

PolSARNet: A Deep Fully Convolutional Network for Polarimetric SAR Image Classification

Adugna G. Mullissa , *Member, IEEE*, Claudio Persello , *Senior Member, IEEE*, and Alfred Stein 

Abstract—Deep learning has successfully improved the classification accuracy of optical remote sensing images. Recent works attempted to transfer the success of these techniques to the microwave domain to classify Polarimetric SAR data. So far, most deep learning networks separate amplitude and phase as separate input images. In this article, we present a deep fully convolutional network that uses real-valued weight kernels to perform pixel-wise classification of complex-valued images. We evaluated the performance of this network by comparing it with support vector machine, Random Forest, complex-valued convolutional neural network (CV-CNN), and a network that uses amplitude and phase information separately as real channels. The evaluation was done on a quad-polarized AIRSAR image and a dual-polarimetric multi-temporal Sentinel-1 data acquired over Flevoland, the Netherlands. The proposed method achieved higher accuracy compared to all other networks with the same architecture.

Index Terms—Convolutional neural network (CNN), deep learning, image classification, machine learning, polarimetric SAR (PolSAR).

I. INTRODUCTION

DEEP learning for remote sensing images refers to a collection of techniques that are capable of learning a hierarchical representation of data for classification and object detection [1]. The core characteristics of a deep learning technique, such as a convolutional neural network (CNN), is the automatic learning of hierarchical spatial features from remote sensing images. Successful demonstration of CNN for the classification of remote sensing data has been reported in [2]–[6].

Standard techniques for fully polarimetric synthetic aperture radar (SAR) data classification are based on the polarimetric coherency or covariance matrices. These methods focus on first estimating the coherency or covariance matrix by reducing speckle followed by feature extraction and classification. Lee *et al.* [7] derived a complex Wishart distance measure to classify pixels, whereas [8]–[11] used polarimetric decomposition to extract the

different scattering mechanisms. A hierarchical classification of the three canonical scattering mechanisms was used in [12] to perform terrain classification. For SAR image classification speckle reduction, feature extraction and classification steps are performed separately to classify the polarimetric SAR (PolSAR) images.

Several researchers transferred the success of CNN on optical remote sensing images to the classification of SAR images. An SAR image, due to its coherent nature, is inherently complex-valued. Initial attempts considered only the amplitude. Chen *et al.* [13] used a deep CNN on the MSTAR-10 class datasets, and Zhou *et al.* [14] applied CNN to a fully polarimetric AIRSAR data. This has several drawbacks, as the information in a SAR image, particularly the phase, is best represented as a complex number [15], [16]. Complex-valued neural networks were pioneered by [17], [18] and applied in relation to signal coherence and filtering. Although complex-valued neural networks have been used in medical imaging and other signal processing applications, they have so far attracted little attention in the remote sensing community. Danilla *et al.* [19] and Mullissa *et al.* [20] used a multitemporal Sentinel-1 amplitude images to classify crops. Recently, Zhang *et al.* [21] demonstrated that a higher classification accuracy can be achieved by jointly using amplitude and phase information by extending the real-valued CNN to the complex domain for the application of landcover classification from fully polarimetric airborne sensors.

So far, CNNs have been trained from patches of SAR data using a series of convolution, activation, and subsampling networks followed by one or several fully connected layers. Once the network was trained, testing was done per patch in a sliding-window manner to label the center pixel in the patch. Such testing is inefficient and time-consuming [22], while fully convolutional networks (FCN) label the image on a pixel-to-pixel basis without the redundant processing of an entire patch to label one pixel, thereby expediting the computational efficiency.

The complex-valued CNN (CV-CNN) proposed in [21] uses complex-valued weight kernels, applied on the complex-valued image data. This procedure scales and rotates the input data in the complex domain extracting features from a spatial neighborhood. The CV-CNN replaces each convolutional layer in the network by four real-valued convolutions that correspond to the individual real and imaginary parts of the complex tensor and weight kernel. When applying a deep neural network consisting of a large number of filter banks, the amount of real-valued computations becomes large. Furthermore, as reported in [15], the training of CV-CNN becomes increasingly unstable, as it depends upon the complexity of the network. Hence, we suggest

Manuscript received May 4, 2019; revised October 6, 2019 and November 12, 2019; accepted November 25, 2019. This research work was funded internally from the Faculty of Geo-information Science and Earth Observation, University of Twente research fund. (*Corresponding author: Adugna G. Mullissa.*)

A. G. Mullissa is with the Laboratory of Geo-Information Science and Remote Sensing, Wageningen University, Wageningen 6700 AA, The Netherlands (e-mail: adugna.mullissa@wur.nl).

C. Persello and A. Stein are with the Department of Earth Observation Science, Faculty of Geo-information Science and Earth Observation, University of Twente, Enschede 7514 AE, The Netherlands (e-mail: c.persello@utwente.nl; a.stein@utwente.nl).

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

Digital Object Identifier 10.1109/JSTARS.2019.2956650

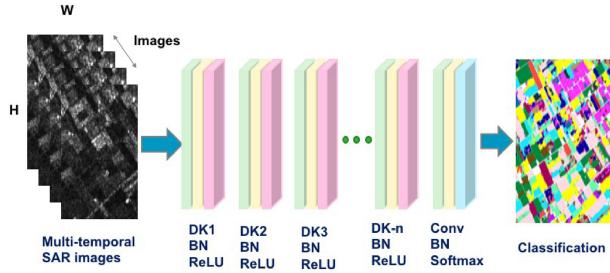


Fig. 1. PolSARNet architecture.

a computationally less expensive deep hybrid fully convolutional network that uses real-valued weight kernels. Convolution of a complex-valued data tensor with real-valued weights yields a complex-valued output at each layer in the network. Hence, we use a complex-valued activation, batch normalization, and backpropagation to train this network. Such a method that is more stable and computationally tenable than the CV-CNN concepts is introduced in [21]. It uses real-valued weight kernels and takes as input the complex data of the polarimetric coherency matrix. It formulates the building blocks of a deep fully convolutional network in the complex domain. The proposed method (PolSARNet) is evaluated on a fully polarimetric AIRSAR dataset acquired over Flevoland, the Netherlands and a Sentinel-1 dual polarimetric multitemporal data acquired over the same area. It is compared with real-valued FCN with the same architecture and hyperparameters.

The novel contributions of this article are as follows.

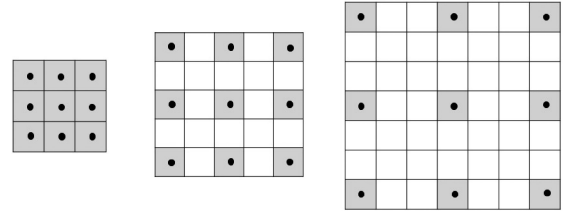
- 1) *Real-valued parameters network for complex-valued data*: A real-valued weight kernels network processes complex-valued data tensors to extract spatial features in a deep fully convolutional neural network. We apply complex-valued activations, batch normalization layers, and backpropagation for training.
- 2) *Demonstrate a robust architecture for multitemporal PolSAR data classification*: The performance of the proposed method is analysed in classifying a multitemporal dual polarimetric Sentinel-1 dataset.

II. POLSARNET

PolSARNet uses real-valued weight kernels to extract spatial features from complex-valued polarimetric data in the complex domain. To this effect, the building blocks of the neural network are redesigned. It contains a feature extraction block consisting of convolutions that are interleaved with activation and batch normalization layers (Fig. 1). After a series of feature extraction steps, a loss function derives the prediction error from the network. A backpropagation step updates the kernel weights in the initial weight kernels. These steps are repeated iteratively until a termination condition is met.

A. Building Blocks of PolSARNet

1) *Complex-Valued Convolutions With Real-Valued Dilated Kernels*: The convolutional block proposed in this article applies convolution to the input complex-valued image with a bank of real-valued filter kernels and adds a real-valued bias term. The learnable weights of the filter bank are represented by a 4-D tensor $\mathbf{w} \in \mathbb{R}^{H \times W \times D \times K}$ and a bias term $\mathbf{b} \in \mathbb{R}$. Here, H

Fig. 2. Kernel dilation for an arbitrary 3×3 filter kernel. From left to right, a dilation factor $d = 1$, $d = 2$, and $d = 3$.

and W are the height and width of the kernel, D is the number of input feature channels, and K is the number of filters. For multi-temporal polarimetric SAR images, each image in the coherency matrix is concatenated with each other. Additional parameters control the size of the output from the convolution operation. The stride s is the number of pixels between convolution centers. Padding consists of adding p zeros around the border of the input image before applying the filter so that the input image and the output feature map have the same dimension. Given an input image $\mathbf{x} \in \mathbb{C}^{M \times N \times D}$ designated as $\mathbf{x} = \mathbf{r} + i\mathbf{z}$, the k th channel of the complex-valued output feature map \mathbf{y} equals [22]

$$\mathbf{y}_{qrk} = b_k + \sum_{i=1}^H \sum_{j=1}^W \sum_{d=1}^D \mathbf{w}_{ijdk} \otimes \mathbf{x}_{s(q-1)+i-p, s(r-1)+j-p, d} \quad (1)$$

where \otimes is the complex convolution operator. The size of the output feature map equals $M' \times N' \times K = \lfloor \frac{M-H+2p}{s} + 1 \rfloor \times \lfloor \frac{N-W+2p}{s} + 1 \rfloor \times K$. In the complex domain, convolution is distributive so we can decompose the complex convolution operator \otimes as

$$\mathbf{x} \otimes \mathbf{w} = (\mathbf{r} * \mathbf{w}) + i(\mathbf{z} * \mathbf{w}) \quad (2)$$

where $*$ is a real-valued convolution.

To increase the spatial receptive field of each filter without increasing the number of learnable parameters, we adopt a kernel dilation. This is obtained by inserting d zeros between kernel elements [23] and effectively achieves large spatial support without increasing the number of learnable parameters. A dilation factor $d = 2$ causes a 1-D kernel $[f_1, f_2]$ to be transformed to $[f_1, 0, 0, f_2]$. Hence, a dilation factor of d alters a filter of size (H, W) to a filter size of $\tilde{H} = d(H - 1) + 1$ and $\tilde{W} = d(W - 1) + 1$ (Fig. 2).

Real-valued convolutions represented by conventional CNNs can be interpreted as a heat map of similarity in between the data and kernel. The dot product representing the convolution operation is maximized when the data and the kernel assume similar values. Similarly, in complex-valued convolution, the output of the complex-valued convolution is maximized when the amplitude of the data and the filter kernel assume similar values. Since the amplitude of a complex number is the square root of the squares of the sum of the real and imaginary parts, it is evident that different filter kernels yield the same amplitude with different phase angles. This leads to inconsistencies in extracting different features from images [15]. The convolution operation used in [21] transforms both the amplitude and phase values in the complex domain. Hence, the complex-valued network trains the complex-valued weight kernel to align with the complex-valued data in the complex domain to obtain higher amplitudes in

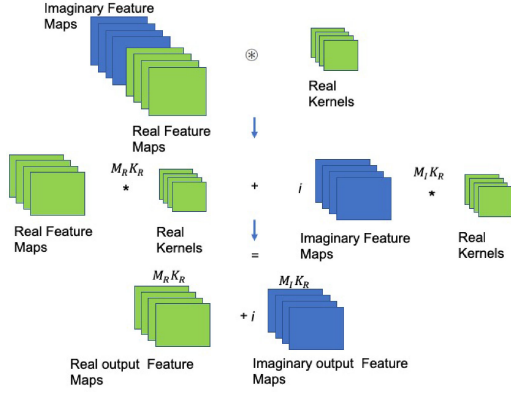


Fig. 3. An illustration of a complex-valued input data convolution with a real-valued weight kernels. In the figure, M_R , M_I , and K_R represent real feature maps, imaginary feature maps, and real kernel weights.

the output feature maps which can be extracted by the activation layer. However, in some instances, a unique position is absent for the filter kernels in the complex domain to achieve higher amplitude in the output feature map, thus presenting an ambiguity [15]. To avoid this ambiguity, we designed the convolution operation of the complex-valued data with a real-valued kernel to scale the amplitude while preserving the phase of the input data tensor. This enhances the amplitude of the input data while preserving the phase, thus avoiding the ambiguity described in the previous sentence. Hence, the convolution with real-valued kernels gives the opportunity to extract features by preserving the representation of the complex tensor (Fig. 3). Since the phase difference in a PolSAR data possess spatial structure, i.e. the phase difference values for different ground features are unique, it is desirable not to change the original phase difference values. Hence, the real-valued filter kernels are trained without falling into phase ambiguities.

2) *Complex Batch Normalization:* Batch normalization is essential to accelerate learning of deep neural networks [24]. The batch size defines the number of samples that will be used in the training process. Batch normalization normalizes each feature channels in a 4-D tensor. The convolution operators proposed in this article use real-valued weights, and we normalize the real and imaginary parts of a complex input data to zero mean. We start by centering the real and imaginary components to zero mean per input feature map as

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - E[\mathbf{x}]}{\sqrt{\sigma_k + \epsilon}} \quad (3)$$

where $E[\cdot]$ is the mean of each feature map

$$E[\mathbf{x}] = \frac{1}{MNK} \sum_{i=1}^M \sum_{j=1}^N \sum_{d=1}^K \mathbf{x}_{ijd} \quad (4)$$

σ_k is the standard deviation of each feature map in the input data tensor \mathbf{x}

$$\sigma = \frac{1}{MNK} \sum_{i=1}^M \sum_{j=1}^N \sum_{d=1}^K (\mathbf{x}_{ijd} - E[\mathbf{x}])^2 \quad (5)$$

and ϵ is the scale factor added to prevent dividing by zero. The complex batch normalization (as its real counterpart) is defined

as

$$BN(\tilde{\mathbf{x}}) = \gamma \tilde{\mathbf{x}} + \beta \quad (6)$$

where γ is a complex scaling parameter and β is a bias term in the batch normalization process. Since both γ and β are learnable parameters, we initialize γ by setting it to one and β to zero. This results in a Gaussian distribution for every feature map in the output tensor in both the real and imaginary components.

3) *Complex Activation Function:* A common nonlinear activation function employed in a neural network is the rectified linear unit (ReLU) [25]. For a real-valued neural networks, the ReLU is defined as

$$\forall (y) \in \mathbb{R} \text{ ReLU}(y) = \begin{cases} y & \text{if } y \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

We expand the real-valued ReLU to the complex data by setting the phase of the complex number strictly positive by keeping $\theta_y \in [0, \frac{\pi}{2}]$ or strictly negative by keeping $\theta_y \in [\pi, \frac{3\pi}{2}]$. Hence, the complex-valued ReLU, or ReLU_c for short, on a complex function y equals

$$\text{ReLU}_c(y) = \text{ReLU}(\Re(y)) + i\text{ReLU}(\Im(y)). \quad (8)$$

Alternatively, using the same criteria described earlier, we can derive the complex-valued sigmoid activation function. For a real-valued network, the sigmoid activation function equals

$$\text{Sigmoid}(y) = \frac{1}{1 + \exp(-y)} \quad (9)$$

and the complex-valued sigmoid function is defined as

$$\text{Sigmoid}_c = \text{Sigmoid}(\Re(y)) + i\text{Sigmoid}(\Im(y)). \quad (10)$$

4) *Loss Function:* In the training of a neural network, a loss function derives the cost of classification errors. In training a deep neural network, a softmax loss function is commonly used to derive both the categorical loss and the differentials with respect to the input data. For semantic segmentation involving C classes, a softmax loss function equals [26]

$$E_n = - \sum_{n=1}^N \mathbf{t}_n \cdot \log(\mathbf{y}_n). \quad (11)$$

Here, E is the loss function, N is the number of samples involved in the estimation, \mathbf{t}_n is a binary vector containing the target label, and \mathbf{y}_n is the class score map: binary target label vector contains values equal to 0 or 1 only. For the class representing the actual label, a value of 1 is assigned, whereas for all other classes, a value of 0 is assigned. The class score maps are obtained using a softmax function

$$y_{kij} = \frac{\exp(x_{kij})}{\sum_{c=1}^C \exp(x_{cij})} \quad (12)$$

where x_{cij} are the unnormalized feature maps at location ij . We take the maximum probability of the class scores to classify the pixel

$$y_{ij} = \underset{c}{\operatorname{argmax}} \{y_{cij}\}. \quad (13)$$

For complex-valued feature maps x_{kij} , the softmax loss cannot be used directly, because it normalizes the feature channels

to resemble class probabilities. Since the output of a sigmoid function is bounded between 0 and 1, we apply a sigmoid activation to the squared magnitude of the output feature map as input in the softmax layer. In doing so, the real-valued softmax operator (13) can be used directly to derive the loss.

B. Training the Network

1) *Complex Backpropagation*: In the training of a deep neural network, the weights and biases are iteratively optimized until the loss function (13) is at a minimum for a predefined number of epochs. We minimize the loss function by deriving the gradient of the parameters and using the stochastic gradient descent (SGD) algorithm to iteratively update their values [27]. The SGD algorithm randomly selects a mini-batch of training samples from the available data stack. By estimating the error gradient of the input filter bank and biases, we update the initial filter parameters to best extract features and predict the target output with a minimum error with respect to the ground truth.

Using the chain rule [21], the loss function is iteratively updated to reach a minimum value by adjusting the parameters as

$$w(l)[t+1] = w(l)[t] - \eta \cdot \frac{\partial z}{\partial w}(l)[t]. \quad (14)$$

Here, $\frac{\partial z}{\partial w}$ is the partial derivative of the output with respect to the weight that is computed by the network, η is the learning rate, l is the layer number in the feedforward architecture and t is the iteration number. The error gradients are consequently derived by following the chain rule. The partial derivative of the output with respect to the weight equals

$$\begin{aligned} \frac{\partial z}{\partial w}(l) = & - \left(\Re \left(\frac{\partial z}{\partial y}(l+1) \right) \cdot \Re(\tilde{\mathbf{x}}(l)) \right) \\ & + \Im \left(\frac{\partial z}{\partial y}(l+1) \right) \cdot \Im(\tilde{\mathbf{x}}(l)) \end{aligned} \quad (15)$$

where $\frac{\partial z}{\partial y}$ is the error term with respect to the previous feature map in the backward pass, and $\tilde{\mathbf{x}}$ is the feature map in the forward pass. In this way, we obtain weights with the smallest classification error.

Next we update the error term with respect to the output ($\frac{\partial z}{\partial y}$), the error term with respect to the data tensor ($\frac{\partial z}{\partial x}$) and the partial derivative of the output with respect to the weight for the weight kernel of that layer ($\frac{\partial z}{\partial w}$). These terms are completed for each layer. To do so, we focus on the derivatives of the output layer, batch normalization and convolution.

Derivative of the Output Layer: The error term with respect to the data for the output feature map from the last layer in the forward pass is derived by using the output of the softmax function as

$$\frac{\partial z}{\partial y}(l) = \frac{\partial z}{\partial y}(l+1)(y-1). \quad (16)$$

Here, $\frac{\partial z}{\partial y}(l+1)$ is the initial error term for the output initialized as a unity tensor (i.e., the dimension of this tensor is the same size as the softmax operator) and y is the output of the softmax operator.

Derivative of the Batch Normalization Layer: For the batch normalization, the derivatives of the output with respect to γ_k

and β_k equal

$$\frac{\partial z}{\partial \gamma}(l) = \sum_i^M \sum_j^N \sum_t^T \frac{\partial z}{\partial y}(l+1) \tilde{\mathbf{x}} \quad (17)$$

$$\frac{\partial z}{\partial \beta}(l) = \sum_i^M \sum_j^N \sum_t^T \frac{\partial z}{\partial y}(l+1) \quad (18)$$

respectively. The derivative of the output with respect to the feature map is influenced by $\frac{\partial z}{\partial \gamma}(l+1)$ and $\frac{\partial z}{\partial \beta}(l+1)$ as well as the batch normalization step in the forward pass, given as

$$\begin{aligned} \frac{\partial z}{\partial x}(l) = & \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} \cdot \left[\frac{\partial z}{\partial y}(l+1) - \frac{1}{MNK} \left(\frac{\partial z}{\partial \beta}(l+1) \right. \right. \\ & \left. \left. - \left(\frac{x - E[\mathbf{x}]}{\sqrt{\sigma^2 + \epsilon}} \right) \left(\frac{\partial z}{\partial \gamma}(l+1) \right) \right) \right]. \end{aligned} \quad (19)$$

By iteratively learning γ and β , the network adjusts the deviation in the data distribution to fit the zero mean Gaussian probability density function.

Derivative of the Convolution Layer: The derivative of the output with respect to the weight for each convolution layer ($\frac{\partial z}{\partial w}(l)$) is obtained from the derivatives of the previous layers with respect to the output ($\frac{\partial z}{\partial y}(l+1)$) and the current layers data tensor $\mathbf{x}(l)$ using (15).

C. Weight Initialization

Weight initialization is critical for successfully training a neural network. In a real-valued network, the weights are automatically initialized from a zero mean Gaussian distribution, which usually suits most optical datasets. Since SAR data follow a zero mean circular Gaussian probability density function, we initialize the real-valued weight kernels from a zero-mean Gaussian probability density functions with variance of ($\sigma = 10^{-2} \cdot \frac{1}{\sqrt{D}}$).

III. DATASETS

To evaluate the performance of PolSARNet, we used a fully polarized AIRSAR data acquired over Flevoland, the Netherlands and two sets of Sentinel-1 data consisting of eight and five dual polarized multitemporal Sentinel-1 images acquired in a different part of Flevoland, the Netherlands (Fig. 10).

A. AIRSAR Data

The AIRSAR sensor acquires data in C, P, and L bands for the quad polarization images. L band was used to evaluate the performance of PolSARNet (Table I). The fully polarimetric AIRSAR data was obtained from a four-look multilooked coherency matrix from the European Space Agency (ESA). The data were used in the network without further processing. The test areas contain 1024×1020 pixels for the AIRSAR data in range and azimuth directions, respectively, and cover completely agricultural areas. The polarimetric scattering properties of the scene can be observed from the false color composite of the region (Fig. 4).

We used ground truth data for the AIRSAR data obtained from the MAESTRO campaign [28]. Labels for the agricultural

TABLE I
JPL/NASA AIRSAR AIRBORNE POLARIMETRIC SAR SENSOR
ACQUISITION PARAMETERS

Parameter	Value
Sensor	AIRSAR airborne sensor
frequency/wavelength	1.26 GHz/23 cm
Polarization	Full
Resolution	6 m × 12 m
Swath width (nominal)	10 km
Off-Nadir angle (normal)	20 – 60°
Date	1989

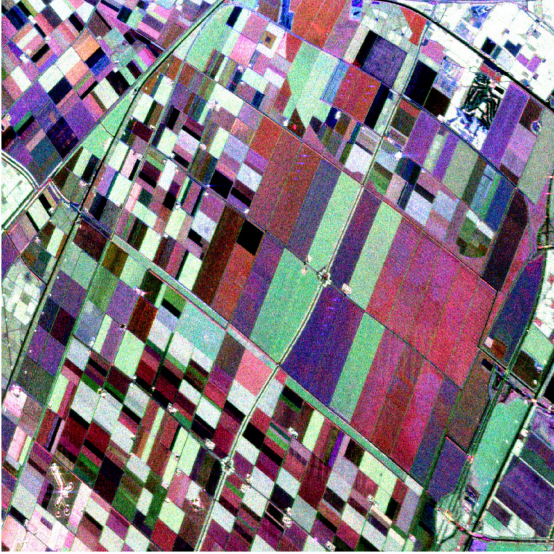
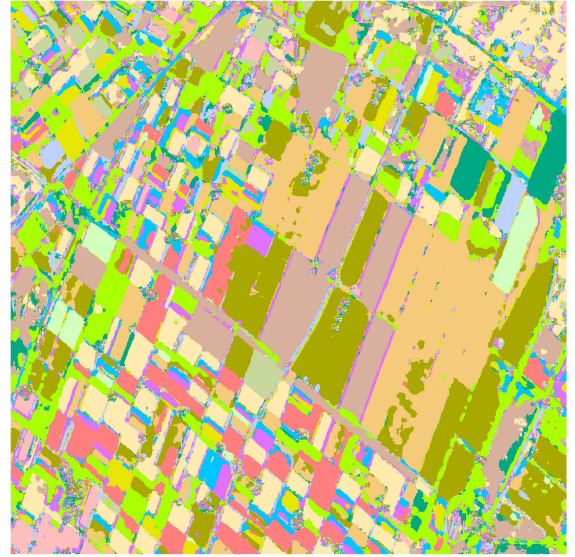


Fig. 4. False color composite of AIRSAR data. The red, green, and blue colors represent double bounce, volume, and single bounce scattering mechanisms, respectively.

land in the AIRSAR data were divided into 14 classes (see legend for Fig. 5). For testing the trained network, we prepared a separate set of testing polygons that were spatially disjoint from the polygons used in the training.

B. Sentinel-1 Data

The Sentinel-1 sensor acquires data in C band for both single and dual polarization data (Table II). The test area contains two separate areas designated as Sentinel-1 Area 1 acquired around the village of Dronten from May to October 2017 and Sentinel-1 Area 2 acquired around the village of Zeewolde from June to September 2016. Sentinel-1 Area 1 consists of eight images covering 3351×3756 pixels in range and azimuth directions, respectively. Sentinel-1 Area 2 consists of five images covering 2992×3260 pixels in range and azimuth directions, respectively. Both datasets were acquired over a completely agricultural area in the Flevoland region of the Netherlands. Even though both datasets are from Sentinel-1 sensor, the two regions have different number of images and agricultural parcel size is also different. Hence, we applied two Sentinel-1 dual-polarization multitemporal experiments (IV-B and IV-C) to show the robustness of the proposed method. The scattering coefficient change in time with respect to the VV channel for the scene can



Legend				
Potato	Beet	Wheat	Maize	Grass
Fruit	Barley	Beans	Flax	Lucerne
Oats	Onion	Peas	Rapeseed	

Fig. 5. Classification maps of PolSARNet for the AIRSAR dataset.

TABLE II
ACQUISITION PARAMETERS FOR THE SENTINEL-1 IMAGES

Parameter	Description
Polarization	VH VV
Sensor	Sentinel-1A
Resolution	3.7 m × 13.89 m
Incidence angle	39.35°
Orbit	Descending
Temporal baseline	12 days
Dates	May 05, 2017 - October 19, 2017 June 11, 2016 - September 27, 2016
Number of images	8 and 5

be observed from the false color composite image of the scene (Figs. 6 and 7).

We used ground truth data for the Sentinel-1 data obtained from the Dutch Cadastre (PDOK). These were divided into six classes and seven classes for Sentinel-1 Areas 1 and 2, respectively, with class names shown in the legend of Figs. 8 and 9. To prepare the testing polygons, we also used a spatially disjoint polygons from the training polygons. The Sentinel-1 data were obtained in a single-look complex (SLC) format. Hence, the single-look multitemporal polarimetric coherency matrix from dual polarized SAR data was obtained by following the procedure detailed in [29] and [30], respectively.

Spatial multilooking was not necessary to the Sentinel-1 data after obtaining the single-look coherency matrix. The single-look upper triangular elements of the coherency matrix in each individual dates (i.e., T_{11} , T_{12} , and T_{22}) in the dual polarized case are concatenated and fed into the network. Since PolSARNet consists of convolution layers, speckle filtering and feature extraction are done in a single run.

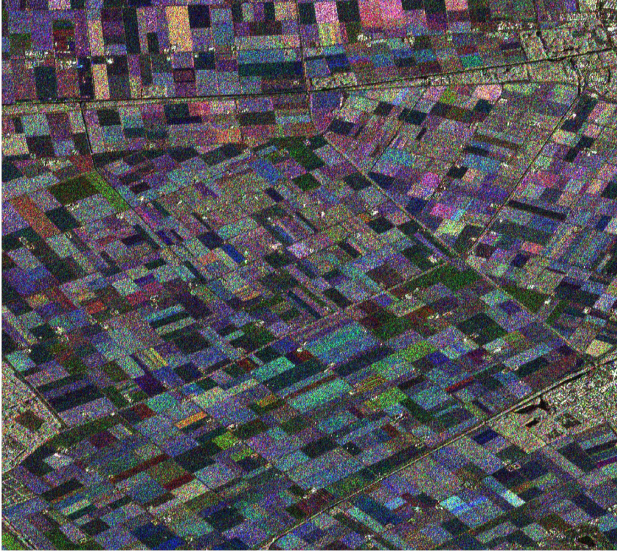


Fig. 6. False color composite of Sentinel-1 data over Area 1. The red, green, and blue colors represent the VV and VH channels on different dates. The red color represents the intensity of the VV channel on August 10, 2016. The green color represents the VH channel on July 17, 2016 and the red color represents the intensity of the VV channel on July 5, 2016.



Legend

 Barley	 Maize	 Potato
 Grass	 Onion	 Wheat

Fig. 8. Classification maps of PolSARNet for the Sentinel-1 dataset over Area 1.

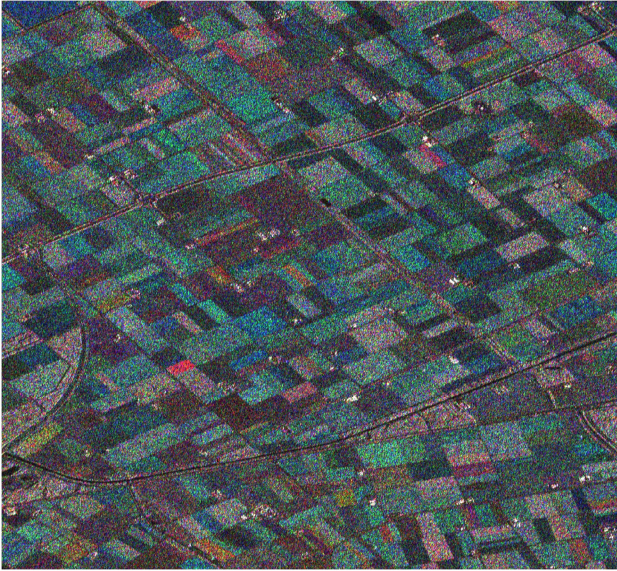


Fig. 7. False color composite of Sentinel-1 data over Area 2. The red, green, and blue colors represent the VV and VH channels on different dates. The red color represents the intensity of the VV channel on August 10, 2016. The green color represents the VH channel on November 2, 2016 and the red color represents the intensity of the VV channel on October 3, 2016.



Legend








 Barley	 Maize	 Potato	 Wheat
 Grass	 Onion	 Beet	

Fig. 9. Classification results of PolSARNet for Area 2 of the Sentinel-1 dataset.

IV. EXPERIMENTAL SETUP

A. AIRSAR Data

We prepared the training data as a 4-D tensor of dimension $13 \times 13 \times 6 \times 12000$ as input to the network. The 4-D tensor represents 12000 image patches randomly selected from the original image with spatial dimension of 13×13 pixels and six channels corresponding to the upper triangular elements of the coherency matrix. The architecture used to train the

network consisted of six layers of convolutions using a dilation factor up to $d = 3$. We also applied a stride of one and an appropriate amount of padding to maintain the same feature map size. The convolutional layers were interleaved by complex activation functions and batch normalization layers. The last two convolution layers used filter kernel dilation, whereas the initial complex convolution layer was not dilated (Table IV). All layers are initialized with random weights as described in Section II. We also adopted a multistage training and validation step to train the network.

TABLE III
LIST OF SENTINEL-1 DATA USED IN THIS ARTICLE

Sentinel-1 area 2		Sentinel-1 area 1
Number	Date	Date
1	2016/09/27	2016/09/03
2	2016/08/10	2016/08/10
3	2016/07/17	2016/11/02
4	2016/07/05	2016/10/03
5	2016/06/11	2016/06/11
6	-	2016/05/30
7	-	2016/07/05
8	-	2017/01/01

TABLE IV
POLSARNET NETWORK ARCHITECTURE WITH DILATED KERNEL
CONVOLUTIONS APPLIED ON THE AIRSAR DATA

Layer	Module type	Dimension	d	s	p
K1	Conv _c	$3 \times 3 \times 6 \times 16$	1	1	1
	ReLU _c				
K2	Conv _c	$3 \times 3 \times 16 \times 32$	1	1	1
	ReLU _c				
K3	Conv _c	$3 \times 3 \times 32 \times 32$	1	1	1
	ReLU _c				
K4	Conv _c	$3 \times 3 \times 32 \times 32$	1	1	1
	ReLU _c				
DK2	Conv _c	$3 \times 3 \times 32 \times 32$	2	1	2
	ReLU _c				
DK3	Conv _c	$3 \times 3 \times 32 \times 32$	3	1	3
	ReLU _c				
Class.	Conv _c softmax	$1 \times 1 \times 32 \times 14$	1	1	0

To evaluate the performance of PolSARNet, we compared it against a real-valued fully convolutional network (FCN_{r3}) that used only intensity images without using phase information, i.e., T11, T22, and T33 with a dimension of $13 \times 13 \times 3 \times 12000$. We also compared it with a real-valued FCN_{r9} which used all nine real and imaginary components of the coherency matrix as real channels in training the network. Hence, the real-valued tensor used in FCN_{r9} has a dimension of $13 \times 13 \times 9 \times 12000$. We further compared PolSARNet to CV-CNN, support vector machine (SVM), and Random Forest. We trained the real-valued fully convolutional method with a variable learning rate and selected the learning rate that provided the best result within 100 epochs. CV-CNN was trained using the same architecture and hyperparameters as described in [21]. We applied a Lee sigma filter [31] of size 7×7 to reduce speckle on the original image before applying SVM and Random Forest. FCN_{r3} and FCN_{r9} were trained following the PolSARNet architecture shown in Fig. 1. Both FCN_{r3} and FCN_{r9} resulted in higher classification accuracy when using learning rate = (10^{-4}) , whereas PolSARNet provided higher accuracy with a lower learning rate = (10^{-7}) . We used early stopping for PolSARNet to select the model that gives the best classification accuracy on the validation set. For training, we randomly selected 12000 labeled patches, a mini batch of 100 samples, and a weight decay factor of 5×10^{-4} . Finally, the trained network was applied to the test images. For a proper accuracy assessment, testing was performed on a separate ground test set.

TABLE V
OVERALL ACCURACY (OA), MEAN USER ACCURACY (UA), AND MEAN
PRODUCERS ACCURACY (PA) FOR THE AIRSAR DATA

	SVM	RF	FCN _{r3}	FCN _{r9}	CV-CNN	PolSARNet
OA	77.86%	74.29%	83.79%	81.17%	82.29%	87.58%
κ	0.74	0.71	0.81	0.78	0.8	0.85
UA	68.82%	70.66%	74.74%	72.55%	77.69%	79.7%
PA	64.96%	78.01%	80.12%	75.86%	81.98%	86.67%

In Table V, we used four comparison metrics: 1) overall accuracy (OA), 2) kappa coefficient (κ), 3) mean user accuracy (UA), and 4) mean producers accuracy (PA). The OA of PolSARNet (87.58%) was higher than that of FCN_{r9} (81.17%) and CV-CNN (82.29%). It achieved high classification accuracy on all classes except on beets, maize, and lucerne. Both the real and complex-valued networks achieved a similar classification accuracies with the worst classification accuracy on lucerne. Although the real and imaginary parts of the complex channels were inserted into the real network, the mentioned landcover classes were better identified and classified when using the complex-valued input channels.

PolSARNet achieved higher classification accuracy as compared to FCN_{r9} (Fig. 5). This indicates that the data are degraded by separating the real and imaginary parts for the complex elements of the coherency matrix. The OA of PolSARNet was higher than that of all compared methods. To investigate the effect of using the same label data set used in training for accuracy assessment, we took the same set of labels for testing. In this regard, PolSARNet obtained an OA of 99.27%, CV-CNN obtained 99.9%, and FCN_{r9} 99.62%, clearly showing an inflated accuracy corresponding with what has been reported previously [21].

B. Sentinel-1 Area 1

To evaluate the performance of PolSARNet on Sentinel-1 Area 1 data, we prepared the input data as a 4-D tensor of dimension $25 \times 25 \times 24 \times 12000$. The input tensor contains 24 channels, since the Coherency matrix from a single Sentinel-1 image contains three upper triangular elements corresponding to three images. The purpose of using all upper triangular elements of the coherency matrix in all eight sets of multitemporal images was to use the change in backscatter property as a discriminative feature. We used six convolutional layers with a maximum dilation factor of six (DK-6) (Table VI). The data processed using the architecture in Table IV is the four-look AIRSAR data. We used a shallower network that uses less kernel dilation to extract features. We used a shallower network because the data is a fully polarimetric data with six unique channels corresponding to the unique upper triangular elements of the coherency matrix. Since there are more channels, it is not necessary to train a deep network. Furthermore, since the data are multilooked, the speckle level is lower than a single-look data so a kernel dilation factor of three is used. On the contrary, the architecture used to process the Sentinel-1 data, as shown in Table VI, is deeper and uses a higher kernel dilation factor than that in Table IV. This is because the data have three unique channels corresponding to the coherency matrix of a dual polarized data resulting in a lower information representation. Hence, a deep network is used to extract features. Furthermore, the Sentinel-1 data used in the

TABLE VI
ARCHITECTURE OF POLSARNET WITH DILATED KERNEL CONVOLUTIONS BY A FACTOR OF SIX (DK-6) APPLIED ON THE SENTINEL-1 MULTITEMPORAL AREA 2 DATA

Layer	Module type	Dimension	d	s	p
DK1	Conv _c	$3 \times 3 \times 15 \times 16$	1	1	1
	ReLU _c				
DK2	Conv _c	$3 \times 3 \times 16 \times 32$	2	1	2
	ReLU _c				
DK3	Conv _c	$3 \times 3 \times 32 \times 32$	3	1	3
	ReLU _c				
DK4	Conv _c	$3 \times 3 \times 32 \times 32$	4	1	4
	ReLU _c				
DK5	Conv _c	$3 \times 3 \times 32 \times 32$	5	1	5
	ReLU _c				
DK6	Conv _c	$3 \times 3 \times 32 \times 32$	6	1	6
	ReLU _c				
Class.	Conv _c softmax	$1 \times 1 \times 32 \times 7$	1	1	0

manuscript are single-look data with a fully developed speckle. To suppress the noise and extract features, a kernel dilation factor of six is used.

The first layer is initialized with random weights as described in Section II without any dilation. We first trained the dilated kernel factor three (DK-3) by initializing the weights randomly. Second, we used the weights trained in the kernels with dilation factor of three as input to the kernel with dilation factor of four. In this way, we progressed to the dilation factor of six.

To evaluate the performance of PolSARNet on the Sentinel-1 data, we compared it with a real-valued FCN with the same architecture. For comparison, we created FCN_{r16} containing 16 images from the multitemporal sequence corresponding with the diagonal element of the coherency matrix. We also compared it with FCN_{r32} created by taking all the real and imaginary components from all the unique members of the coherency matrix as a separate real-valued channel. The real-valued FCN resulted in higher classification accuracy when using larger learning rates (10^{-3}), whereas PolSARNet provided higher accuracy with a lower learning rate (10^{-7}). We used early stopping in the case of PolSARNet to select the model that gives the best classification accuracy on the validation model. To apply the training, we randomly selected 12 000 labeled patches with a batch of 100 samples and a weight decay factor of 5×10^{-4} . Finally, the trained network was applied to the test images.

The complex-valued convolutions with real-valued weights improved the computational efficiency of training when compared with complex-valued convolutions with complex-valued weights of the same size by reducing the effective number of convolutions from four to two, as done in CV-CNN. This improved the computational time from 17.2 s per epoch to 14.04 s per epoch when training on input data tensor size of $13 \times 13 \times 15 \times 12\,000$ run on a Linux environment with a Xeon E5-2660 2.6 GHZ CPU processor with ten cores. This can translate to a significant improvement if hundreds of epochs are used to train the network.

The OA of PolSARNet (82.58%) was higher as compared to FCN_{r32} (80.95%) (Table VII). It achieved high classification accuracy on all classes except maize. PolSARNet achieved higher classification accuracy expressed as OA and κ (Table VII). This shows the advantage of using phase information in training deep

TABLE VII
OVERALL ACCURACY (OA), MEAN USER ACCURACY (UA), AND MEAN PRODUCERS ACCURACY (PA) FOR THE SENTINEL-1 DATA AREA 1

	SVM	RF	FCN _{r16}	FCN _{r32}	CV-CNN	PolSARNet
OA	67.81%	57.56%	78.56%	80.95%	21.9%	82.58%
κ	0.59	0.48	0.73	0.76	0.01	0.78
UA	64.85%	55.77%	76.27%	77.91%	14%	79.82%
PA	60.18%	54.67%	72.33%	77.54%	16.27%	79.12%

TABLE VIII
OVERALL ACCURACY (OA), MEAN USER ACCURACY (UA), AND MEAN PRODUCERS ACCURACY (PA) FOR THE SENTINEL-1 DATA AREA 2

	SVM	RF	FCN _{r10}	FCN _{r20}	CV-CNN	PolSARNet
OA	68.43%	65.72%	84.02%	84.23%	25.42%	87.39%
κ	0.61	0.57	0.80	0.81	0.12	0.84
UA	66.51%	61.12%	83.74%	80.85%	46.28%	84.45%
PA	61.02%	61.27%	82.08%	83.52%	24.34%	86.74%

neural networks. Although the real and imaginary parts of the complex channels were inserted into the real network in FCN_{r32}, the landcover classes were better identified and classified when using the complex-valued input channels (Fig. 8). When applying CV-CNN on a single-look multitemporal Sentinel-1 data, however, a problem of convergence was observed. Changing the hyperparameters and network depth did not improve the accuracy. The best overall accuracy of 21.9% is observed when using a learning rate of 0.7.

The use of PolSARNet in the classification of Sentinel-1 data proved effective to extract features and classify the images. Both PolSARNet and FCN_r were generally less sensitive to patch size and filter size. However, PolSARNet was more sensitive to the learning rate than the real-valued counterparts. The patch size was also important in managing the computational efficiency. Since we did not use GPU to process PolSARNet, selecting the optimal patch size by considering tradeoff between accuracy and computational efficiency was important. In this regard, a patch size of 25×25 was optimal.

C. Sentinel-1 Area 2

To further evaluate the performance of PolSARNet on Sentinel-1 Area 2, we prepared the input data as a 4-D tensor of dimension $25 \times 25 \times 15 \times 12\,000$. The training procedure followed the same step as the one applied for Sentinel-1 Area 1.

We found that the OA of PolSARNet (87.39%) was higher as compared to FCN_{r20} (84.23%) (Table VIII). Similar to Sentinel-1 Area 1, it achieved high classification accuracy on all classes except in maize (Fig. 9). Similarly, when applying CV-CNN on single-look multitemporal Sentinel-1 data, a problem of convergence was observed. A maximum overall accuracy of 25.42% was obtained when using a learning rate of 0.5. Overall, PolSARNet achieved high classification accuracy as compared to its real-valued counterparts.

V. DISCUSSION

PolSARNet proved to be effective in achieving higher classification accuracy both for the quad-polarized AIRSAR data and the dual polarization multitemporal Sentinel-1 data. The application of real-valued filter kernels on complex data reduced the effective number of convolutions from a fully complex-valued

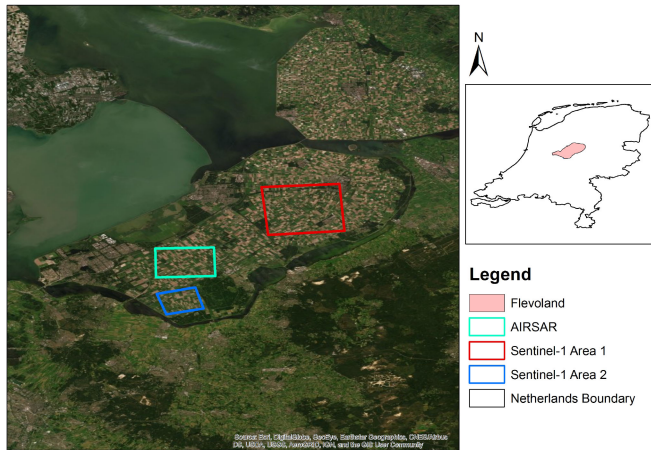


Fig. 10. Location map of the two study areas. The AIRSAR data and Sentinel-1 Area 2 are acquired over the same area.

convolutions by half; therefore, it was beneficial in reducing the computational time. The complex-valued off-diagonal elements of the coherency matrix represent the complex correlation between different Pauli channels [11]. Hence, the phase difference in the image is due to the target physical properties. This induces a spatial structure that can be clearly seen in the phase images. The application of real-valued kernel helped to scale the complex-valued data, thereby enhancing the magnitude of different features. This increased the feature identification capability and helped exploit the spatial structure that exists within the complex-valued dataset.

Extra care should be taken to select uncorrelated training and testing samples in a neural network. In [19], even though the training and testing samples are split in the experiments, the samples are not selected from a spatially disjointed region. Hence, even though the samples are different, they are selected from the same spatial region that is correlated, leading to the inflated accuracies reported in the manuscript.

The application of real-valued weight kernels on complex-valued data tensor contributed to the stability of the network since the real-valued weight kernels enhanced the amplitude of the feature maps by preserving the phase. Since the complex-valued feature maps possessed a spatial structure related to ground features, it reliably extracted features compared to CV-CNN which transformed the phases. This was readily apparent in processing the two Sentinel-1 data areas.

For the AIRSAR dataset, the data have already been multilooked four times, and hence the level of speckle found in the image was lower. This helped the network to extract the required features with high accuracy with a few convolution layers. Furthermore, one can clearly differentiate crop classes from the false color composite image, and hence both the complex and real-valued networks achieved relatively similar overall accuracies. This indicates that the phase information improved the identification capability of some classes.

In contrast, the single-look Sentinel-1 data were affected by fully developed speckle, and the respective crop types were not readily discernible from only the intensity images represented by the diagonal elements of the coherency matrix. Hence, using the phase information contained in the coherency matrix

was important. This was exemplified by the difference in the overall accuracy obtained between FCN_r and PolSARNet. Furthermore, phase information was better represented when the input channels are complex-valued as opposed to separating the real and imaginary parts of the off-diagonal elements of the coherency matrix and processing them as real channels as in FCN_r .

The selection of the filter size was done to preserve small agricultural plots and subtle features within the image scene. The 5×5 filter kernel resulted in a smooth classified map and with slightly higher classification accuracy as compared to the 3×3 filter kernel for the AIRSAR dataset. This is due to the application of filter kernel dilation. With a dilation factor equal to three, the 5×5 filter is equivalent to 11×11 , which provides long-distance spatial support to the network. However, the improved accuracy came at the cost of distortion of plot boundaries and smoothing of small agricultural plots. These effects occurred in both the real and complex-valued networks. Hence, the selection of filter sizes should be adjusted to the application at hand. With respect to the data patch size, the network did not show much sensitivity. Hence, patch sizes were selected based on computational efficiency. The most sensitive hyperparameter in training PolSARNet is the learning rate. It achieved the best results in between 10^{-8} and 10^{-7} . In contrast, the real-valued networks gave the best results in between 10^{-4} and 10^{-3} . Both FCN_r networks resulted in low classification accuracy when the learning rate was lower than 10^{-6} .

VI. CONCLUSION

We conclude that PolSARNet achieved higher classification accuracy as compared to its real-valued counterpart using the same architecture. It was able to better exploit the information contained in complex-valued images than the real-valued networks, taking the amplitude and phase separately. The potential of the PolSARNet was also revealed when processing images whose amplitude variation was less visible as opposed to the phase variation. For future work, to exploit the temporal dependencies of a multi-temporal sequence of images, we plan to use a complex-valued recurrent neural network.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [3] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. van den Hengel, "Semantic labeling of aerial and satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 2868–2881, Jul. 2016.
- [4] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," 2016, *arXiv:1606.02585*.
- [5] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.
- [6] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2017.
- [7] J. S. Lee, M. R. Grunes, and R. Kwok, "Classification of multi-look polarimetric SAR imagery based on complex Wishart distribution," *Int. J. Remote Sens.*, vol. 15, no. 11, pp. 2299–2311, 1994.

- [8] J. S. Lee, M. R. Grunes, T. L. Ainsworth, L. Du, D. L. Schuler, and S. R. Cloude, "Unsupervised classification using polarimetric decomposition and the complex Wishart classifier," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 5, pp. 2249–2258, Sep. 1999.
- [9] F. Liu, L. Jiao, B. Hou, and S. Yang, "Pol-SAR image classification based on Wishart DBN and local spatial information," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3292–3308, Jun. 2016.
- [10] L. Jiao and F. Liu, "Wishart deep stacking network for fast PolSAR image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3273–3286, Jul. 2016.
- [11] A. G. Mullissa, V. Tolpekin, and A. Stein, "Scattering property based contextual PolSAR speckle filter," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 63, pp. 78–89, 2017.
- [12] A. Freeman and S. L. Durden, "A three-component scattering model for polarimetric SAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 3, pp. 963–973, May 1998.
- [13] S. Chen, H. Wang, F. Xu, and Y. 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.
- [14] Y. Zhou, H. Wang, F. Xu, and Y. Jin, "Polarimetric SAR image classification using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1935–1939, Dec. 2016.
- [15] N. Guberman, "On complex valued convolutional neural networks," 2016, *arXiv:1602.09046*.
- [16] C. Trabelsi *et al.*, "Deep complex networks," 2017, *arXiv:1705.09792*.
- [17] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.
- [18] A. Hirose, *Complex-Valued Neural Networks: Advances and Applications*, vol. 18. Hoboken, NJ, USA: Wiley, 2013.
- [19] C. Danilla, C. Persello, V. Tolpekin, and J. R. Bergado, "Classification of multitemporal sar images using convolutional neural networks and Markov random fields," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 2231–2234.
- [20] A. G. Mullissa, C. Persello, and V. Tolpekin, "Fully convolutional networks for multi-temporal SAR image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 6635–6638.
- [21] Z. Zhang, H. Wang, F. Xu, and Y. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177–7188, Dec. 2017.
- [22] C. Persello and A. Stein, "Deep fully convolutional networks for the detection of informal settlements in VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2325–2329, Dec. 2017.
- [23] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
- [24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
- [25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, vol. 30, p. 3.
- [26] J. R. Bergado, C. Persello, and A. Stein, "Recurrent multi-resolution convolutional networks for VHR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6361–6374, Nov. 2018.
- [27] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Artif. Intell. Statist.*, 2015, pp. 562–570.
- [28] P. Basili, P. Quarto, P. Ciotti, G. D'auria, F. S. Marzano, and N. Pierdicca, "Assessment of polarimetric features to discriminate land cover from the maestro 1 campaign," *Int. J. Remote Sens.*, vol. 15, no. 14, pp. 2887–2899, 1994.
- [29] A. G. Mullissa, D. Perissin, V. A. Tolpekin, and A. Stein, "Polarimetry-based distributed scatterer processing method for PSI applications," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3371–3382, Jun. 2018.
- [30] A. G. Mullissa, V. Tolpekin, A. Stein, and D. Perissin, "Polarimetric differential SAR interferometry in an arid natural environment," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 59, pp. 9–18, 2017.
- [31] J. S. Lee, J. Wen, T. L. Ainsworth, K. Chen, and A. J. Chen, "Improved sigma filter for speckle filtering of SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 202–213, Jan. 2009.



Adugna G. Mullissa (M'18) received the M.Sc. degree in remote sensing for earth science applications from Addis Abeba University, Addis Abeba, Ethiopia, in 2008, and the Ph.D. degree in radar remote sensing from the University of Twente, Enschede, The Netherlands, in 2017.

From September 2016 to January 2017, he was a Visiting Scientist with the Lyle School of Civil Engineering, Purdue University, West Lafayette, IN, USA. From January 2018 to January 2019, he was a Researcher with the University of Twente, investigating deep learning methodologies for crop classification using polarimetric SAR data. He is currently a Researcher on near real-time deforestation monitoring using Sentinel-1 multitemporal images and machine learning with the Laboratory of Geoinformation Science and Remote Sensing, Wageningen University, The Netherlands. His research interests include microwave remote sensing, pattern recognition, and machine learning for SAR and polarimetric SAR data.



Claudio Persello (S'07–M'11–SM'17) received the Laurea (B.S.) and Laurea Specialistica (M.S.) degrees in telecommunications engineering and the Ph.D. degree in communication and information technologies from the University of Trento, Trento, Italy, in 2003, 2005, and 2010, respectively.

He is currently an Assistant Professor with the Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, The Netherlands. In 2011–2014, he was a Marie Curie Research Fellow, conducting research activity at the Max Planck Institute for Intelligent Systems and the Remote Sensing Laboratory of the University of Trento. His main research interests include analysis of remote sensing data, machine learning, image classification, pattern recognition, and unmanned aerial vehicles.

Dr. Persello is a Referee for multiple journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and *Remote Sensing and Pattern Recognition Letters*. He has served on the Scientific Committee of the Sixth International Workshop on the Analysis of Multi-temporal Remote-Sensing Images (MultiTemp 2011). His Ph.D. thesis was awarded with the prize for the best Ph.D. thesis on Pattern Recognition published between 2010 and 2012 by the GIRPR, i.e., the Italian branch of the International Association for Pattern Recognition (IAPR).



Alfred Stein received the M.Sc. degree in mathematics and information science, with a specialization in applied statistics, from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1983, and the Ph.D. degree in spatial statistics from the Wageningen University, Wageningen, The Netherlands, in 1991.

He is currently a Professor of spatial statistics and image analysis with the Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands. He is supervising eight Ph.D. students working under him. His research interests include the statistical aspects of spatial and spatiotemporal data, such as monitoring data, in the widest sense; optimal sampling; image analysis; spatial statistics; the use of prior information; issues of data quality, fuzzy techniques, and random sets in a Bayesian setting. Recently, he is involved in setting up the new scientific domain "Geohalth."

Prof. Stein is a member of the Socio-Economic and Natural Sciences of the Environment (SENSE) Research School. Since 2011, he has been the Founding Editor-in-Chief of the *Spatial Statistics* journal, which is the new leading platform in the field of spatial statistics.