARTS [35] is applied to our search space. Let $O$ be a set of candidate operations, for the input $x$, continuous relaxation can be defined as follows:

$$\tilde{o}(x) = \sum_{o \in O} \sigma(A_o)o(x)$$

(4)

where $A_o$ notes the architecture parameters assigned to operation $o()$, and $\sigma()$ means the weight of corresponding operation generated by an activation function. From the definition, continuous relaxation can be seen as using a mixed operation to replace the categorical choice of a particular operation. Softmax activation is used over architecture parameters to generate the weights of $o()$ in [35], and (4) specified to

$$\tilde{o}(x) = \sum_{o \in O} \sigma_{\text{softmax}}(A_o)o(x)$$

$$= \sum_{o \in O} \frac{\exp(A_{o})}{\sum_{o' \in O} \exp(A_{o'})} o(x).$$

(5)

In this article, sparsemax activation [44], which can be seen as a truncated softmax, is used in PDAS to increase the sparsity of the architecture parameters. If the architecture parameters are sorted as $A_{(1)} \geq \cdots \geq A_{(K)}$, a set $k$ can be found with $k(A) := \max\{k \in [K]: 1 + kA_{(k)} > \sum_{j \leq k} A_{(j)}\}$. Then the mixed operation of PDAS can be written as

$$\tilde{o}(x) = \sum_{o \in O} \sigma_{\text{sparsemax}}(A_o)o(x)$$

$$= \sum_{o \in O} \max(0, A_o - \tau(A))o(x)$$

(6)

where $\tau(\cdot)$ is a threshold function defined as $\tau(A) = (\sum_{j \leq k(A)} A_{(j)}) - 1)/k(A)$. The vector activated by sparsemax tends to be a normalized sparse vector, which can accelerate the convergence of the search process.

Specifically, for the case of PDAS, $o(x)$ represents convolution with different hyperparameter combination, and $A = \{a, \beta\}$ where $a$ notes the architecture parameter of kernel size and $\beta$ for the one of output depth. If there are $n$ choices for kernel size and $m$ choices for output depth in a certain
layer, (6) can be expressed as

\[
\bar{\sigma}(x) = \sum_{i=1}^{m} \sigma_{\text{sparsmax}}(\alpha_i) \left( \sum_{j=1}^{n} \sigma_{\text{sparsmax}}(\beta_j) (W_{i,j} \ast x) \right)
\]

\[
= \sum_{i,j=1}^{n,m} \alpha_i^j \beta_j^i (W_{i,j} \ast x)
\]  (7)

where the weights generated by sparsmax activation are simplified as \(\alpha', \beta'\), and \(W_{i,j}\) represents corresponding convolution matrix, \(\ast\) means convolution operation. In this way, the discrete search space of PDAS can be relaxed to be continuous, and the search of architectures can be transformed into the optimization of the architecture parameters \(\alpha\) and \(\beta\).

3) Linear Combination of Convolution: Note that (4)–(7) need to calculate all possible operations and then weighted sum, which brings huge computational complexity. Especially for (7), nanometer times convolution are required, which is impractical when the product is very large. To handle this problem, the linearity of convolution is utilized. Repeated calculations can be avoided by using linear combination of convolutions instead of (7), which can be expressed as

\[
\bar{\sigma}(x) = \sum_{i,j=1}^{n,m} (\alpha_i^j \beta_j^i W_{i,j}) \ast x.
\]  (8)

To support the linear combination of convolutions, each kernel matrix must be the same size. However, different hyperparameter combinations lead to different sizes of kernel matrices. So transformations of each kernel matrix in both spatial and channel are required. In PDAS, we follow the transformation method of [38].

The process of spatial transformation can be seen from Fig. 4(a). Let \(h_1 \times w_1, h_2 \times w_2, \ldots, h_n \times w_n\) denote all candidate kernel sizes. Based on the fact that when the convolution with the size of \(h_1 \times w_1\) is appropriately zero-padded to the size of \(h_{\text{max}} \times w_{\text{max}}\), \(h_{\text{max}} = \max(h_1, h_2, \ldots, h_n)\) and \(w_{\text{max}} = \max(w_1, w_2, \ldots, w_n)\), the calculation process will not change. Thus, all kernel matrices are spatially padding to the size of \(h_{\text{max}} \times w_{\text{max}}\) while keeping the original operation unchanged.

The process of channel-wise transformation can be seen from Fig. 4(b). A simple trick is used to realize the channel-wise transformation by adjusting the number of input channels instead of changing the one of output channels. In this way, all kernel matrices are channel-wise padding to the size of \(c_{\text{max}} \times c_{\text{out}}\). In order to save computational resources, in this article, \(h_{\text{max}}\) and \(w_{\text{max}}\) are set to 5 for \(l = 1, 2, 3\) and 3 for \(l = 3\). \(h_{\text{max}} = w_{\text{max}} = 1\) for the fully connected layers. In addition, \(c_{\text{max}}\) is set to 64 for \(l = 1, 2, 3\), and 1024 for \(l = 4\). Similar ideas have emerged in some channel-wise pruning studies [45] whose connotation is to use the statistical information of \((i + 1)\)th layer to guide the architecture search of \(i\)th layer.

4) Two-Step One-Shot Architecture Search: With the preparation of the above two aspects, a two-stage one-shot search strategy can be introduced to reduce the resource requirements as well as keep the search precision. Through continuous relaxation, the efficient one-shot search strategy [38], [39], [46] becomes supported. At the same time, the addition of linear combination of convolution reduces the difficulty of calculation. However, it can be found that (8) contains \(\sum_{i,j} h_1 \times w_1 \times c_j \times c_{\text{out}}\) weight parameters. Such a large number of trainable parameters not only cause the demand for memory but also aggravate the difficulty of optimization. To solve this problem, the optimization of \(\alpha\) and \(\beta\) is carried out in an alternating iterative way to avoid the multiplication of weight parameters.

Fig. 3. Illustration of the proposed PolSAR-tailored search space for PDAS. Architecture search is carried out based on a five-layer CNN. The variables in the proposed search space are hyperparameters of each trainable layer, not the topology.

Fig. 4. Transformation of convolution kernels. The dark solid part represents the untransformed convolution kernel and the light dotted part represents zero-padding. (a) Process of spatial transformation. (b) Process of channel-wise transformation.
caused by jointly optimization. When $\beta$ is fixed, (8) is reduced to the following expression:

$$\tilde{\sigma}(x) = \sum_{i=1}^{n} (a_i^* b_i^* W_i) \ast x$$  \hspace{1cm} (9)

and when $\alpha$ is fixed, it can be expressed as

$$\tilde{\sigma}(x) = \sum_{j=1}^{m} (a_j^* b_j^* W_j) \ast x.$$  \hspace{1cm} (10)

Due to the intentionally diversified candidate architectures in DARTS, separate optimization of architecture and weight parameters are needed [46]. In contrast, the diversity of the proposed search space is relatively small, so it is acceptable to optimize the architecture and weight parameters on single training set. We introduce the optimization problem of PDAS as follows:

$$(\alpha^*, \beta^*) = \arg \min_{\alpha, \beta \in A} \{L_{\text{train}}(A, w) + \gamma R(A)\}$$  \hspace{1cm} (11)

which can be successively optimized by

$$\alpha^* = \arg \min_{\alpha} \{L_{\text{train}}(\alpha, w) + \gamma R(\alpha)\}$$  \hspace{1cm} (12)

and

$$\beta^* = \arg \min_{\beta} \{L_{\text{train}}(\beta, w) + \gamma R(\beta)\}$$  \hspace{1cm} (13)

where $L_{\text{train}}(\cdot)$ notes the cross-entropy loss on training set and $R(\cdot)$ is a regularization term weighted by $\gamma$. $L_1$ regularization is used in PDAS to add sparsity to architecture parameters. The iterative search process of PDAS for architecture parameters is outlined as Algorithm 1.

Once the optimal $\alpha^*, \beta^*$ are determined, the corresponding architecture can be constructed by keeping top-k strongest operations of each optimal architecture parameter ($k = 1$ in this article), and retrained to achieve PolSAR classification.

C. Complex-Valued PDAS

The proposed PDAS is extended to the complex domain for better performance complex-valued-PDAS (CV-PDAS). Instead of only supporting real-valued calculations, all real operations are converted to complex operations in CV-PDAS. In CV-CNNs [26]–[28], the weight of filters should be complex numbers so the weight $w \in \mathbb{C}$ and bias $b \in \mathbb{C}$ of a layer can be expressed as $w = \Re(w) + j \Im(w)$ and $b = \Re(b) + j \Im(b)$. Complex-valued convolution can be defined as follows:

$$x^{i+1} = (\Re(W^i) \ast \Re(x^i) - \Im(W^i) \ast \Im(x^i))$$

$$+ j (\Re(W^i) \ast \Im(x^i) + \Im(W^i) \ast \Re(x^i))$$

$$+ \Re(B^i) + j \Im(B^i)$$  \hspace{1cm} (14)

where $x^i$ and $x^{i+1}$ denote the output feature map of $(i-1)$th and $i$th layer, $W^i$ and $B^i$ are the weight matrix and bias vector of $i$th layer. Complex ReLU (CReLU) activation [47] is used in PDAS for activate the complex-valued products of convolutions. It can be defined as

$${\text{CReLU}}(z) = \max(0, \Re(z)) + j \max(0, \Im(z)).$$  \hspace{1cm} (15)

Algorithm 1 Two-Step One-Shot Architecture Search for PDAS

1: Begin;
2: Prepare Training and validation sets $D = \{x_i, y_i\}_{i=1}^{N}$, $x_i \in \mathbb{R}^{n \times n}$, positive integers $I_1, I_2$ and $B_1, B_2$, hyperparameter $\gamma$ and learning rate $\xi$.

Step One: Searching for $\alpha^*$
3: Prepare $\beta_{\text{fixed}}$. Initialize weight parameters $w, b$ and architecture parameters $\alpha$.
4: for epoch in $I_1$: do
5: for batch in $B_1$: do
6: Forward propagation by (9) with trainable parameters of $w, \alpha$;
7: Update $\alpha$ by gradient descent for (12);
8: end for
9: end for

Step Two: Searching for $\beta^*$;
10: Prepare $\alpha^*$. Initialize weight parameters $w, b$ and architecture parameters $\beta$.
11: for epoch in $I_2$: do
12: for batch in $B_2$: do
13: Forward propagation by (10) with trainable parameters of $w, \beta$;
14: Update $\beta$ gradient descent for (13);
15: end for
16: end for
17: return $\alpha^*, \beta^*$

Complex-valued fully connected layer can be seen as a special complex-valued convolution layer with $1 \times 1$ kernel size. A feature vector whose dimension is the number of target categories can be obtained after two times of fully connected layers, and its amplitude is used to make final predictions. Other parts of PDAS, including search space and search strategy, remain unchanged. Compared with real-valued PDAS, CV-PDAS can maintain the integrity of PolSAR data and mine the physical scattering mechanism hidden in the complex-valued covariance or coherency matrices.

III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, experiments used data sets and evaluation criteria are introduced first. Based on that, the performance of the proposed methods are demonstrated and analyzed under three widely used PolSAR benchmark data sets. The classification results of the whole map and testing accuracy under three different criteria are recorded and compared.

A. Data Sets Description

We employ three widely used PolSAR data sets in the experiments: AIRSAR San Francisco, AIRSAR Flevoland, and ESAR Oberpaffenhoen. Figs. 5–7 show their Pauli maps and ground truth maps, respectively. Besides, Tables I–III show some details about the three benchmark data sets.

1) AIRSAR San Francisco: This PolSAR image has been widely used in many studies. It is an L-band, full polarimetric...
image of San Francisco, as shown in Fig. 5. The size of this image is 900 × 1024 and the spatial resolution is 0.6 m × 1.6 m. Each pixel of this image can be classified into five categories, including mountain, ocean, urban, vegetation, and bare soil. The number of the labeled pixels can be seen in Table I.

2) AIRSAR Flevoland: As shown in Fig. 6, an L-band, full polarimetric image of the agricultural region of the Netherlands is obtained through the NASA/Jet Propulsion Laboratory AIRSAR. The size of this image is 750 × 1024, and the spatial resolution is 0.6 m × 1.6 m. There are 15 kinds of ground objects including buildings, stembeans, rapeseed, beet, bare soil, forest, potatoes, peas, lucerne, barley, grasses, water, wheat one, wheat two, and wheat three. The number of the labeled pixels can be seen in Table II.

3) ESAR Oberpfaffenhofen: An L-band, full polarimetric image of Oberpfaffenhofen, Germany, 1200 × 1300 scene size, are obtained through ESAR airborne platform. Its Pauli color-coded image and ground truth map can be seen in Fig. 7. Each pixel in the map is divided into three categories: built-up areas, wood land, and open areas, except for some unknown regions. The number of the labeled pixels can be seen in Table III.

### Table I

| AIRSAR San Francisco | | | |
| --- | --- | --- | --- |
| Class code | Name | Reference data |
| 1 | Mountain | 59035 |
| 2 | Ocean | 321583 |
| 3 | Urban | 334418 |
| 4 | Vegetation | 53509 |
| 5 | Bare soil | 13487 |
| Total | | 782032 |

### Table II

| AIRSAR Flevoland | | | |
| --- | --- | --- | --- |
| Class code | Name | Reference data |
| 1 | Buildings | 963 |
| 2 | Rapeseed | 17195 |
| 3 | Beet | 11516 |
| 4 | Stembeans | 6812 |
| 5 | Peas | 11394 |
| 6 | Forest | 20458 |
| 7 | Lucerne | 11411 |
| 8 | Potatoes | 19480 |
| 9 | Bare soil | 6116 |
| 10 | Grass | 8159 |
| 11 | Barley | 8046 |
| 12 | Water | 8824 |
| 13 | Wheat one | 16906 |
| 14 | Wheat two | 12728 |
| 15 | Wheat three | 24584 |
| Total | | 184592 |

### Table III

| ESAR Oberpfaffenhofen | | | |
| --- | --- | --- | --- |
| Class code | Name | Reference data |
| 1 | Built-up areas | 310829 |
| 2 | Wood land | 263238 |
| 3 | Open areas | 730705 |
| Total | | 1307142 |

### B. Experimental Setting

1) Data Representations: During the training and testing, PolSAR images are represented by the polarization coherence matrix $T$, which can be defined as

$$
\bar{k} = \frac{1}{\sqrt{2}} [S_{HH} + S_{VV}, S_{HH} - S_{VV}, 2S_{HV}]^T 
$$

$$
[T] = \langle \bar{k}\bar{k}^H \rangle 
$$

where $S_{PO}(P, Q \in \{H, V\})$ represents the backscattering coefficient of the polarized electromagnetic wave in emitting $Q$-direction and receiving $P$-direction. $H$ and $V$ represent the horizontal and vertical polarization, respectively. According to the reciprocity theorem, the $S$ matrix satisfies $S_{HV} = S_{VH}$. Notice that the polarization coherence matrix $T$ is a Hermitian
matrix, and every element except the diagonal element is a complex number. Generally, we take the upper triangular elements \(T_{11}, T_{12}, T_{13}, T_{22}, T_{23}, \text{ and } T_{33}\) as the input data. For real-valued models, the original data are divided into real parts and imaginary parts as the input; for complex-valued models, the original data are directly used as the input.

Benchmark PolSAR images are sliced into the size of \(15 \times 15\) image patches around the central pixel with the stride sliding windows of 1 according to the ground truth map, as can be seen from Tables I to III, to generate the training, validation, and testing data sets.

2) Comparing Methods and Parameter Settings: The effectiveness of the proposed methods are proved by comparisons with some effective alternatives. The architectures searched by the proposed methods, i.e., PDAS and CV-PDAS, are noted as PDAS-CNN and CVPDAS-CNN for convenience. Three nondeep learning methods are chosen, including Wishart [48], support vector machines (SVMs) [49], and SVM, with radial basis function kernel (RBF-SVM). Three deep learning methods are chosen, including CNN [18], CV-CNN [26], and polarimetric-feature-driven CNN (SF-CNN) [23]. Moreover, a related NAS method for the search of hyperparameters [38] is compared to show the validity of tailored search space and improved search strategy.

Considering that the network structure obtained through the search is highly transferable [34], we do architecture search on a small data set and then evaluate the obtained architectures in different benchmarks. In this article, the process of architecture search is carried out on Flevoland data, with 300 training samples of each category and 200 training epoch. The complete search process is repeated ten times to reduce the randomness of the searched architectures.

After the architecture search, the model retraining is needed to achieve the classification. The number of training epoch is an important hyperparameter for the training of all involved methods, which determines whether weights of the network converge or not. In order to obtain a suitable value of epoch on every data set, experiments are carried out and the results are shown in Figs. 8–10.

From the experimental results, one can see that the error rate of real-valued CNN and PDAS-CNN on training and validation sets tend to be stable within 500 iterations. Therefore, the value of training epoch is set to 500 in all experiments and a small validation set is used to select the best model from many iterations. Some other setting as follows: After each convolution layer, a ReLU activation layer is added. In order to speed up the optimization of objective function and obtain better approximate solution, SGD with adaptive moment estimation [50] are used with the learning rate of \(10^{-4}\). Number of mini-batch is set to 64 during the training. Deep learning toolbox [51] is utilized to minimize the difficulty of algorithm implementation.

3) Evaluation Criteria: To evaluating the performance of the algorithms mentioned in this article, OA, average accuracy (AA), and kappa coefficient (Kappa) are chosen as criteria, which can be defined as follows:

\[
OA = \frac{\sum_{i=1}^{c} M_i}{\sum_{i=1}^{c} N_i}, \quad AA = \frac{1}{c} \sum_{i=1}^{c} \frac{M_i}{N_i}
\]

(18)

where \(c\) is the number of categories. \(M_i, N_i\) denotes the number of \(i\)th category correctly classified samples and the number of the \(i\)th category labeled samples, respectively

\[
Kappa = \frac{OA - P}{1 - P}, \quad \text{with} \quad P = \frac{1}{N^2} \sum_{i=1}^{c} H(i, :)H(:, i)
\]

(19)

where \(N\) is the number of testing samples and \(H\) denotes the classification confusion matrix.

C. Experimental Results

The experimental results of the three PolSAR benchmarks are shown in Tables IV–VI. The whole map classification results are shown in Figs. 11–13. Generally speaking, deep
Learning-based methods show more powerful classification ability than rest of the methods. Moreover, two improved versions, i.e., CV-CNN and SF-CNN, have better performance than the ordinary CNN. Although no attention has been paid to the network design, architectures searched by the proposed PDAS and CV-PDAS emerge the best generalization performance in three benchmark data sets. It is worth noting that the proposed methods may not improve the classification accuracy of each category. The inherent reason is that the purpose of architecture search is to maximize OA in this article, rather than to optimize a specific category. This approach is reasonable to look for a good architecture for multiclass PolSAR terrain classification. Of course, using the accuracy of a certain category as the search criterion is also supported in order to achieve some other tasks, e.g., target detection. In summary, strictly following the proposed methods may not be suitable for more accurate detection of a particular category. However, the proposal is quite flexible, and the desired architectures

| Method     | Wishart | SVM   | RBF-SVM | CNN  | CV-CNN | SF-CNN | PDAS-CNN | CVPDAS-CNN |
|------------|---------|-------|---------|------|--------|--------|----------|------------|
| Mountain   | 56.25   | 64.45 | 67.40   | 90.99| 88.19  | 87.50  | 92.43    | 95.12      |
| Ocean      | 91.67   | 98.19 | 97.97   | 93.24| 95.61  | 96.79  | 95.62    | 94.66      |
| Urban      | 76.24   | 66.49 | 82.00   | 88.57| 90.05  | 89.34  | 92.77    | 95.28      |
| Vegetation | 80.59   | 68.34 | 64.80   | 82.47| 83.83  | 83.49  | 85.38    | 85.68      |
| Bare soil  | 86.69   | 83.09 | 62.59   | 95.70| 93.40  | 97.11  | 93.55    | 92.73      |
| OA         | 81.49   | 75.89 | 81.34   | 90.38| 91.83  | 92.01  | 93.43    | 94.32      |
| AA         | 78.29   | 76.11 | 74.95   | 90.19| 90.22  | 90.85  | 91.95    | 92.69      |
| Kappa      | 72.87   | 67.63 | 74.03   | 85.47| 87.54  | 87.82  | 89.92    | 91.35      |

| Method     | Wishart | SVM   | RBF-SVM | CNN  | CV-CNN | SF-CNN | PDAS-CNN | CVPDAS-CNN |
|------------|---------|-------|---------|------|--------|--------|----------|------------|
| Buildings  | 68.02   | 88.47 | 87.54   | 100.0| 99.58  | 100.0  | 100.0    | 100.0      |
| Rapeseed   | 50.16   | 81.38 | 85.04   | 80.77| 92.21  | 93.79  | 89.82    | 96.97      |
| Beet       | 91.86   | 90.11 | 90.91   | 94.54| 95.33  | 98.82  | 95.15    | 96.18      |
| Stembeans  | 93.04   | 85.26 | 88.21   | 98.96| 98.22  | 99.47  | 99.90    | 99.94      |
| Peas       | 87.06   | 94.30 | 94.38   | 98.21| 98.44  | 97.54  | 98.10    | 98.97      |
| Forest     | 72.94   | 82.22 | 85.19   | 97.85| 99.45  | 96.96  | 98.39    | 98.87      |
| Lucerne    | 90.82   | 87.91 | 89.10   | 93.82| 97.68  | 98.04  | 96.29    | 96.34      |
| Potatoes   | 58.83   | 74.14 | 81.50   | 95.30| 87.33  | 94.36  | 95.00    | 98.61      |
| Bare soil  | 94.46   | 95.14 | 94.72   | 90.79| 83.65  | 99.13  | 99.33    | 98.14      |
| Grass      | 65.36   | 70.52 | 66.01   | 82.14| 91.85  | 92.82  | 95.21    | 97.24      |
| Barley     | 94.24   | 95.45 | 94.95   | 96.91| 99.58  | 97.92  | 96.96    | 99.95      |
| Water      | 52.95   | 97.08 | 97.67   | 99.77| 99.99  | 99.90  | 100.00   | 100.00     |
| Wheat one  | 89.76   | 70.16 | 79.11   | 96.43| 96.84  | 96.00  | 98.38    | 98.92      |
| Wheat two  | 84.41   | 82.79 | 78.57   | 90.26| 96.67  | 92.51  | 94.72    | 98.62      |
| Wheat three| 82.44   | 78.66 | 86.73   | 98.96| 98.96  | 95.29  | 98.69    | 98.49      |
| OA         | 77.15   | 82.69 | 85.77   | 94.23| 95.70  | 96.17  | 96.64    | 98.32      |
| AA         | 78.42   | 84.91 | 86.64   | 94.31| 95.72  | 96.84  | 97.06    | 98.45      |
| Kappa      | 75.15   | 81.18 | 84.50   | 93.70| 95.31  | 95.83  | 96.34    | 98.16      |

| Method     | Wishart | SVM   | RBF-SVM | CNN  | CV-CNN | SF-CNN | PDAS-CNN | CVPDAS-CNN |
|------------|---------|-------|---------|------|--------|--------|----------|------------|
| Built-up areas | 45.19   | 55.35 | 47.20   | 77.85| 77.02  | 78.09  | 81.55    | 80.48      |
| Wood land  | 74.87   | 79.38 | 90.07   | 91.44| 90.75  | 88.43  | 91.03    | 93.55      |
| Open areas | 94.49   | 89.48 | 91.35   | 93.10| 94.23  | 93.87  | 94.51    | 94.38      |
| OA         | 78.80   | 77.83 | 79.15   | 89.14| 89.44  | 89.02  | 90.72    | 90.91      |
| AA         | 71.52   | 74.07 | 76.21   | 87.46| 87.33  | 86.80  | 89.03    | 89.47      |
| Kappa      | 63.39   | 65.00 | 66.98   | 81.59| 82.00  | 81.33  | 84.17    | 84.55      |
can be obtained by controlling the evaluation criterion of the search process. The experimental results on each data set are analyzed as follows.

Classification results of the whole map on San Francisco data can be seen from Fig. 11. Architectures obtained by the proposed search methods, especially CV-PDAS-CNN, achieve higher completeness of the terrains in the classification maps. It can be seen from the results that the proposed methods have better recognition accuracy than the comparing methods for the identification of urban and vegetation categories. The proposed methods also avoid the situation that urban is wrongly divided into mountain category in SF-CNN. This observation proves that the proposed methods can obtain better CNN architectures to achieve good classification results on the San Francisco data set. The experimental results on the testing set are shown in Table IV, in which the proposed methods achieve the best and the second results on three criteria; 3.05% and 4.45% increase of OA and Kappa are accomplished through PDAS for real-valued CNN and 2.49%, 3.71% for complex-valued CNN. This can verify the correctness of the theoretical analysis from the experimental point of view.
With the addition of PDAS, the performance of CNN has been improved in all categories except for the category bare soil. This shows that PDAS has good adaptability in the real-valued topology in experiments. After the search of CV-PDAS, the architecture has a little degeneration in ocean and bare soil categories, and improved in other categories, especially in the recognition of mountain category.

The whole map classification results on Flevoland data are presented in Fig. 12, from which one can see that the classification results of the searched architectures are clearer and more close to the ground truth map. Table V shows the experimental results of each algorithm on the Flevoland data set. The proposed two methods achieve the best and the second performance. With the help of PDAS and CV-PDAS, the recognition accuracy of real-valued and complex-valued architectures in almost every category has been improved, especially for the categories of rapeseed, bare soil, and grass. From the global point of view, the architecture searched by CV-PDAS gets 98.32% OA, which is much higher than the results of other comparing methods. The process of search improves 2.41% OA and 2.64% Kappa for real-valued CNN, 2.62% OA and 2.85% Kappa for CV-CNN. This further proves that the proposed two search methods can improve the performance of a specific architecture in a plug-and-play manner.

Fig. 13 reports the whole map classification results on the Oberpfaffenhofen data set. It can be seen from the results of involved methods on testing set, from which one can see that the addition of the proposed search methods has brought certain improvements in the three evaluation criteria of the comparing methods. PDAS is capable of getting accuracy increments of 1.58%, 1.57%, and 2.58% for OA, AA, and Kappa coefficient, respectively. For CV-PDAS, the value of increments changes to 1.47%, 2.14%, and 2.55%. These results indicate that the automatically optimized architectures obtained by the proposed search methods do have superior performance compared to the hand-crafted ones.

Comparisons of the five CNN-based methods are shown in Fig. 14, and it is shown that the proposed two methods have better performance on each benchmark data set for the three evaluation criteria. It is worth noting that the architectures of PDAS-CNN and CV-PDAS-CNN used in three groups of experiments are obtained by searching on the Flevoland data set. It can be seen that transferring the searched architectures to other data sets still shows quite good performance, which proves that the proposed search methods have good robustness [34].

A practical hyperparameter search method was proposed in [38], called DAS, which can be regarded as our basis and some customized improvements are implemented. First, a tailored search space is proposed to adapt the NAS method to the task of PolSAR classification. In addition, \( n \times m \) groups of weight parameters were merged into one with the size of \( h_{\text{max}} \times w_{\text{max}} \times c_{\text{max}} \times c_{\text{out}} \) in [38]. Although the number of parameters is reduced by parameter merging, untractable bias is introduced because all the branches share the parameter from the biggest kernel matrix. Instead of roughly relaxing \( n \times m \) groups parameters to 1, in PDAS, a two-stage

![Fig. 13. Classification results of whole map on ESAR Oberpfaffenhofen data with different methods. Results of (a) Wishart, (b) SVM, (c) RBF-SVM, (d) CNN, (e) CV-CNN, (f) SF-CNN, (g) PDAS-CNN, and (h) CVPDAS-CNN.](image-url)
optimization for architecture parameters is used to relax $n \times m$ to $n + m$. This can be seen as an improvement in the search strategy for higher precision. In order to verify the correctness of the above analysis, the two methods are tested on three PolSAR benchmark data sets. The experimental results are shown in Table VII.

It can be seen from the results that the performance of standard CNN is improved by architecture search. And the proposed PDAS shows better performance by all the criteria on the three data sets compared to the baseline, which is closely related to the changes in the search space and search strategy.

D. Discussion

The above experimental results show that the architectures obtained by the proposed PDAS and CV-PDAS achieved the state of the art performance. A global analysis and discussion are given after experimental simulations.

First, thanks to the continuous relaxed search space, the proposed methods can be optimized by SGD to do a one-shot architecture search, which brings about a very fast search speed. In our experimental environment, it takes only a few hours to implement a one-shot architecture search. This is beneficial for others to repeat and follow-up our work. Not only this, the architectures obtained by searching have a strong diversity compared with the current widely used ones in PolSAR area. Convolution kernels with $3 \times 3$ spatial size have been the choice of most existing studies, because its performance in optical image processing tasks is quite good. However, in our experiments, the size of $3 \times 3$ cannot show an overwhelming advantage. $2 \times 1, 5 \times 5, 1 \times 1$ convolutions are used in PDAS-CNN and $3 \times 1, 1 \times 1, 1 \times 1$ ones are used in CVPDAS-CNN. This result also shows the difference between PolSAR image classification and optical image classification. We suspect that the reason for this phenomenon may be that image slices require a smaller convolution kernel to keep more details. The selected number of nodes of fully connected layers also show good diversity, but the trend of change is roughly predictable, that is, performance increases as the number of nodes are increased. Furthermore, the improved optimization method of PDAS avoids the bilevel optimization problem that classical gradient-based NAS methods need to face. In our test, this improvement indeed improves the stability of the search process. Last but not least, the proposed search method is highly flexible and can be extended to any network architectures in any task to optimize their hyperparameters. Therefore, the proposed PDAS and CV-PDAS are not just searching suitable architectures for PolSAR image classification from scratch, they also can be used as plug-and-play complementary components to improve the performance of existing methods.

IV. Conclusion

In this article, PolSAR image classification is promoted with the help of NAS for the first time. On the basis of the gradient-based NAS, a tailored DAS method for PolSAR classification is proposed, which fully considers the characteristics of PolSAR classification task. In the proposed PDAS, the search space is customized to adapt NAS to the PolSAR classification, and the search strategy is improved so as to seek suitable CNN architectures efficiently. Compared with the ordinary paradigm of CNN-based PolSAR classification, the proposed method is quite cost-effective. Furthermore, the complex-valued version of PDAS is introduced to fit the data format of PolSAR images.

Experiments on three widely used PolSAR benchmark data sets show that the architectures obtained by the proposed search methods have certain advantages over hand-crafted ones, which is reflected in the higher accuracy and more close to the ground truth map in the whole map classification. The main reason why the proposed methods have more powerful

---

TABLE VII

| Dataset         | Criterion | DAS    | PDAS    |
|-----------------|-----------|--------|---------|
| San Francisco   | OA        | 91.78  | 93.43   |
|                 | AA        | 89.06  | 91.95   |
|                 | Kappa     | 87.44  | 89.92   |
| Flevoland       | OA        | 95.14  | 96.64   |
|                 | AA        | 95.74  | 97.06   |
|                 | Kappa     | 94.70  | 96.34   |
| Oberpfaffenhofen| OA        | 90.08  | 90.72   |
|                 | AA        | 88.44  | 89.03   |
|                 | Kappa     | 83.17  | 84.17   |
performance is that most of the PolSAR classification used CNN architectures are only paths in the search space of PDAS or CV-PDAS. The proposal of PDAS and CV-PDAS indicates the future research direction of deep learning techniques in PolSAR image classification. For our future works, better input forms of PolSAR data, high-precision NAS methods, and their application to PolSAR semantic segmentation are all the issues we are considering.

REFERENCES

[1] W. L. Cameron and L. K. Leung, “Feature motivated polarization scattering matrix decomposition,” in Proc. IEEE Int. Conf. Radar, May 1990, pp. 549–557.

[2] Y. Yamaguchi, A. Sato, W.-M. Boerner, R. Sato, and H. Yamada, “Four-component scattering power decomposition with rotation of coherency matrix,” IEEE Trans. Geosci. Remote Sens., vol. 49, no. 6, pp. 2251–2258, Jun. 2011.

[3] L. Zhang, B. Zou, H. Cai, and Y. Zhang, “Multiple-component scattering model for polarimetric SAR image decomposition,” IEEE Trans. Geosci. Remote Sens. Lett., vol. 5, no. 4, pp. 603–607, Oct. 2008.

[4] 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.

[5] M. Turk and A. Pentland, “Eigenfaces for recognition,” J. Cognit. Neurosci., vol. 3, no. 1, pp. 71–86, Jan. 1991.

[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 5, pp. 436–444, May 2015.

[7] Y. LeCun et al., “Backpropagation applied to handwritten zip code recognition,” Neural Comput., vol. 1, no. 4, pp. 541–551, Dec. 1989.

[8] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Lake Tahoe, CA, USA, Dec. 2012, pp. 1097–1105.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[10] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, “When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 5, pp. 2811–2821, May 2018.

[11] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, “Exploring hierarchical convolutional features for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[12] P. Zhou, J. Han, G. Cheng, and B. Zhang, “Learning compact and discriminative stacked autoencoder for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[13] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, “High-resolution SAR image classification via deep convolutional autoencoders,” IEEE Trans. Geosci. Remote Sens. Lett., vol. 12, no. 11, pp. 2351–2355, Nov. 2015.

[14] 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, Aug. 2016.

[15] Z. Lin, K. Ji, M. Kang, X. Leng, and H. Zou, “Deep convolutional highway unit network for SAR target classification with limited labeled training data,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 7, pp. 991–995, Jul. 2017.

[16] J. Pei, Y. Huang, W. Huo, Y. Zhang, J. Yang, and T.-S. Yeo, “SAR automatic target recognition based on multi-view deep learning framework,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 4, pp. 2196–2210, Apr. 2018.

[17] R. Shang, J. Wang, L. Yao, R. Stolkin, B. Hou, and Y. Li, “SAR targets classification based on deep memory convolution neural networks and transfer parameters,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 8, pp. 2834–2846, Aug. 2018.

[18] 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.

[19] W. Wu, H. Li, L. Zhang, X. Li, and H. Guo, “High-resolution PolSAR scene classification with pretrained deep convnets and manifold polarimetric parameters,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 10, pp. 6159–6168, Oct. 2018.
[44] A. Martins and R. Astudillo, “From softmax to sparsemax: A sparse model of attention and multi-label classification,” in Proc. 33rd Int. Conf. Mach. Learn. (ICML), 2016, pp. 1614–1623.

[45] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, “ThiNet: Pruning CNN filters for a thinner net,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 10, pp. 2525–2538, Oct. 2019.

[46] S. Singh, A. Khetan, and Z. Karnin, “DARC: Differentiable ARchitecture compression,” May 2019, arXiv:1905.08170. [Online]. Available: http://arxiv.org/abs/1905.08170

[47] C. Trabelsi et al., “Deep complex networks,” (Feb. 2018), arXiv:1705.09792. [Online]. Available: https://arxiv.org/abs/1705.09792

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

[49] C. Lardeux et al., “Support vector machine for multifrequency SAR polarimetric data classification,” IEEE Trans. Geosci. Remote Sens., vol. 47, no. 12, pp. 4143–4152, Dec. 2009.

[50] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Dec. 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[51] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” Mar. 2016, arXiv:1603.04467. [Online]. Available: https://arxiv.org/abs/1603.04467

Hongwei Dong received the M.S. degree from the College of Science, China Agricultural University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree with Harbin Institute of Technology, Harbin, China. His research interests include applied mathematics, optimization methods, and machine learning.

Bin Zou (Senior Member, IEEE) received the B.S. degree in electronic engineering from the Harbin Institute of Technology, Harbin, China, in 1990, the M.Sc. degree in space studies from International Space University, Strasbourg, France, in 1998, and the Ph.D. degree in information and communication engineering from Harbin Institute of Technology, in 2001.

From 1990 to 2000, he was with the Department of Space Electro-Optic Engineering, Harbin Institute of Technology. From 2003 to 2004, he was a Visiting Scholar with the Department of Geological Sciences, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor and the Vice Head with the Department of Information Engineering, Harbin Institute of Technology. His research interests include SAR image processing, polarimetric SAR, and polarimetric SAR interferometry.

Lamei Zhang (Senior Member, IEEE) received the B.S., M.Sc., and Ph.D. degrees in information and communication engineering from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively.

She was a Visiting Scholar with the Department of Geological Sciences, University of Manitoba, Winnipeg, MB, Canada, from 2014 to 2015. She is currently an Associate Professor with the Department of Information Engineering, Harbin Institute of Technology. She serves as the Secretary of the IEEE Harbin Geoscience and Remote Sensing (GRSS) Chapter. Her research interests include remote sensing images processing, information extraction, and intelligent interpretation of high-resolution SAR, polarimetric SAR, and polarimetric SAR interferometry.

Siyu Zhang received the B.S. degree in electronic and information engineering from Dalian Maritime University, Dalian, China, in 2019. He is currently pursuing the M.S. degree in information and communication engineering with Harbin Institute of Technology, Harbin, China. His research interest includes PolSAR image interpretation and machine learning.