remote sensing

MDPI

Article

PolSAR Image Classification with Lightweight 3D Convolutional Networks

Hongwei Dong, Lamei Zhang and Bin Zou *

Department of Information Engineering, Harbin Institute of Technology, Harbin 150001, China; 18b905036@stu.hit.edu.cn (H.D.); lmzhang@hit.edu.cn (L.Z.)

* Correspondence: zoubin@hit.edu.cn

Received: 22 December 2019; Accepted: 21 January 2020; Published: 26 January 2020

Abstract: Convolutional neural networks (CNNs) have become the state-of-the-art in optical image processing. Recently, CNNs have been used in polarimetric synthetic aperture radar (PolSAR) image classification and obtained promising results. Unlike optical images, the unique phase information of PolSAR data expresses the structure information of objects. This special data representation makes 3D convolution which explicitly modeling the relationship between polarimetric channels perform better in the task of PolSAR image classification. However, the development of deep 3D-CNNs will cause a huge number of model parameters and expensive computational costs, which not only leads to the decrease of the interpretation speed during testing, but also greatly increases the risk of over-fitting. To alleviate this problem, a lightweight 3D-CNN framework that compresses 3D-CNNs from two aspects is proposed in this paper. Lightweight convolution operations, i.e., pseudo-3D and 3D-depthwise separable convolutions, are considered as low-latency replacements for vanilla 3D convolution. Further, fully connected layers are replaced by global average pooling to reduce the number of model parameters so as to save the memory. Under the specific classification task, the proposed methods can reduce up to 69.83% of the model parameters in convolution layers of the 3D-CNN as well as almost all the model parameters in fully connected layers, which ensures the fast PolSAR interpretation. Experiments on three PolSAR benchmark datasets, i.e., AIRSAR Flevoland, ESAR Oberpfaffenhofen, EMISAR Foulum, show that the proposed lightweight architectures can not only maintain but also slightly improve the accuracy under various criteria.

Keywords: deep learning; polarimetric synthetic aperture radar (PolSAR) classification; 3D convolution; pseudo-3D convolution; depthwise separable convolution

1. Introduction

Polarimetric synthetic aperture radar (PolSAR), as one of the most advanced detectors in the field of remote sensing, can provide rich target information in all-weather and all-time. In recent years, more and more attention has been paid to the development of PolSAR information extractions due to the good properties of PolSAR systems. Especially, PolSAR image classification has been extensively studied as the basis of PolSAR image interpretation.

Deep learning [1] has made remarkable progress in natural language processing and computer vision, and it has the potential to be applied in many other fields. Convolutional neural networks (CNNs), as one of the representative methods of deep learning, have shown strong abilities in the task of image processing [2]. It has been proved that CNNs can obtain more abstract feature representations than traditional hand-engineered filters. The generalization performance of machine learning-based image classification algorithms has been greatly improved with the rise of CNNs. Big data, advanced algorithms, and improvements in computing power are the key factors for the

success of CNNs. These factors also exist in PolSAR image classification. Therefore, it is promising to use CNNs to improve PolSAR image classification.

Before the popularity of deep learning, machine learning algorithms have been applied in PolSAR image classification for a long time. Statistical machine learning methods represented by support vector machines have been utilized to implement PolSAR image feature classification [3]. Considering the significant achievements made by CNNs, many studies have applied CNNs to the task of SAR or PolSAR image classification and achieved remarkable results [4]. Ding et al. introduced a four-layers CNN architecture [5] to do SAR image target recognition for the first time. A more carefully-designed network architecture was proposed to further explore deep features [6]. The impact of target angles was taken into consideration, and a multi-view metric based CNN was proposed to achieve high precision classification on MSTAR dataset [7]. Ren et al. introduced a patch-sorted based architecture for high-resolution SAR image classification [8]. Some complex tasks were implemented on the basis of deep features extracted by CNNs, such as change detection [9] and road segmentation [10]. In contrast, the application level of CNNs in PolSAR image classification is lower, but it is in the stage of rapid development. After some attempts on stacking shallow models [11], Zhou et al. applied CNN to PolSAR image classification for the first time [12]. In their work, a three-layers architecture was introduced to classify PolSAR images and obtained promising classification results. After that, many CNN architectures were introduced such as graph-based architecture [13], fully convolutional networks [14] and some advanced network backbones [15,16]. However, due to different imaging mechanisms, directly following the architectures of optical image classification may not fully utilize the capabilities of CNNs in PolSAR image classification. In other words, CNNs still have the potential to be explored in the task of PolSAR image classification.

As mentioned above, designing suitable CNN architectures for PolSAR image classification is necessary to pursue more powerful performances. Related studies are being carried out and they can be roughly divided into two parts according to their focus, i.e., task characteristics and data form. Lack of supervision information is a representative task characteristic of PolSAR image classification. Although the acquisition of PolSAR images is not difficult, they do not have labels. In other words, most of the acquired PolSAR images cannot be directly used by the existing mainstream CNNs. Moreover, it is more difficult to label them manually compared with optical images. To handle this problem, weakly supervised methods, such as automatic pseudo-labels, transfer learning, and regularization techniques, were introduced to achieve PolSAR image small sample classification. A super-pixel restrained network was designed to do semi-supervised PolSAR classification with the aid of a pseudo-labels strategy [17]. Similarly, active learning was used to do pseudo labels and deep learning-based semi-supervised PolSAR classification was achieved in [18]. Wu et al. implemented transfer learning on a modified U-Net [19] to realize PolSAR small sample pixel-wise classification. Bi et al. added a graph-based regularization term to the ordinary CNN and achieved semi-supervised PolSAR classification [13]. In addition to improving the CNN architectures according to the characteristics of PolSAR classification tasks, making the architectures adapt to the complex-valued PolSAR data has also been extensively considered. Different from optical sensors, PolSAR can obtain the phase information between target and radar because of its unique scattering imaging mechanism. Therefore, the architectures which can make full use of the information contained in PolSAR data are of great significance to the development of CNNs in PolSAR image classification. This is also the objective of this work. Chen et al. tried to use the hand-engineered features as the input of CNNs to make better use of PolSAR data without changing the network architecture [20]. Different from changing the inputs, An intuitive improvement to better utilize complex-valued PolSAR data was extending the real-valued architectures to the complex domain [21,22]. Zhang et al. elaborated on the previous studies in detail and designed a three-layers complex-valued CNN to adapt to the characteristics of PolSAR data and implement PolSAR image classification [23]. At present, complex-valued architectures have been followed by many studies. Shang et al. introduced a complex-valued convolutional autoencoder network for

PolSAR classification [24]. Complex-valued fully convolutional networks were proposed in [25] to do PolSAR semantic segmentation. Sun et al. proposed a complex-valued generative adversarial network for semi-supervised PolSAR classification [26]. However, the development of complex-valued architectures is still in its infancy. To avoid complex operations, Liu et al. attempted to learn the feature of phase independently [27]. A two-stream architecture was proposed to extract features from amplitude and phase respectively with the aid of a multi-task feature fusion mechanism [28]. It is worth noting that PolSAR covariance matrix has been used as the input of CNNs in most studies [12,15,16,23,28], and the phase information is hidden between the input channels when each element of the upper triangle of PolSAR covariance matrix is regarded as a channel of the input. Recent works have revealed that, with the aid of 3D operations, channel-wise correlations can be plugged in as an additional dimension of convolution kernels to solve the problem of feature mining on special data (e.g. videos) [29,30]. Such improvements induce considerable advantages when processing PolSAR data by CNNs. Zhang et al. introduced 3D operations for the first time to implement PolSAR classification [31], which effectively improved the performance of ordinary 2D-CNNs. Tan et al. integrated complex-valued and 3D operations, and proposed a complex-valued 3D-CNN for PolSAR classification [32]. However, the performance improvement brought by 3D convolutions is based on greatly increased model parameters [30]. A large number of model parameters limit the speed of classification, which hinders the practical implementations of 3D-CNNs and the development of real-time interpretation systems [33]. Lightweight alternatives of 3D convolutions, e.g. pseudo-3D convolution [34] and depthwise separable convolution [35,36] are good means to solve this dilemma.

Based on the above analysis, the objective of this work is to find 3D-CNNs architectures with low computational cost as well as competitive performance for PolSAR image classification. It can be observed that almost all model parameters of a CNN exist in the convolution and fully connected layers. For these two key components, lightweight strategies are developed in this paper to compress the network architecture so as to reduce the model complexity of 3D-CNNs. Firstly, pseudo-3D convolution-based CNN (P3D-CNN) is introduced which replaces the convolution operations of 3D-CNNs by pseudo-3D convolutions. P3D-CNN uses two successive 2D operations to approximate the features extracted by 3D-CNNs. In addition, 3D-depthwise separable convolution-based CNN (3DDW-CNN) is proposed in parallel. Different from P3D-CNN, 3DDW-CNN decouples the spatial-wise and channel-wise operations that were previously mixed together to find more effective features than 3D-CNNs. The number of model parameters contained in convolution layers can be greatly reduced in the proposed two lightweight architectures. Moreover, fully connected layers of the above two architectures are eliminated and replaced by global average pooling layers [37]. This measure reduces more than 90% of the model parameters in 3D-CNNs and greatly improves the computational efficiency. The dropout mechanism [38] is configured in the proposed architectures to further prevent over-fitting. The proposed architectures can be summarized as a lightweight 3D-CNN framework, which has more efficient convolution and fully connected operations. The proposal has inspirations for the development of many other lightweight architectures. The number of trainable parameters and the computational complexity of the involved models are compared and analyzed, which illustrates the superiority of the lightweight architectures. The classification performance of the proposed methods is tested on three PolSAR benchmark datasets. Experimental results show that considerable accuracy can be maintained by the proposed methods. The main contribution of this paper can be summarized as follows:

- Two lightweight 3D-CNN architectures are introduced for the fast PolSAR interpretation speed during testing.
- Two lightweight 3D convolution operations, i.e., pseudo-3D and 3D-depthwise separable convolutions, and global average pooling are applied to reduce the redundancy of 3D-CNNs.
- A lightweight 3D-CNN framework can be summarized. Compared with ordinary 3D-CNNs, the architectures under the framework have fewer model parameters and lower computational complexity.

- The performance of the lightweight architectures is verified on three PolSAR benchmark datasets.

The rest of this paper is organized as follows. In Section 2, the background of vanilla convolutions and their variants are introduced. The proposed methods are introduced in Section 3. The experimental results and analysis are presented in Section 4. The conclusion is discussed in Section 5.

2. Related Works

In this section, 2D convolution, 3D convolution and its lightweight versions, i.e., pseudo-3D convolution and 3D-depthwise separable convolution, are briefly analyzed. Formula expressions are avoided and graphical illustrations are used to facilitate understanding.

2.1. Vanilla Convolutions

2D convolution is the choice of most CNNs, which can be used to extract the information from the input maps. The process of vanilla 2D convolution operation is shown in Figure 1, from which one can see that the output of a 2D convolution is always two-dimensional, i.e., one feature map, for any size of inputs. Therefore, 2D convolution can only extract spatial information, and it is not conducive to process the data which has a relationship between channels by 2D convolutions.

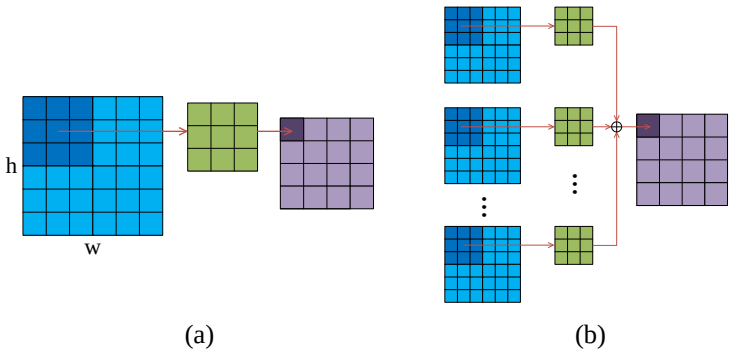


Figure 1. Illustrations of vanilla 2D convolution. (**a**) When the input is a single $h \times w$ map, each kernel is $k \times k$ and the corresponding output is a 2D $(h-k+1) \times (w-k+1)$ map. (**b**) When the input is c numbers $h \times w$ maps, each kernel is $k \times k \times c$. Doing the same operation on each channel as in (a), getting c 2D maps and add them up. The outputs of two sub-graphs are 2D maps with the same size.

Vanilla 3D convolution (C3D) can be seen as an intuitive extension of 2D convolutions and a dimension is added to extract more information [30]. As shown in Figure 1, the process of vanilla 2D convolution can be expressed as

$$z^{(t,h)} = \sum_{i=1}^{k_h k_w} x_i^{(t)} y_i^{(h)} + b^{(h)}, \tag{1}$$

where t and h mean the tth sliding window and the hth convolution kernel, k_h and k_w represent the spatial kernel size, $x^{(t)}$ and $z^{(t)}$ denote the tth input and the tth output, and $y^{(h)}$ and $b^{(h)}$ denote the hth kernel matrix and its bias. Similarly, C3D can be expressed as

$$z^{(t,h)} = \sum_{j=1}^{k_d} \sum_{i=1}^{k_h k_w} x_{i,j}^{(t)} y_{i,j}^{(h)} + b^{(h)}, \tag{2}$$

where k_d represents the depth of kernels. The process of C3D can be seen from Figure 2, where the extra depth dimension is added to the 2D convolution kernels. The difference between 2D and 3D convolutions can be seen by comparing Figure 1b with Figure 2a. Similar to 2D convolutions to maintain the spatial size of the inputs, the size of the depth dimension is maintained through 3D convolutions. In other words, the input is only manipulated spatially for 2D convolutions and the

output is always maps. However, C3D extract features from spatial and depth dimensions at the same time, and outputs cubes. The latter undoubtedly contains more information as well as more model parameters to be trained.

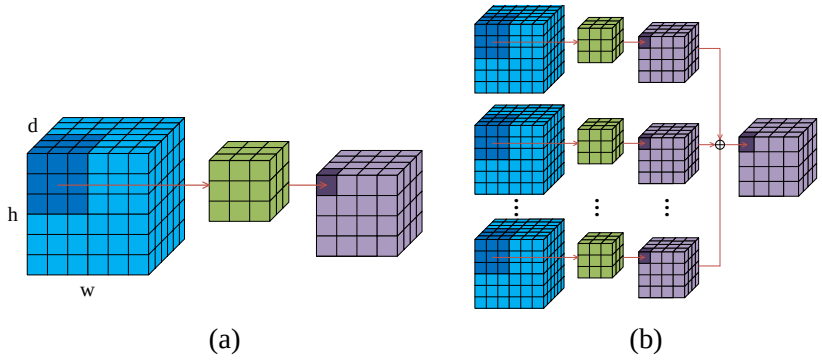


Figure 2. Illustrations of vanilla 3D convolution (C3D). C3D is an intuitive extension of 2D convolution. (**a**) When the input is a single $h \times w \times d$ cube, each kernel is $k \times k \times k$ and the corresponding output is a 3D $(h-k+1) \times (w-k+1) \times (d-k+1)$ cube. (**b**) When the input is c numbers $h \times w \times d$ cubes, each kernel is $k \times k \times k \times c$. Same as the operations in (a), c numbers 3D cubes can be obtained and add them up. The outputs of two sub-graphs are 3D cubes with the same size.

2.2. Pseudo-3D Convolution

The process of pseudo-3D convolution (P3D) can be seen in Figure 3. Two successive sub-operations work on spatial dimension and depth dimensions respectively are used by P3D to simulate the effect of C3D. It has been proven that P3D can greatly reduce the number of trainable parameters while keeping accuracy [34].

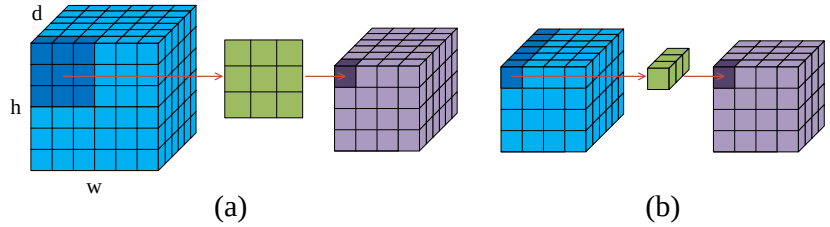


Figure 3. The process of pseudo-3D convolution (P3D). P3D is divided into two steps to achieve low-latency approximation to C3D, and a nonlinear activation exists between the two. (**a**) Step 1: Operating 2D convolution in the spatial dimension of the $h \times w \times d$ input, each kernel is $k \times k \times 1$ and the corresponding output is a $(h-k+1) \times (w-k+1) \times d$ cube. (**b**) Step 2: Operating 1D convolution in the depth dimension, each kernel is $1 \times 1 \times k$. Getting the final output with the size of $(h-k+1) \times (w-k+1) \times (d-k+1)$.

As shown in Figure 3, P3D decomposes the $k \times k \times k$ C3D kernel into $k \times k \times 1$ and $1 \times 1 \times k$. The number of model parameters of each kernel is reduced from k^3 to $k(k+1)$. Such divide-and-conquer heuristic modeling ideas are familiar and usually effective [35,39,40]. Intuitively, assigning clear task requirements to convolution operations can increase their productivity. Therefore, compared with C3D, P3D can not only reduce the number of model parameters but also slightly improve accuracy.

2.3. 3D-Depthwise Separable Convolution

The simultaneous existence of multiple convolution kernels provides a guarantee for the powerful feature extraction capability of CNNs. In fact, the feature maps extracted by multiple convolution kernels can be regarded as many different kinds of features [41]. However, from the comparison of the

two sub-graphs in Figure 1, multiple groups of convolution kernels have brought about several times of parameters. Depthwise separable convolution [35] was proposed as an effective way to reduce the increasing of parameters in this case, which realized a very efficient replacement by decoupling the spatial and channel-wise operations of the vanilla 2D convolution. For an $h \times w \times c$ input map, 2D convolution kernels with the size of $k \times k \times c \times c$ are required to produce the output with the size of $h \times w \times c$ (performing the operation in Figure 1b c times with zero-padding). However, the convolution kernels with the size of $k \times k \times c \times 1 + 1 \times 1 \times c \times c$ are needed for depthwise separable convolution to achieve the same effect. Due to the good performance of depthwise separable convolution in 2D tasks, it is a natural idea to extend it to 3D tasks. A similar idea has also been considered in [36].

The improved strategy is straightforward, that is, replacing the 2D convolutions in 2D-depthwise separable convolution with 3D operations. The comparison between C3D and 3D-depthwise separable convolution is shown in Figure 4. It can be seen from Figure 4a that c times C3D operations are implemented (different colors represent different groups of filters) to generate c numbers of 3D feature cubes. The process of 3D-depthwise separable convolution is shown in Figure 4b. Similar to 2D operations, 3D-depthwise separable convolution can also be divided into depthwise and pointwise operations. The kernels of 3D depthwise convolution are shown in the second column in Figure 4b, and 3D pointwise convolution kernels are shown as the fourth column. Obviously, the idea of depthwise separable convolution is inherited, and an extra dimension is added to implement 3D feature extraction. $k \times k \times k \times c \times c$ numbers of model parameters are needed in Figure 4a, and they can be decomposed into c times 3D depthwise convolution with the parameters of $k \times k \times k$ and c times 3D pointwise convolution with the parameters of $1 \times 1 \times 1 \times c$. Therefore, the model complexity can be greatly reduced, which makes it possible to be utilized with limited resources.

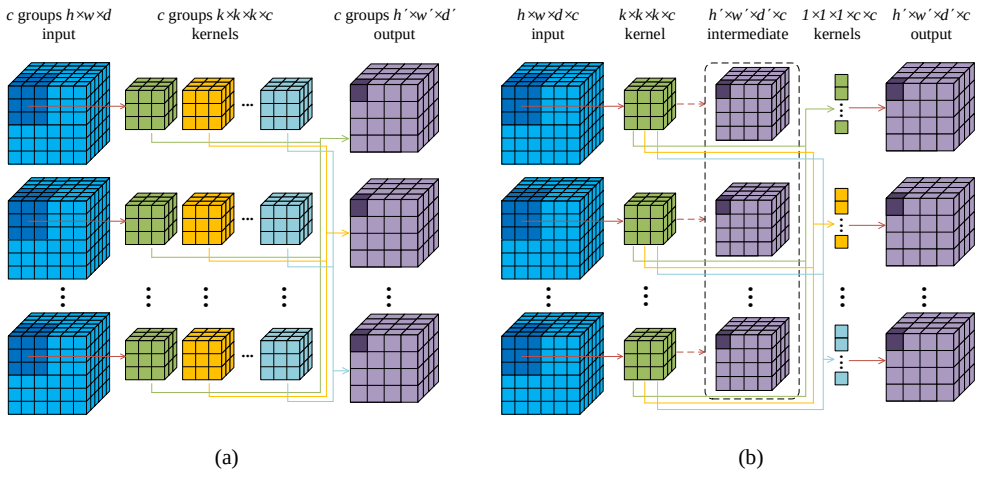


Figure 4. Illustrations of C3D and 3D-depthwise separable convolution with multi groups of kernels. Different filters are coded by different colors, and the convolution kernels within the same group are marked by the same color. (**a**) When the number of the kernels of C3D is c. (**b**) The process of 3D-depthwise separable convolution in the same situation. All 2D operations in depth separable convolution are replaced by 3D operations in (b). Firstly, doing vanilla 3D convolutions on each channel of the input with the kernel size of $k \times k \times k$ (3D depthwise convolution). Then, doing c times $1 \times 1 \times 1$ convolutions to the intermediates (3D pointwise convolution), and the output with the same size of C3D can be obtained.

3. Proposed Methods

In this section, the representation of PolSAR images is present firstly. PolSAR coherence matrix T is adopted as the starting point in this work. Then the implementation details of the proposed architectures are introduced.

3.1. Representation of PolSAR Images

A polarized scattering matrix can fully characterize the electromagnetic scattering properties of ground targets. The scattering matrix is defined as:

$$\left[S \right] = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix}, \tag{3}$$

where $S_{PQ}(P, Q \in \{H, V\})$ 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 $S_{HV} = S_{VH}$. To describe the scattering properties of targets more clearly, the S matrix is usually transformed into the polarization coherence matrix or polarization covariance matrix. The polarization vector and coherence matrix based on Pauli decomposition are expressed as (4) and (5)

$$\vec{k} = \frac{1}{\sqrt{2}}[S_{HH} + S_{VV}, S_{HH} - S_{VV}, 2S_{HV}]^T, \tag{4}$$

$$[T] = \langle \vec{k}\vec{k}^H \rangle. \tag{5}$$

The polarization coherence matrix T is a Hermitian matrix, and all its elements except the diagonal element, are complex numbers. Generally, the upper triangular elements $[T_{11}, T_{12}, T_{13}, T_{22}, T_{23}, T_{33}]$ are taken and divided into their real and imaginary parts as the input of CNNs. At this point, there are nine real-valued numbers to describe each pixel of PolSAR images.

3.2. Lightweight 3D-CNNs for PolSAR Classification

Data preprocessing, model design, and network training and testing are generally included steps of the CNNs-based PolSAR classification methods, as shown in Figure 5.

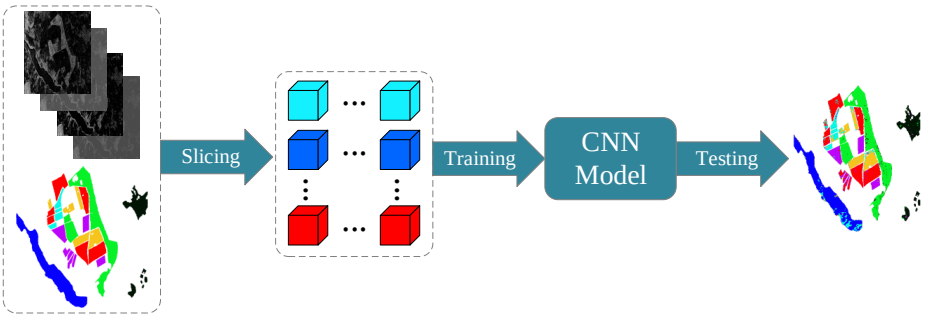


Figure 5. General flow chart of the CNNs-based PolSAR images classification methods.

In this work, the steps of classification can be summarized as follows: (1) Labeled image slices with the size of 15×15 are cut around the central pixel according to the source data of PolSAR image (polarization coherence matrix is used in this work) and the ground truth map. (2) The training set, validation set, and testing set can be obtained from the labeled samples. (3) A tailored CNN architecture is designed according to the characteristics of PolSAR data. (4) The architecture is trained and saved on the training and validation sets, and then tested on the testing set. (5) Each sample of the original data is input to the neural network and the interpretation results of the whole map can be obtained. In most cases, the construction of the CNN architecture is the central part. Some available architectures can be seen in Figure 6.

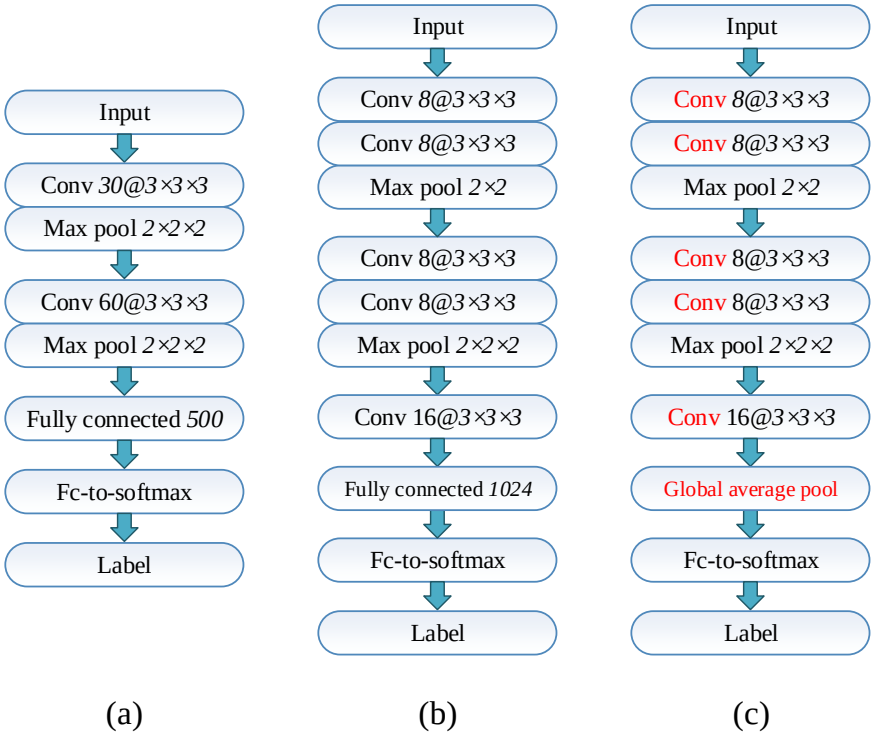


Figure 6. 3D architectures for PolSAR image classification. (**a**) The architecture of 3D-convoluted neural networks (CNN) proposed by [31]. (**b**) The updated version of 3D-CNN in this paper. (**c**) The proposed 3D-CNN framework with lightweight 3D convolutions and global average pooling.

The original 3D-CNN architecture used for PolSAR image classification [31] is shown in Figure 6a. Compared to their work, a deeper architecture can be seen in Figure 6b, in which the updated network has three additional convolution layers and the network width is reduced to alleviate the adverse effects of the increase of depth. Such a 3D architecture can not only mine the spatial relations but also explore the correlations between different elements of the polarization coherence matrix so as to extract more comprehensive information. Therefore, this architecture is chosen to be the backbone of this paper. It is worth noting that building the network backbone is not the objective of this paper, but to compare the proposed lightweight methods with the ordinary ones in a fair environment. Although 3D-CNNs showed a promising performance [31], it also brought a slower interpretation speed due to more model parameters and higher complexity. The computational difficulty mainly centers on the convolution and fully connected layers for the architecture in Figure 6b. Thus, the lightweight improvements designed for these two parts are implemented.

The C3D operations of 3D-CNNs are replaced with the two former introduced lightweight convolution operations to reduce the computational complexity of the convolution layer. It can be seen from Figure 6c that only the way the convolutions is changed, without modifying their depth, width, and kernel size. It can be easily proven that the lightweight convolution layers contain a similar number of model parameters as the 2D layers, and only half or even less of the C3D layers. A more detailed analysis of the changes in the number of model parameters will be given later.

In the architecture shown in Figure 6a, the data is expanded into a 1D vector and enters the fully connected layers when the processing of convolutions is finished. The role of fully connected layers is to reduce the dimension of the outputs of convolution layers. Results of the fully connected layers will be activated by softmax activation to achieve feature classification, which can be defined as

$$\sigma_{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}, \tag{6}$$

where $\sigma_{softmax}(x)$ means the softmax activation of the input x, i denotes the ith category, and j is the number of categories. Thus, a $j \times 1$ vector whose element represents the probability of belonging to the corresponding category can be obtained as the final prediction.

Improvements to the fully connected layer have been concerned a lot because it occupies more than 90% model parameter of CNNs. Global average pooling (GAP) has been proved that it can be seen as a plug-and-play replacement for fully connected layers to save the computational resource [37]. As can be seen from Figure 7a, $m \times m$ three-channels 2D feature maps are flattened to a vector as the input of fully connected layers. When the number of the category is J and the hidden node of the fully connected layers is H, the total number of parameters is $(m \times m \times 3 \times H) + (H \times J)$ for the two fully connected layers. When the input becomes multi-channel 3D features, the large amount of parameters are multiplied by the depth of features. Such a large number of parameters not only brings computational difficulties but also increases the risk of over-fitting. In the proposed architectures, spatial global average pooling is performed as shown in Figure 7b. For each channel of the output feature cube, the process of the used GAP can be defined as

$$y^{(d)} = f_{gap}(x^{(d)}) = \frac{1}{h \times w} \sum_{h} \sum_{w} x_{h,w}^{(d)}, \tag{7}$$

where x and y represent the input and output of the GAP layer. d denotes the depth of the feature cube, and h and w represent its height and width. The above operations are performed on each channel for the multi-channel input 3D feature cubes, which can greatly reduce the number of model parameters so as to cut down the computational cost.

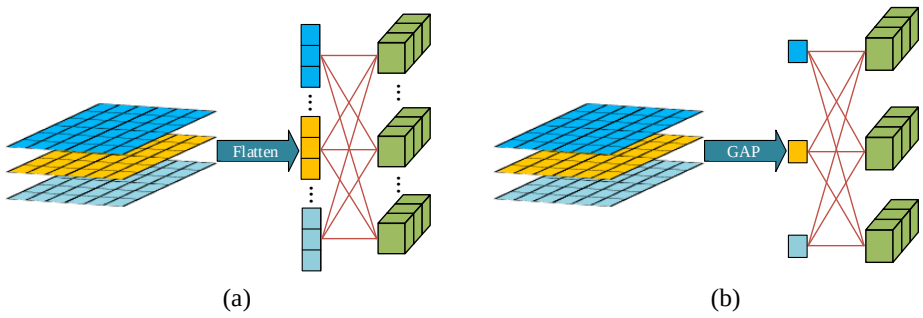


Figure 7. An intuitive comparison between fully connected layer and global average pooling layer for multi-channel 2D input.

4. Experiments

In this section, to evaluate the performance of the proposed methods, they are tested on three PolSAR benchmark datasets and compared with several alternatives. The experimental environment uses a PC with Intel Core i7-7700 CPU with 16 GB RAM. A deep learning toolbox [42] is utilized to minimize the difficulty of algorithm implementation.

4.1. Datasets and Settings

Three widely-used PolSAR benchmark datasets are employed in the experiments: *AIRSAR Flevoland*, *ESAR Oberpfaffenhofen*, and *EMISAR Foulum*. Figures 8–10 show their Pauli maps and ground truth maps, respectively.

4.1.1. AIRSAR Flevoland

As shown in Figure 8, an L-band, full polarimetric image of the agricultural region of the Netherlands is obtained through NASA/Jet Propulsion Laboratory AIRSAR [43]. The size of this image is 750 × 1024 and the spatial resolution is 0.6 m × 1.6 m. The ground truth map is shown

in Figure 8b, which is adapted from [44]. There are 15 kinds of ground objects including buildings, rapeseed, beet, stem beans, peas, forest, lucerne, potatoes, bare soil, grass, barley, water, and three kinds of wheat, and a total of 184,592 image slices are contained in this dataset. The details of each category are shown in Table 1.

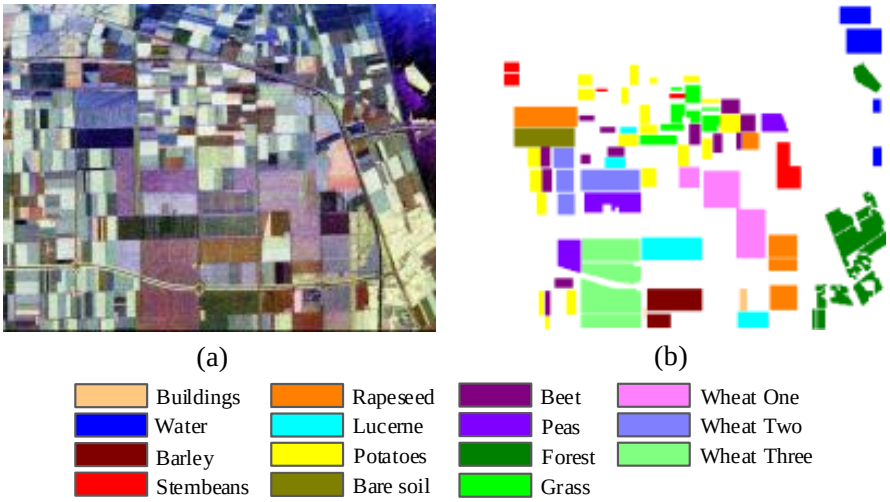


Figure 8. AIRSAR Flevoland dataset. (**a**) Pauli RGB map. (**b**) Ground truth map.

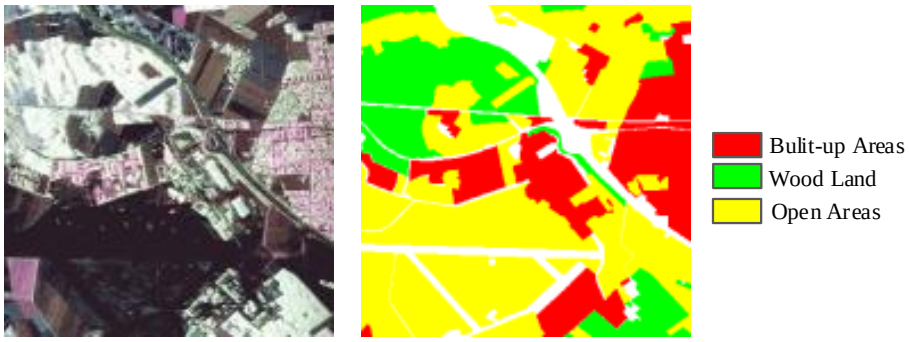
Table 1. Number of pixels in each category for AIRSAR Flevoland.

AIRSAR Flevoland		
Category Code	**Name**	**Reference Data**
1	Buildings	963
2	Rapeseed	17,195
3	Beet	11,516
4	Stem beans	6812
5	Peas	11,394
6	Forest	20,458
7	Lucerne	11,411
8	Potatoes	19,480
9	Bare soil	6116
10	Grass	8159
11	Barley	8046
12	Water	8824
13	Wheat one	16,906
14	Wheat two	12,728
15	Wheat three	24,584
Total	-	184,592

4.1.2. ESAR Oberpfaffenhofen

An L-band, full polarimetric image of Oberpfaffenhofen, Germany, 1200 × 1300 scene size, are obtained through the ESAR airborne platform [43]. Its Pauli color-coded image and ground truth map can be seen in Figure 9. The ground truth map is adapted from [45]. According to the ground truth, each pixel in the map is divided into three categories: built-up areas, wood land, and open

areas, except for some unknown regions. A total of 1,307,142 image slices are contained in this dataset. The details of each category are shown in Table 2.



(a) (b)

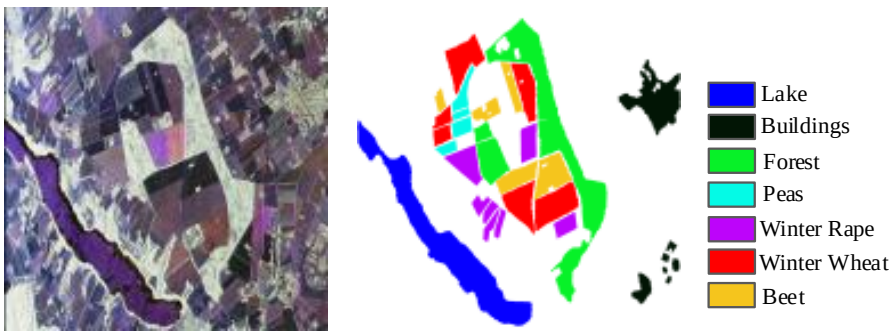
Figure 9. ESAR Oberpfaffenhofen dataset. (**a**) Pauli RGB map. (**b**) Ground truth map.

Table 2. Number of pixels in each category for ESAR Oberpfaffenhofen.

ESAR Oberpfaffenhofen		
Category Code	**Name**	**Reference Data**
1	Built-up areas	310,829
2	Woodland	263,238
3	Open areas	733,075
Total	-	1,307,142

4.1.3. EMISAR Foulum

The last full polarimetric image used in this experiment is the L-band image taken by EMISAR in Foulum, Denmark. EMISAR is a full polarized airborne SAR operating in L and C bands with a resolution of 2 m $\times$ 2 m and mainly acquired and studied by Danish Center for Remote Sensing (DCRS). Figure 10 shows its Pauli RGB image and ground truth map. The size of this image is 1000 $\times$ 1750. The calibration of the terrains in Figure 10b refers to [46,47], and each pixel in the map is divided into seven categories: lake, buildings, forest, peas, winter rape, winter wheat, and beet. There are 431,088 image slices in this dataset. The details of each category are shown in Table 3.



(a) (b)

Figure 10. EMISAR Foulum dataset. (**a**) Pauli RGB map. (**b**) Ground truth map.

Table 3. Number of pixels in each category for the EMISAR Foulum.

EMISAR Foulum		
Category Code	**Name**	**Reference Data**
1	Lake	93,829
2	Buildings	41,098
3	Forest	113,765
4	Peas	26,493
5	Winter rape	37,240
6	Winter wheat	76,401
7	Beet	42,263
Total	-	431,088

4.2. Experiments Starting

To validate the significance of the proposed PolSAR image classification framework, ordinary CNN (CNN), 2D-depthwise separable convolution CNN (DW-CNN) and C3D CNN (3D-CNN) are chosen to be compared. Their architectures and hyperparameters are set as Figure 6b except for the way of convolution. The proposed two classifiers are denoted as P3D-CNN and 3DDW-CNN for convenience. During the training and testing, the size of kernels are 3×3 for 2D convolutions and $3 \times 3 \times 3$ for 3D convolutions. The dropout rate is 0.8 for fully connected layers. An improved stochastic gradient descent optimization method [48] is chosen to train the involved architectures with the learning rate of 0.001.

To evaluate the performance of the algorithms mentioned in this paper, the overall accuracy (OA) and kappa coefficient (Kappa) [49] are chosen as criteria, which can be defined as follows:

$$OA = \frac{\sum_{i=1}^{c} M_i}{\sum_{i=1}^{c} N_i} \tag{8}$$

where c is the number of categories. M_i and N_i denote the number of correctly classified categories and the total number of ith categories, respectively.

$$Kappa = \frac{OA - P}{1 - P}, \; with \; P = \frac{1}{n^2} \sum_{i=1}^{c} H(i,:)H(:,i) \tag{9}$$

where n is the number of testing samples and H denotes the classification confusion matrix.

The number of training epoch is important which determines whether the model converges or not. 9000 and 4500 samples of the AIRSAR Flevoland dataset, and 7000 and 3500 samples of the EMISAR Foulum dataset are randomly chosen without overlaps as training and validation sets. Then experiments of 3D-CNN are carried out to find a suitable value of training epoch. The experimental results are shown in Figure 11.

One can see from the experimental results that the training accuracy tends to be stable after 100 iterations and the validation accuracy does not change much after 200 iterations. Combined with these two points, the value of the epoch is set to 250 in the experiments to ensure convergence. When the training epoch reaches the upper limit, the model with the highest OA on the validation set should be selected as the final trained model to ensure the stability of the training process.

The size of the training set also needs to be carefully considered. We do comparative experiments to find an appropriate number of training samples in order to save memory as much as possible under the premise of guaranteeing the training effects. In the experiments, we randomly extract a certain number of samples (the latter size is twice as large as the former for easy analysis) from each category of labeled samples to form the training set. Two basic models including CNN [12] and 3D-CNN [31]

are tested on different training sets of three benchmarks. One thing can be found that the number of buildings category is small, so in the experiments on the AIRSAR dataset, the number of training samples is fixed to 600 for buildings when the extracted number is more than 600. The experimental results are listed in Table 4, from which we can see that when the scale of the training set moves from small to large, the accuracy of both the CNN and 3D-CNN shows an upward trend. Results on the AIRSAR and EMISAR datasets show that this upward trend eases when the number of training samples of each category exceeds 1200 and 4000. Although the large size of the training set brings a slight improvement for the ESAR dataset, 1000 samples per category met our needs. After obtaining the training set, half of the training set were extracted from the remaining samples to form the validation set, and 30% of the remaining were taken as the testing set.

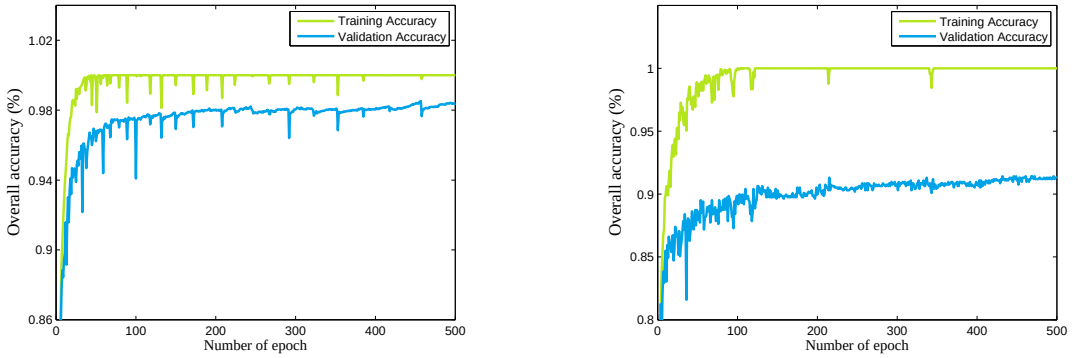


Figure 11. The influence of epoch on the performance of 3D-CNN. (**a**) The results on the AIRSAR Flevoland dataset. (**b**) The results on the EMISAR Foulum dataset.

Table 4. Overall accuracy (%) under different sized training sets.

EMISAR Foulum				ESAR Oberpfaffenhofen				AIRSAR Flevoland			
Num	**CNN**	**3D-CNN**	**Increase**	**Num**	**CNN**	**3D-CNN**	**Increase**	**Num**	**CNN**	**3D-CNN**	**Increase**
500	73.57	76.45	N/A	300	90.10	91.81	N/A	300	76.80	90.47	N/A
1000	79.26	83.11	5.69/6.66	600	91.26	92.75	1.16/0.94	600	87.99	95.40	11.19/4.93
2000	83.81	87.15	4.55/4.04	1000	92.36	93.83	1.10/1.08	1200	93.46	97.21	5.47/1.81
4000	87.39	89.15	3.58/2.00	2000	92.97	94.22	0.61/0.39	2400	93.61	97.55	0.15/0.34
6000	87.75	89.67	0.36/0.52	4000	93.02	94.57	0.05/0.35	3600	93.83	97.58	0.22/0.03

4.3. Results and Comparisons

Under the experimental environment and settings described earlier, the classification results of different methods are shown in Figures 12–14, and the accuracies are listed in Tables 5–7, respectively. Generally, the proposed methods achieve better performance than the compared ones. The experimental results on the AIRSAR Flevoland dataset can be seen from Table 5 and the classification results of the whole map are listed in Figure 12.

The results in Table 5 prove that the proposed methods slightly improve the classification accuracy on this data set. From the experimental results, it can be seen that 3D networks have a better performance than 2D networks, which confirms the importance of 3D convolutions for the PolSAR classification. Furthermore, it can be seen that the OA and Kappa of lightweight 3D convolution-based methods are higher and ordinary 3D-CNN, especially in the identification of the rapeseed and wheat categories. This shows that there is potential redundancy in C3D operations and the lightweight strategies can improve not only the computational efficiency but also the classification performance.

Table 5. Classification results (%) on the AIRSAR Flevoland dataset.

Category	CNN [12]	DW-CNN [35]	3D-CNN [31]	P3D-CNN	3DDW-CNN
1	100.00	100.00	100.00	100.00	100.00
2	82.38	92.65	90.63	95.65	96.55
3	93.20	96.68	96.50	96.25	97.98
4	98.18	99.45	99.05	99.48	99.55
5	96.60	98.10	98.55	98.85	99.05
6	94.70	98.38	97.40	98.68	95.93
7	93.60	97.15	98.83	98.80	98.73
8	90.53	97.60	96.88	97.10	97.15
9	98.68	98.63	99.73	94.38	98.98
10	95.48	96.03	97.03	96.50	95.98
11	90.98	96.45	98.23	99.65	99.55
12	97.50	99.28	100.00	100.00	100.00
13	91.85	97.80	97.33	99.25	97.35
14	91.04	93.42	93.22	97.05	94.52
15	92.15	95.65	96.85	99.53	96.50
OA	93.46	97.00	97.21	97.97	97.74
Kappa	92.97	96.77	97.00	97.82	97.57

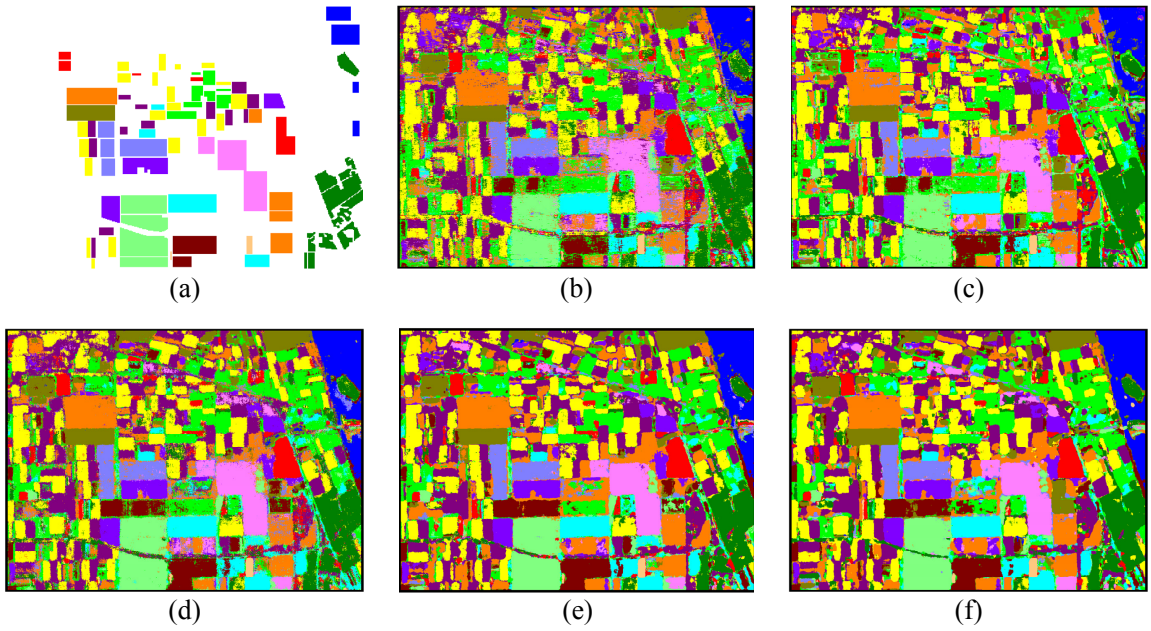


Figure 12. Classification results of the whole map on the AIRSAR Flevoland data with different methods. (**a**) Ground truth. (**b**) Result of CNN. (**c**) Result of depthwise separable (DW)-CNN. (**d**) Result of 3D-CNN. (**e**) Result of P3D-CNN. (**f**) Result of 3D-depthwise separable convolution-based CNN (3DDW-CNN).

Whole map classification results can be seen in Figure 12, it can be seen that the proposed methods have more powerful capabilities for distinguishing between forest and grass. In addition, apart from rapeseed and three types of wheat, the proposed methods are also effective for classifying beet and potatoes.

The experimental results on ESAR Oberpfaffenhofen can be seen from Table 6 and the classification results of the whole map are listed in Figure 13. On this data set, the analysis results are generally consistent with the previous ones. The 3D models achieve better results than the 2D models under different criteria. In these experiments, 3DDW-CNN achieves the best performance. It has a 1.37% improvement of OA and 2.04% improvement of kappa compared with the ordinary 3D-CNN. The P3D-based model also achieves a several signs of progress. Similar conclusions under different datasets also confirm the generalization performance of the proposed methods.

Table 6. Classification results (%) on ESAR Oberpfaffenhofen dataset.

Category	CNN [12]	DW-CNN [35]	3D-CNN [31]	P3D-CNN	3DDW-CNN
1	89.19	91.25	92.14	94.27	92.93
2	93.35	93.97	94.85	94.44	95.79
3	94.55	93.85	94.51	94.99	96.87
OA	92.36	93.02	93.83	94.53	95.20
Kappa	88.54	89.53	90.75	91.63	92.79

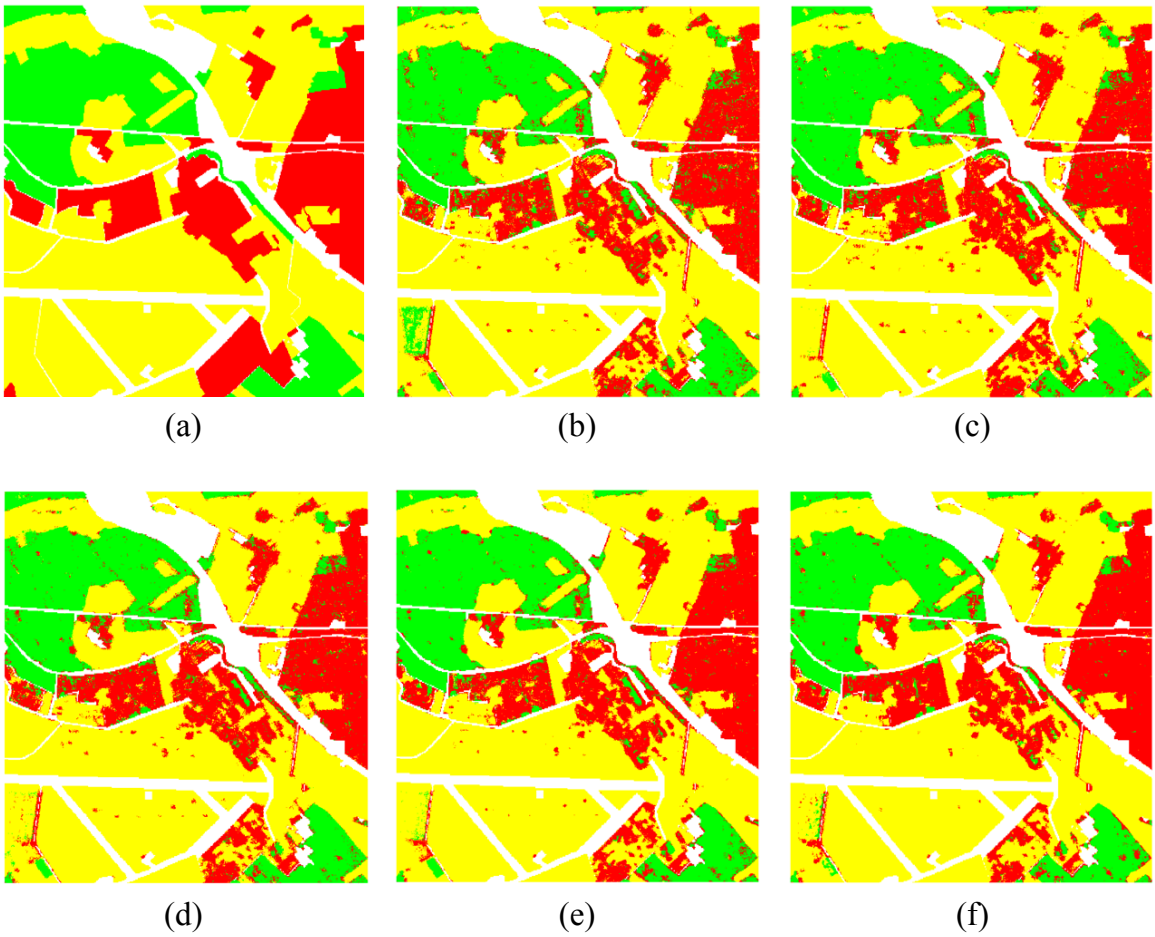


Figure 13. Classification results overlaid with the ground truth map on ESAR Oberpfaffenhofen data with different methods. (**a**) Ground truth. (**b**) Result of CNN. (**c**) Result of DW-CNN. (**d**) Result of 3D-CNN. (**e**) Result of P3D-CNN. (**f**) Result of 3DDW-CNN.

The results overlaid with the ground truth map on ESAR Oberpfaffenhofen are shown in Figure 13, where it can be seen that serious confusions exist between built-up areas and woodlands for 2D models.

This phenomenon has been weakened in 3D-CNN, and the proposed methods further alleviate this problem. In addition, compared with other methods, the proposed methods have more complete and pure classification results for the open areas.

The experimental results on the EMISAR Foulum can be seen from Table 7 and the classification results of the whole map are listed in Figure 14. Compared with the former two datasets, EMISAR Foulum data which contains quite complex terrain information is not so widely used. Similar conclusions can be drawn from the analysis of the experimental results shown in Table 7, where the proposed P3D-CNN achieves the best classification results. It is worth pointing out that although the results of 3DDW-CNN is slightly lower than 3D-CNN, such a small performance degradation (about 0.03%) is acceptable under the premise of reducing computational complexity.

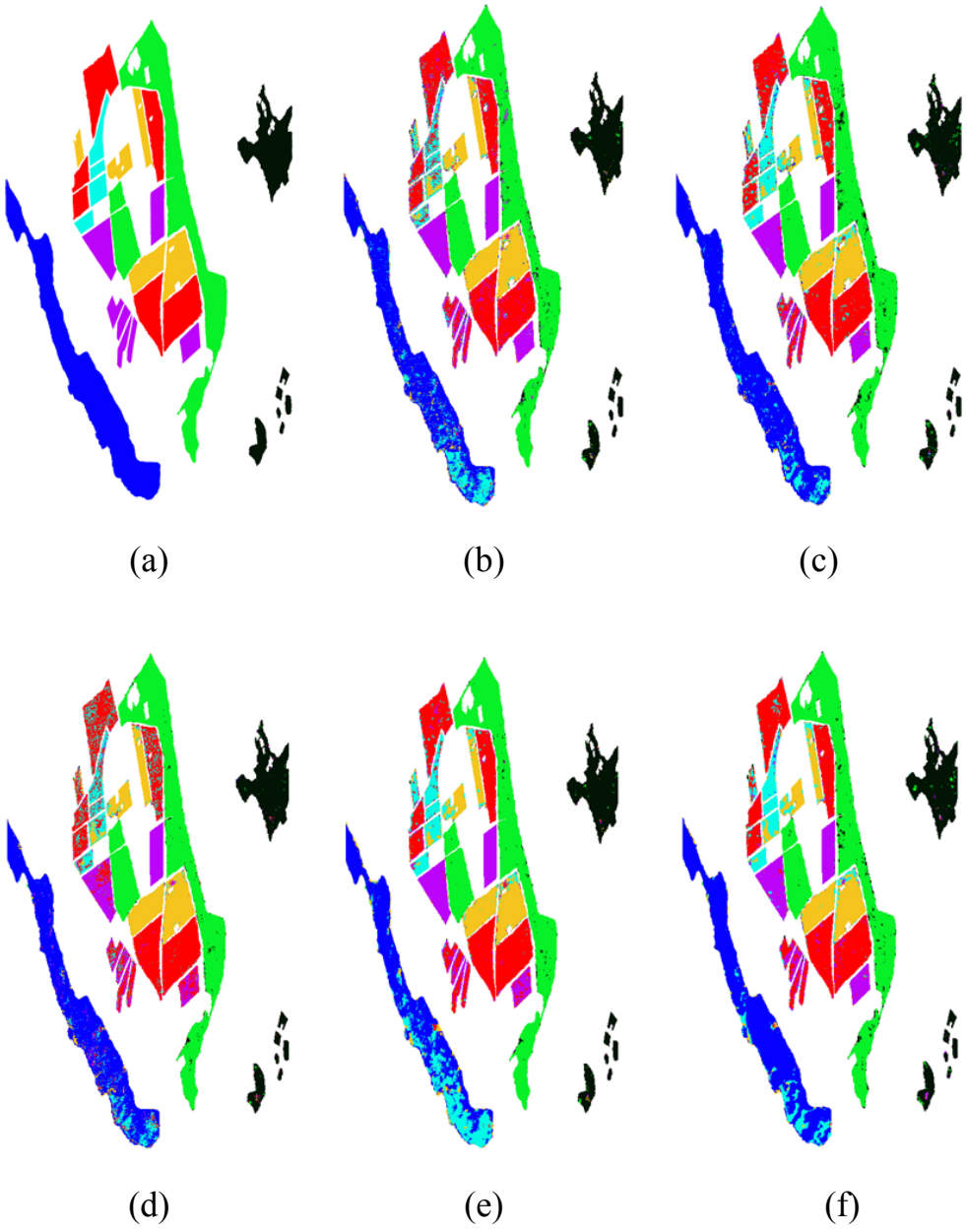


Figure 14. Classification results overlaid with the ground truth map on the EMISAR Foulum data with different methods. (**a**) Ground truth. (**b**) Result of CNN. (**c**) Result of DW-CNN. (**d**) Result of 3D-CNN. (**e**) Result of P3D-CNN. (**f**) Result of 3DDW-CNN.

Table 7. Classification results (%) on the EMISAR Foulum dataset.

Category	CNN [12]	DW-CNN [35]	3D-CNN [31]	P3D-CNN	3DDW-CNN
1	88.66	92.16	94.06	94.56	94.32
2	97.48	95.32	99.10	97.34	98.04
3	97.10	96.06	98.46	99.10	98.30
4	71.48	73.84	69.96	80.12	74.28
5	84.24	87.14	84.28	83.08	82.52
6	81.70	86.17	84.97	86.06	86.75
7	91.04	92.18	93.16	90.24	89.58
OA	87.39	88.99	89.15	90.08	89.12
Kappa	85.29	87.15	87.34	88.43	87.30

One can see from Figure 14 that the following groups of objects are easy to be misclassified, including lake-peas, peas-winter wheat, buildings-forest. The proposed methods show competitive performance when generally solving the above problems, although the results of P3D-CNN for the lake is not very good.

4.4. Studies of Complexity

In previous experiments, the classification performance of the proposed methods have been verified. In this part, we analyze the number of trainable parameters and the computational complexity of the proposed methods. An intuitive comparison of the number of trainable parameters and overall accuracy of the involved models on the AIRSAR Flevoland dataset can be seen in Table 8.

Table 8. Comparison of the number of contained model parameters between different architectures on the AIRSAR Flevoland dataset.

Method	OA	Conv Param	Reduced	Total Param	Reduced
3D-CNN [31]	97.21%	10,632	N/A	1,075,607	N/A
P3D-CNN	97.97%	5160	51.47%	6135	99.43%
3DDW-CNN	97.74%	3208	69.83%	4183	99.61%
CNN [12]	93.46%	3576	N/A	73,223	N/A
DW-CNN [35]	97.00%	809	77.38%	70,456	3.78%
P3D-CNN	97.97%	5160	–44.30%	6135	91.62%
3DDW-CNN	97.74%	3208	10.29%	4183	94.29%

As can be seen from Table 8, P3D-CNN contains half of the parameters of 3D-CNN in convolution layers, which is 1.44 times that of 2D-CNN. 3DDW-CNN is even lighter, which cuts about 70% trainable parameters in convolution layers of 3D-CNN. As the GAP is introduced to replace the fully connected layer, the total parameters contained in the model are greatly reduced. Meanwhile, the proposed two methods not only maintain the accuracy of 3D-CNN but also improve slightly.

Furthermore, the value of the floating point operations (FLOPs) in the convolution layers of each method is calculated, which is a popular evaluation metric to compare the complexity of algorithms. FLOPs in convolution layers of the proposed and comparing methods are calculated. Then the comparison combining accuracy and complexity can be seen from Figure 15, in which the x-axis represents the value of convolution FLOPs, and the y-axis represents the overall accuracy. Four involved methods, i.e., CNN, two proposed ones, and 3D-CNN, are shown in the figure from the left to the right. Each one has three bars, which represent its OA on the three different datasets.

It can be seen that the proposed methods, i.e., the middle two of the four columns, not only have lower FLOPs, but also improve the classification accuracy slightly than 3D-CNN (the rightmost column). This result can verify the theoretical analysis.

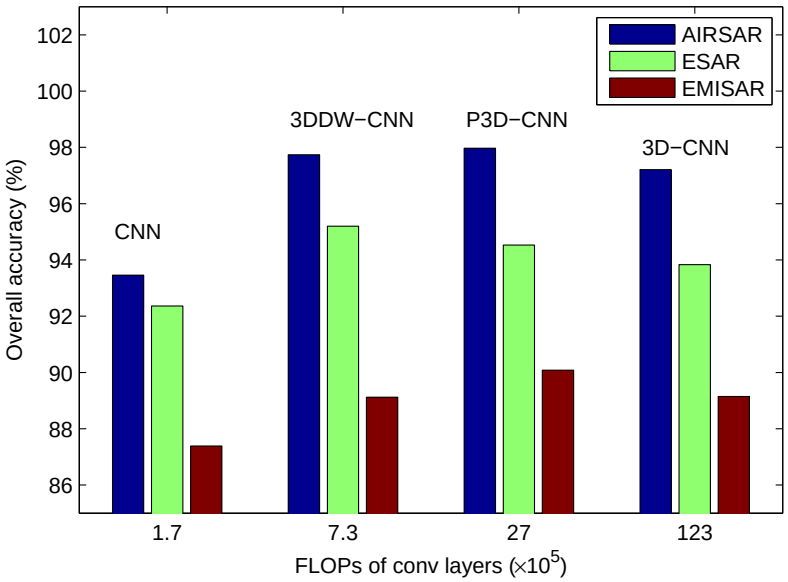


Figure 15. Comparisons of accuracy and complexity.

5. Conclusion

Inspired by the recent lightweight improvements for deep neural networks, in this paper, two lightweight 3D-CNN architectures are proposed for PolSAR image classification. Lightweight 3D convolutions, i.e., pseudo-3D and 3D-depthwise separable convolutions, are introduced to perform feature extraction and reduce the redundancy of 3D convolutions. Meanwhile, global average pooling is introduced to replace the fully connected layer considering the huge number of model parameters contained in it. In this way, over 90% model parameters of 3D-CNNs can be compressed so as to support the high-precision interpretation under the resource-constrained system. Moreover, a general lightweight 3D-CNN framework can be summarized, which can help future research. Such a PolSAR tailored classification framework can not only improve the running speed but also boost the performance of convolutions. Experimental results on three PolSAR benchmark datasets show that the proposed architectures have promising classification performance and low computational complexity. In the future, complex-valued CNN architectures, weakly-supervised classification methods and finding the optimal hyperparameters automatically are all issues we are considering.

Author Contributions: All the authors made significant contributions to this work. H.D. and L.Z. devised the approach and wrote the paper. H.D. conducted the experiments and analyzed the data. Supervision and suggestions, L.Z. and B.Z.; writing—review and editing, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (61401124, 61871158), in part by Scientific Research Foundation for the Returned Overseas Scholars of Heilongjiang Province (LC2018029), in part by Aeronautical Science Foundation of China (20182077008)

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. doi:10.1038/nature14539. [CrossRef]

2. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105. doi:10.1145/3065386. [CrossRef]
3. Lardeux, C.; Frison, P.; Tison, C.; Souyris, J.; Stoll, B.; Fruneau, B.; Rudant, J. Support vector machine for multifrequency SAR polarimetric data classification. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 4143–4152. doi:10.1109/TGRS.2009.2023908. [CrossRef]
4. Zhu, X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. doi:10.1109/MGRS.2017.2762307. [CrossRef]
5. Ding, J.; Chen, B.; Liu, H.; Huang, M. Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. doi:10.1109/LGRS.2015.2513754. [CrossRef]
6. Chen, S.; Wang, H.; Xu, F.; Jin, Y. Target classification using the deep convolutional networks for SAR images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. doi:10.1109/TGRS.2016.2551720. [CrossRef]
7. Pei, J.; Huang, Y.; Huo, W.; Zhang, Y.; Yang, J.; Yeo, T. SAR automatic target recognition based on multiview deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2196–2210. doi:10.1109/TGRS.2017.2776357. [CrossRef]
8. Ren, Z.; Hou, B.; Wen, Z.; Jiao, L. Patch-sorted deep Feature Learning for high resolution SAR image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3113–3126. doi:10.1109/JSTARS.2018.2851023. [CrossRef]
9. Gong, M.; Zhao, J.; Liu, J.; Miao, Q.; Jiao, L. Change detection in synthetic aperture radar images based on deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 125–138. doi:10.1109/TNNLS.2015.2435783. [CrossRef]
10. Corentin, H.; Azimi, S.; Merkle, N. Road segmentation in SAR satellite images with deep fully convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1867–1871. doi:10.1109/LGRS.2018.2864342. [CrossRef]
11. Jiao, L.; Liu, F. Wishart deep stacking network for fast PolSAR image classification. *IEEE Trans. Image Process.* **2016**, *25*, 3273–3286. doi:10.1109/TIP.2016.2567069. [CrossRef]
12. Zhou, Y.; Wang, H.; Xu, F.; Jin, Y. Polarimetric SAR image classification using deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *13*, 1935–1939. doi:10.1109/LGRS.2016.2618840. [CrossRef]
13. Bi, H.; Sun, J.; Xu, Z. A graph-based semisupervised deep learning model for PolSAR image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2116–2132. doi:10.1109/TGRS.2018.2871504. [CrossRef]
14. Yan, W.; Chu, H.; Liu, X.; Liao, M. A hierarchical fully convolutional network integrated with sparse and low-rank subspace representations for PolSAR imagery classification. *Remote Sens.* **2018**, *10*, 342. doi:10.3390/rs10020342. [CrossRef]
15. De, S.; Bruzzone, L.; Bhattacharya, A.; Bovolo, F.; Chaudhuri, S. A novel technique based on deep learning and a synthetic target database for classification of urban areas in PolSAR data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 154–170. doi:10.1109/JSTARS.2017.2752282. [CrossRef]
16. Dong, H.; Zhang, L.; Zou, B. Densely connected convolutional neural network based polarimetric SAR image classification. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, Japan, 28 July–2 August 2019; pp. 3764–3767. doi:10.1109/IGARSS.2019.8900292. [CrossRef]
17. Geng, J.; Ma, X.; Fan, J.; Wang, H. Semisupervised classification of polarimetric SAR image via superpixel restrained deep neural network. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 122–126. doi:10.1109/LGRS.2017.2777450. [CrossRef]
18. Bi, H.; Xu, F.; Wei, Z.; Xue, Y.; Xu, Z. An active deep learning approach for minimally supervised PolSAR image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9378–9395. doi:10.1109/TGRS.2019.2926434. [CrossRef]
19. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. *arXiv* **2015**, arXiv:1505.04597.
20. Chen, S.; Tao, C. PolSAR image classification using polarimetric-feature-driven deep convolutional neural network. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 627–631. doi:10.1109/LGRS.2018.2799877. [CrossRef]
21. Hänsch, R. Complex-valued multi-layer perceptrons-An application to polarimetric SAR data. *Photogramm. Eng. Remote Sens.* **2010**, *76*, 1081–1088. doi:10.14358/PERS.76.9.1081. [CrossRef]

22. Hänsch, R.; Hellwich, O. Complex-valued convolutional neural networks for object detection in PolSAR data. In Proceedings of the 8th European Conference on Synthetic Aperture Radar (EUSAR), Aachen, Germany, 7–10 June 2010; pp. 1–4.
23. Zhang, Z.; Wang, H.; Xu, F.; Jin, Y. Complex-valued convolutional neural network and its application in polarimetric SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7177–7188. doi:10.1109/TGRS.2017.2743222. [CrossRef]
24. Shang, R.; Wang, G.; Michael, A.; Jiao, L. Complex-valued convolutional autoencoder and spatial pixel-squares refinement for polarimetric SAR image classification. *Remote Sens.* **2019**, *11*, 522. doi:10.3390/rs11050522. [CrossRef]
25. Cao, Y.; Wu, Y.; Zhang, P.; Liang, W.; Li, M. Pixel-wise PolSAR image classification via a novel complex-valued deep fully convolutional network. *Remote Sens.* **2019**, *11*, 2653. doi:10.3390/rs11222653. [CrossRef]
26. Sun, Q.; Li, X.; Li, L.; Liu, X.; Liu, F.; Jiao, L. Semi-supervised complex-valued GAN for polarimetric SAR image classification. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, Japan, 29 July–2 August 2019; pp. 3245–3248. doi:10.1109/IGARSS.2019.8898217. [CrossRef]
27. Liu, X.; Tu, M.; Wang, Y.; He, C. Polarimetric phase difference aided network for PolSAR image classification. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Valencia, Spain, 22–27 July 2018; pp. 6667–6670. doi:10.1109/IGARSS.2018.8517572. [CrossRef]
28. Zhang, L.; Dong, H.; Zou, B. Efficiently utilizing complex-valued PolSAR image data via a multi-task deep learning framework. *ISPRS J. Photogramm. Remote Sens.* **2019**, *157*, 59–72. doi:10.1016/j.isprsjprs.2019.09.002. [CrossRef]
29. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. doi:10.1109/TPAMI.2012.59. [CrossRef]
30. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497. doi:10.1109/ICCV.2015.510. [CrossRef]
31. Zhang, L.; Chen, Z.; Zou, B. Polarimetric SAR terrain classification using 3D convolutional neural network. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium(IGARSS), Valencia, Spain, 22–27 July 2018; pp. 4551–4554. doi:10.1109/IGARSS.2018.8519557. [CrossRef]
32. Tan, X.; Li, M.; Zhang, P.; Wu, Y.; Song, W. Complex-valued 3-D convolutional neural network for PolSAR image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, in press. doi:10.1109/LGRS.2019.2940387. [CrossRef]
33. Chen, H.; Zhang, F.; Tang, B.; Yin, Q.; Sun, X. Slim and efficient neural network design for resource-constrained SAR target recognition. *Remote Sens.* **2018**, *10*, 1618. doi:10.3390/rs10101618. [CrossRef]
34. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3D residual networks. In Proceedings of IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5534–5542. doi:10.1109/ICCV.2017.590. [CrossRef]
35. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. doi:10.1109/CVPR.2017.195. [CrossRef]
36. Ye, R.; Liu, F.; Zhang, L. 3D depthwise convolution: Reducing model parameters in 3D vision tasks. *arXiv* **2018**, arXiv:1808.01556.
37. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958. doi:10.1162/neco.1989.1.4.541. [CrossRef]
39. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *arXiv* **2014**, arXiv:1406.2199.
40. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. doi:10.1109/CVPR.2017.634. [CrossRef]

41. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141. doi:10.1109/CVPR.2018.00745. [CrossRef]
42. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
43. Earth Online. Available online: http://envisat.esa.int/POLSARpro/datasets.html2 (accessed on 1 December 2019).
44. Yu, P.; Qin, A.; Clausi, D. Unsupervised polarimetric SAR image segmentation and classification using region growing with edge penalty. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1302–1317. doi:10.1109/TGRS.2011.2164085. [CrossRef]
45. Liu, B.; Hu, H.; Wang, H.; Wang, K.; Liu, X.; Yu, W. Superpixel-based classification with an adaptive number of classes for polarimetric SAR images. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 907–924. doi:10.1109/TGRS.2012.2203358. [CrossRef]
46. Skriver, H.; Dall, J.; Le Toan, T.; Quegan, S.; Ferro-Famil, L.; Pottier, E.; Lumsdon, P.; Moshammer, R. Agriculture classification using PolSAR data. In Proceedings of the 2nd International Workshop on POLinSAR, Frascati, Italy, 17–21 January 2005; pp. 213–218.
47. Conradsen, K.; Nielsen, A.; Schou, J.; Skriver, H. A test statistic in the complex wishart distribution and its application to change detection in polarimetric SAR data. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 4–19. doi:10.1109/TGRS.2002.808066. [CrossRef]
48. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.
49. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. doi:10.1177/001316446002000104. [CrossRef]