Full-learning quaternion convolutional neural networks and cross-entropy-based confluence of differently represented data for PolSAR land classification

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

25-11-2021 / 06-12-2021

CITATION

Matsumoto, Yuya (2021): Full-learning quaternion convolutional neural networks and cross-entropy-based confluence of differently represented data for PolSAR land classification. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.17078351.v1>

DOI

[10.36227/techrxiv.17078351.v1](https://doi.org/10.36227/techrxiv.17078351.v1)

Full-learning quaternion convolutional neural networks and cross-entropy-based confluence of differently represented data for PolSAR land classification

Yuya Matsumoto, Ryo Natsuaki *Member, IEEE*, and Akira Hirose, *Fellow, IEEE*

Abstract—This paper proposes two methods in adaptive PolSAR land classification. The first proposal is full-learning quaternion convolutional neural networks (QCNNs). QCNNs can learn the relationship between the components of input vectors by quaternion rotations unlike conventional real-valued convolutional neural networks (RVCNNs). They have been employed in some image classification tasks including PolSAR images. However, the existing QCNNs fix their quaternion-rotation axes, so that they cannot realize arbitrary quaternion rotations. Therefore, the degree of freedom (DoF) in the fixed-axis quaternion is two while a quaternion holds DoF of four intrinsically. Our proposed full-learning QCNN is a novel neural network which updates all the four elements of respective quaternion neural weights including its convolutional kernels by backpropagation. The novel QCNN has the maximum DoF so that it takes advantage of quaternionic operation. The second proposal is a method of combining multiple-network classification results for different features. Classification results for PolSAR images largely depend on input features. Therefore, merging multiple results for different features improves the final classification results by compensating for the shortcomings of respective feature sets. In our experiments, we employ the following two features, namely, PolSAR pseudo-color features and normalized Stokes vectors. After the classification for the respective features, the classification results are merged based on the cross-entropy loss. We compare three neural networks, namely, RVCNNs, fixed-axis QCNNs and our proposed full-learning QCNNs. Our proposed QCNNs show the best classification results for respective features. We also show that the merging process highly improves the final classification results measured by F-scores.

Index Terms—Polarimetric synthetic aperture radar (PolSAR), convolutional neural network (CNN), quaternion neural network (QNN)

I. INTRODUCTION

Polarimetric synthetic aperture radar (PolSAR) systems obtain more information by using multiple polarizations than by using a single polarization [1]. Land classification is one of the most important PolSAR applications. There are various methods based on, e.g., entropy and alpha angle [2], matrix

decomposition [3]–[5], wishart distribution [6], [7], and traditional machine learning methods such as support vector machine (SVM) [8], [9] and random forest [10], [11]. Recently, neural-network methods have been actively studied and they have shown much better performance than the conventional methods [12]–[15].

There are unique difficulties for PolSAR image classification, unlike general image classification, such as selection of optimal features, less labeled data, and speckle noise [16]. Hence, some unique neural networks for PolSAR land classification have been proposed.

Quaternion neural networks (QNNs) [17] have been employed for PolSAR land classification [18]–[20]. In QNN methods, a three dimensional vector is regarded as a single quaternion. In other words, the three components of the vector are assigned to the three imaginary parts of a quaternion. QNNs restrict the linear transformation to quaternion rotation, that is, orthogonal transformation. Therefore, the input features do not degenerate. When we pay attention to a single neuron, we find that it maps them to a space of the same dimension with less parameters than three real-valued neurons. Then, the QNNs have shown higher accuracy than the conventional PolSAR classification methods.

Quaternion convolutional neural networks (QCNNs) are employed for visible RGB-color image [21] and PolSAR image [22] classification. The QCNNs have kernels consisting of quaternion neurons to learn spatial texture. In general, real-valued convolutional neural networks (RVCNNs) simply sum up the components of the input vectors in the convolutional processing, so that they ignore the relationship between the components. On the other hand, the QCNNs perform quaternionic rotation, that is, orthogonal transformation, on the respective input vectors in the convolutional processing. Hence, in the QCNNs, the components of input vectors do not degenerate, so that the QCNNs can learn the relationship between the components.

However, there are two problems in the QCNN methods. First, the existing QCNNs fix the rotation axes of their quaternion-weight transform in neurons to reduce the degree of freedom [21], [22]. Hence the QCNNs cannot represent arbitrary rotations, and lose the expression ability. Second, the only PolSAR pseudo-color features are employed as input features [22]. With only intensity features such as PolSAR pseudo-color features, it is difficult to discriminate classes

A part of this work was supported by JSPS KAKENHI under Grant 18H04105 and also by Cooperative Research Project Program of the Research Institute of Electrical Communication (RIEC), Tohoku University. The Advanced Land Observing Satellite (ALOS) original data are copyrighted by the Japan Aerospace Exploration Agency (JAXA) and provided under JAXA Fourth ALOS Research Announcement PI No. 1154 (AH).

The authors are with the Department of Electrical Engineering and Information Systems, The University of Tokyo, Tokyo 113-8656, Japan (e-mail: matsumoto@eis.t.u-tokyo.ac.jp; natsuaki@eis.t.u-tokyo.ac.jp; ahirose@ee.t.u-tokyo.ac.jp).

with low back-scattering such as sea and grass, which is to be discussed later.

This paper proposes the following two methods to address the above problems. First, we propose a full-learning QCNN which updates all four parameters of its quaternion weights of neurons, even in its convolutional kernels, by backpropagation without fixing the rotation axes. The full-learning QCNNs can perform arbitrary quaternion rotation on the input vectors. In other words, our proposed QCNNs have the maximum degree of freedom to take advantage of quaternionic rotations unlike the existing fixed-axis QCNN. We propose the detailed dynamics in this paper.

Second, we propose a method of combining two classification results for different features. It is very effective to merge classification results for different features since different features are suitable for representing diverse nature. In our experiments, we employ two sets of features, namely, PolSAR pseudo color images and Stokes vectors normalized by their total power. The former are derived by assigning the intensities of scattering coefficients to red, green, and blue components based on the Pauli decomposition. These intensity features include fine textures of a PolSAR image. By learning the spatial textures, neural networks can distinguish between classes such as town and forest, which have so similar scattering characteristics in pixels that they are often misclassified each other. Another set of features, the Stokes vectors normalized by their total power, have the degree of polarization (DoP) representing the state of a partially polarized wave. Most scattered waves are partially polarized, and the DoP is significant information for land classification. It is very effective to learn the DoP in discriminating between classes with low backscattering such as sea and grass, on which we focus in this paper. The cross-entropy suggests which set of features is more effective in the pixel-wise classification result.

In the experiments in this paper, we compare three neural networks, namely, RVCNNs, fixed-axis QCNNs and our proposed QCNN. As experimental results, our proposed QCNNs show the best classification performance by learning the relationship between the components of input vectors with arbitrary quaternion rotations. The neural networks learning the PolSAR pseudo-color images succeed in discriminating town and forest, while the neural networks learning the normalized Stokes vectors succeed in sea and grass discrimination. Merging the two classification results for different features compensates for the shortcomings of respective features to improve the final classification results measured by F-scores.

This paper is organized as follows. Section II presents the theory of the full-learning QCNN. Section III describes the details of the experiments. The experimental results are explained in Section IV. Section V concludes this paper.

II. QUATERNION CONVOLUTIONAL NEURAL NETWORKS

A. Quaternion arithmetic

This section introduces the detailed theory of a full-learning quaternion convolutional neural network. First of all, we

describe the expressions of various operations of quaternions. A quaternion is expressed as

$$\mathbf{x} = x^e + x^i \mathbf{i} + x^j \mathbf{j} + x^k \mathbf{k} \quad (1)$$

$$= (x^e, x^i, x^j, x^k) \quad (2)$$

$$= \begin{bmatrix} x^e \\ x^i \\ x^j \\ x^k \end{bmatrix} \quad (3)$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ stand for the three imaginary units of a quaternion. They satisfy the following Hamilton rules:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1 \quad (4)$$

$$\mathbf{i}\mathbf{j} = -\mathbf{j}\mathbf{i} = \mathbf{k}, \mathbf{j}\mathbf{k} = -\mathbf{k}\mathbf{j} = \mathbf{i}, \mathbf{k}\mathbf{i} = -\mathbf{i}\mathbf{k} = \mathbf{j} \quad (5)$$

The operators between quaternions $\mathbf{p} = (p^e, p^i, p^j, p^k) = (p^e, \mathbf{v}_p)$ and $\mathbf{q} = (q^e, q^i, q^j, q^k) = (q^e, \mathbf{v}_q)$ are defined as

Addition and subtraction:

$$\mathbf{p} \pm \mathbf{q} = (p^e \pm q^e, \mathbf{v}_p \pm \mathbf{v}_q) \quad (6)$$

Outer product:

$$\mathbf{p} \otimes \mathbf{q} = (p^e q^e - \mathbf{v}_p \cdot \mathbf{v}_q, p^e \mathbf{v}_q + q^e \mathbf{v}_p + \mathbf{v}_p \times \mathbf{v}_q) \quad (7)$$

Hadamard product:

$$\mathbf{p} \odot \mathbf{q} = (p^e q^e, p^i q^i, p^j q^j, p^k q^k) \quad (8)$$

Inner product:

$$\mathbf{p} \cdot \mathbf{q} = p^e q^e + p^i q^i + p^j q^j + p^k q^k \quad (9)$$

Conjugate:

$$\mathbf{p}^* = (p^e, -p^i, -p^j, -p^k) \quad (10)$$

Norm:

$$|\mathbf{p}| = \sqrt{(p^e)^2 + (p^i)^2 + (p^j)^2 + (p^k)^2} \quad (11)$$

Scalar:

$$\lambda \mathbf{p} = (\lambda p^e, \lambda p^i, \lambda p^j, \lambda p^k) \quad (12)$$

B. Construction of the QCNN

The whole structure of the QCNN is shown in Fig. 1. The basic architecture is the same as that of the conventional convolutional neural networks (CNNs). The QCNN has L layers. The 0-th layer is the input-terminal layer. The last L -th layer is the output layer. All the signals, weights and neurons are in the quaternion domain except for the real-valued L -th layer, which is explained later. The output values of the network are compared with the output teacher signals for learning, which is explained below. After the completion of the learning, classification is conducted.

In Fig. 1, c is the channel index for the l -th layer's input $\mathbf{X}^{(l)}$ in the convolutional layer, k is the index for kernels, (p, q) , (a, b) and (m, n) are positions in the kernels, the output of the convolutional layer and the output of the pooling layer, respectively, for each channel.

Fig. 1: The whole architecture of the QCNN.

C. Forward processing

1) *Convolutional layer*: The inputs and outputs of the l -th layer are denoted $\mathbf{X}^{(l-1)} = [\mathbf{x}_{a+p,b+q,c}^{(l-1)}]$ and $\mathbf{V}_k^{(l)} = [\mathbf{v}_{a,b,k}^{(l)}]$, respectively, and the connection weights of convolutional kernels are represented as $\mathbf{W}^{(l)} = [\mathbf{w}_{p,q,c,k}^{(l)}]$, whose height and width are denoted $P_{conv}^{(l)}$ and $Q_{conv}^{(l)}$. The quaternion convolution is defined as

$$\mathbf{V}^{(l)} = \mathbf{f}(\mathbf{U}^{(l)}) \quad (13)$$

$$\mathbf{u}_{a,b,k}^{(l)} = \sum_{p,q,c} \frac{\mathbf{w}_{p,q,c,k}^{(l)} \otimes \mathbf{x}_{a+p,b+q,c}^{(l-1)} \otimes \mathbf{w}_{p,q,c,k}^{*(l)}}{|\mathbf{w}_{p,q,c,k}^{(l)}|} + \mathbf{b}_k^{(l)} \quad (14)$$

where $\mathbf{U}^{(l)} = [\mathbf{u}_{a,b,k}^{(l)}]$ and $\mathbf{b}_k^{(l)}$ are the neural inner state and bias, respectively. The function $\mathbf{f}$ are an activation function defined as

$$\mathbf{f}(\mathbf{p}) = f(p^i)\mathbf{i} + f(p^j)\mathbf{j} + f(p^k)\mathbf{k} \quad (15)$$

where $\mathbf{p}$ is a purely imaginary quaternion and f is rectified linear unit (ReLU) in our experiments.

2) *Pooling layer*: In our experiments, the pooling processing averages the inputs as

$$\mathbf{x}_{m,n,k}^{(l)} = \frac{1}{P_{pool}^{(l)} \cdot Q_{pool}^{(l)}} \sum_{(a,b) \in R_{m,n}} \mathbf{v}_{a,b,k}^{(l)} \quad (16)$$

where $P_{pool}^{(l)}$ and $Q_{pool}^{(l)}$ are the height and width of the pooling window $R_{m,n}$ to calculate $\mathbf{x}_{m,n,k}^{(l)}$, respectively. After the quaternion signals pass through convolutional layers and pooling layers, they are flattened and fed into quaternion fully-connected layers, whose processing follows Ref. [18].

3) *Connection between a quaternion fully-connected layer and a real-valued output layer*: The output layer is a real-valued fully-connected layer directly following the last quaternion hidden layer. Before the quaternion signals are fed into the real-valued output layer, they are flattened to real-valued signals as

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \end{bmatrix} \rightarrow \begin{bmatrix} x_1^i \\ x_1^j \\ x_1^k \\ x_2^i \\ x_2^j \\ x_2^k \\ \vdots \end{bmatrix} \quad (17)$$

where $\mathbf{x}_d$ ($d = 1, 2, \dots$) are purely imaginary quaternions whose imaginary components are x_d^i , x_d^j , and x_d^k .

Finally, the output of the QCNN goes through a softmax function, and the loss E is calculated with the output teacher signal $\mathbf{t}$, which is a one-hot vector. In our experiments, the loss E is the following cross entropy:

$$E = \sum_{j=1}^J -t_j \log y_j \quad (18)$$

where t_j denotes the j -th component of the teacher signal $\mathbf{t}$ and y_j presents the output of the j -th neuron of the output layer.

D. Backward propagation in learning

1) *Connection between a quaternion fully-connected layer and a real-valued output layer*: Before real-valued gradients of the loss E in terms of the real-valued signals are passed

to the previous quaternion fully-connected layer, they are converted to quaternion gradients as

$$\begin{bmatrix} \partial E / \partial x_1^i \\ \partial E / \partial x_1^j \\ \partial E / \partial x_1^k \\ \partial E / \partial x_2^i \\ \partial E / \partial x_2^j \\ \partial E / \partial x_2^k \\ \vdots \end{bmatrix} \rightarrow \begin{bmatrix} \partial E / \partial x_1 \\ \partial E / \partial x_2 \\ \vdots \end{bmatrix} \quad (19)$$

where $\partial E / \partial x_d^i$, $\partial E / \partial x_d^j$ and $\partial E / \partial x_d^k$ ($d = 1, 2, \dots$) are converted to a single quaternion $\partial E / \partial x_d$. After this conversion, the backpropagation dynamics in quaternion fully-connected layers follows Ref. [18].

2) *Pooling layer*: In the l -th pooling layer, $\partial E / \partial \mathbf{x}_{a,b,k}^{(l)}$ propagates from the $(l+1)$ -th layer. There are no trainable parameters in the pooling layer. However, we have to transfer the gradient given by the next layer as

$$\frac{\partial E}{\partial \mathbf{v}_{a,b,k}^{(l)}} = \frac{1}{P_{\text{pool}}^{(l)} \cdot Q_{\text{pool}}^{(l)}} \frac{\partial E}{\partial \mathbf{x}_{m,n,k}^{(l)}} \quad ((a,b) \in R_{m,n}) \quad (20)$$

3) *Convolutional layer*: Based on the chain rule in Ref. [17], the gradient of a loss function E in terms of l -th-layer-kernel parameters is calculated as

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{w}_{p,q,c,k}^{(l)}} = & - \sum_{a,b} \frac{1}{|\mathbf{w}_{p,q,c,k}^{(l)}|} \\ & \cdot \left\{ \frac{\partial E / \partial \mathbf{u}_{a,b,k}^{(l)} \cdot (\mathbf{w}_{p,q,c,k}^{(l)} \otimes \mathbf{x}_{a+p,b+q,c}^{(l-1)} \otimes \mathbf{w}_{p,q,c,k}^{*(l)})}{|\mathbf{w}_{p,q,c,k}^{(l)}|^2} \mathbf{w}_{p,q,c,k}^{(l)} \right. \\ & \left. - 2 \partial E / \partial \mathbf{u}_{a,b,k}^{(l)} \otimes \mathbf{w}_{p,q,c,k}^{(l)} \otimes \mathbf{x}_{a+p,b+q,c}^{*(l-1)} \right\} \end{aligned} \quad (21)$$

and

$$\frac{\partial E}{\partial \mathbf{b}_k^{(l)}} = \sum_{a,b} \frac{\partial E}{\partial \mathbf{u}_{a,b,k}^{(l)}} \quad (22)$$

where

$$\frac{\partial E}{\partial \mathbf{u}_{a,b,k}^{(l)}} = \mathbf{f}'(\mathbf{u}_{a,b,k}) \odot \frac{\partial E}{\partial \mathbf{v}_{a,b,k}^{(l)}} \quad (23)$$

The gradient which propagates to the previous layer is calculated as (24), where $S_{m,n}$ is a set of points (a,b) connected to $\mathbf{x}_{m,n,k}^{(l-1)}$.

III. EXPERIMENTS

A. The datasets and the areas where we obtained training data and test data

PolSAR images in our experiments are a $4,000 \times 5,000$ -pixel image of Ishikari Bay and a $8,000 \times 3,000$ -pixel image of Tomakomai, Hokkaido, Japan obtained by Japan Aerospace Exploration Agency (JAXA) Advanced Land Observing Satellite (ALOS)-2, observed in August 2015 (offnadir angle: 25.4° , ascending, level 1.1 (no averaging)). The RGB visualization of these two PolSAR images is shown in Fig. 2. We classify their pixels into four classes (water, town, grass and forest).

Fig. 3 shows the corresponding optical images obtained by Sentinel-2 in September 2017. They also present the areas where we obtained training data (unframed) and test data (framed) are also shown in Fig. 3. We place a data window with a height and a width of both 15 pixels. We pick up 600-window training data for each class (total 2,400) from the unframed areas in the Ishikari image in Fig. 3(a). The 2,400 test data (600 for each class) are obtained from the framed areas in each of the two PolSAR images so that we obtain 4,800 test data in total.

B. The features used in the experiments

The experimental features are derived from the complex scattering matrix expressed as

$$\mathbf{S} = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \quad (25)$$

We obtain two sets of features, namely, the PolSAR pseudo-color features and normalized Stokes vectors, from the above scattering matrix. In our experiments, we train three types of neural networks, namely, RVCNNs, fixed-axis QCNNs and our proposed QCNNs with these two sets of features.

1) *The PolSAR pseudo-color image*: The scattering matrix can be decomposed under the Pauli decomposition as

$$a = \frac{S_{HH} + S_{VV}}{\sqrt{2}}, b = \frac{S_{HH} - S_{VV}}{\sqrt{2}}, c = \frac{S_{HV} + S_{VH}}{\sqrt{2}} \quad (26)$$

The PolSAR pseudo-color features are constructed by assigning $|b|^2$, $|c|^2$ and $|a|^2$ to red, green and blue components, respectively [23]. They are employed in some methods as the classification features [22], [24], [25]. Intensity features such as PolSAR pseudo-color features have so fine texture of land surface that neural networks are able to extract the spatial texture by learning them. SAR images are often multi-look processed to mitigate speckle noise. However, the fine textures of a SAR image are sacrificed by the averaging processing. We do not perform the multi-look processing on the PolSAR images to extract finer textures.

When the PolSAR pseudo-color features are fed into the networks, they should be normalized to a certain range. In general, some pixels of a PolSAR image have very large amplitude values compared to others. If we try to normalize the feature values in such a manner that their maximum value becomes unity, almost all the other amplitude values become zero. Therefore, we normalize them in another way that their 90 percentile values in an image becomes unity and that, after the normalization, the amplitude value larger than unity is set as unity.

The architectures of the neural networks learning the PolSAR pseudo-color features are shown in Fig. 4. All the neural networks consist of one convolutional layer, one pooling layer, one hidden fully-connected layer and the output layer. The pooling layer conducts average pooling to mitigate speckle noise. Since in the sliding window classification, the positional relationship on the feature map is not important for classification, the pooling layer averages the entire feature map.

$$\frac{\partial E}{\partial \mathbf{x}_{m,n,c}^{(l-1)}} = \sum_k \sum_{(a,b) \in S_{m,n}} \frac{\mathbf{w}_{P_{conv}^{(l)}+m-a-1, Q_{conv}^{(l)}+n-b-1, c, k}^{*(l)} \otimes (\partial E / \partial \mathbf{u}_{a,b,k}^{(l)}) \otimes \mathbf{w}_{P_{conv}^{(l)}+m-a-1, Q_{conv}^{(l)}+n-b-1, c, k}^{(l)}}{|\mathbf{w}_{P_{conv}^{(l)}+m-a-1, Q_{conv}^{(l)}+n-b-1, c, k}^{(l)}|} \quad (24)$$

Fig. 2: The amplitude of scattering-matrix components of (a)Ishikari and (b)Tomakomai PolSAR images with $4,000 \times 5,000$ and $8,000 \times 3,000$ pixels, respectively. Note that the 90 percentile value of each components of scattering matrix in this image is set as the upper limits for color assignment, respectively, (R: $|S_{HH}|$, G: $|S_{HV}|$, B: $|S_{VV}|$).

2) *Stokes vectors normalized by their total power:* Here, we explain how to derive the Stokes vectors normalized by their total power. First, we consider six incidence polarizations $\mathbf{E}^i = [E_H^i, E_V^i]^T$ (T : transpose), namely, horizontal polarization, vertical polarization, $+45^\circ$ linear polarization, -45° linear polarization, left-handed circular polarization and right-handed circular polarization in our experiments. Then, the received waves $\mathbf{E}^r = [E_H^r, E_V^r]^T$ are expressed as

$$\begin{bmatrix} E_H^r \\ E_V^r \end{bmatrix} = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \begin{bmatrix} E_H^i \\ E_V^i \end{bmatrix} \quad (27)$$

Then, the corresponding Stokes vectors $\mathbf{g}$ are obtained as

$$\mathbf{g} = \begin{bmatrix} \langle g_0 \rangle \\ \langle g_1 \rangle \\ \langle g_2 \rangle \\ \langle g_3 \rangle \end{bmatrix} = \begin{bmatrix} \langle E_H^r E_H^{r*} + E_V^r E_V^{r*} \rangle \\ \langle E_V^r E_H^{r*} - E_H^r E_V^{r*} \rangle \\ \langle E_H^r E_V^{r*} - E_V^r E_H^{r*} \rangle \\ \langle j(E_H^r E_V^{r*} - E_V^r E_H^{r*}) \rangle \end{bmatrix} \quad (28)$$

where $\langle \cdot \rangle$ denotes spatial averaging process. Finally, we normalize the Stokes vectors by their total power $\langle g_0 \rangle$ as

$$\mathbf{P} = \begin{bmatrix} \langle g_1 \rangle / \langle g_0 \rangle \\ \langle g_2 \rangle / \langle g_0 \rangle \\ \langle g_3 \rangle / \langle g_0 \rangle \end{bmatrix} \quad (29)$$

The normalized Stokes vectors have the Degree of Polarization (DoP) information in their norm as

$$DoP = \sqrt{\frac{\langle g_1 \rangle^2 + \langle g_2 \rangle^2 + \langle g_3 \rangle^2}{\langle g_0 \rangle^2}} \quad (30)$$

In general, scattered waves are partially polarized, and it is very important to learn the DoP information, which is totally different from intensity.

The architectures of the neural networks learning the normalized Stokes vectors are shown in Fig. 5. We determined them empirically.

C. Learning and classification processing

The neural networks are trained with a minibatch having 40 data. One epoch has $2,400/40 = 60$ times weight updates. We train the neural networks learning the PolSAR pseudo-color features for 1,000 epochs and those learning the normalized Stokes vectors for 200 epochs. The loss function is calculated as the cross-entropy loss in (18). The weight updating is conducted by stochastic gradient descent with momentum whose learning rate is 0.01 and momentum parameter is 0.9 in our experiments.

In classification processing, a 15×15 -pixel-window image cropped from the PolSAR image is fed into the neural networks one-by-one. Then, the loss is calculated by the outputs and the output teacher signals for each class, and the class having minimum loss is the class which the pixel belongs to. This process is conducted until all the pixels are classified.

Fig. 3: (a)Ishikari Bay and (b)Tomakomai observed by Sentinel-2 optical images including the areas from which we obtained training data (unframed) and test data (framed.)

Fig. 4: The architectures of the three neural networks learning the PolSAR pseudo-color, namely, (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN. The symbols in the left side of this figure correspond to parameters in Fig. 1.

Fig. 5: The architectures of the three neural networks learning the Stokes vectors normalized by their total power, namely, (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN. The symbols in the left side of this figure correspond to parameters in Fig. 1.

D. Combining classification results obtained for the two sets of features

We explain a method of combining two classification results obtained for the two different sets of features here. First, a neural network trained with each set of features classifies both of the PolSAR images. In the classification processing, we obtain the losses in (18) per pixel. Then, the two losses based

on two different features are compared pixel-to-pixel, and the class with the smaller loss is regarded as the final result. In other words, the class of the merged result c_m is obtained as

$$c_m = \begin{cases} c_1 & (E_1 < E_2) \\ c_2 & (E_1 > E_2) \end{cases} \quad (31)$$

where c_i and E_i ($i = 1, 2$) are the classification result and loss based on feature i , respectively.

IV. RESULTS

A. The classification results for the PolSAR pseudo-color features

Fig. 6 shows the classification results of the PolSAR pseudo-color features obtained for Ishikari scene. It is found that the sea area in the upper left is significantly misclassified into grass. Since the backscattering in the sea and grass is very small, the neural networks learning intensity features such as PolSAR pseudo-color features cannot differentiate between them. In addition, there are some misclassifications between forest and town in the lower left of the results. In general, town and forest are misclassified each other because of their similar scattering characteristics.

Fig. 7 shows the classification results of Tomakomai by training data obtained from the Ishikari image. The town area in the upper right is misclassified into forest while the forest area in the lower left into town. The buildings in this town area have such a large angle to the direction of the radar beam that the cross-polarization components of scattering in this town area, namely, S_{HV} and S_{VH} , become relatively large like forest. Therefore, it is very difficult to discriminate between town and forest generally. Fig. 8 shows the enlarged views of this town area. Our proposed QCNN classifies it more accurately than the RVCNN and fixed-axis QCNN because it can learn the relationship between the components of input vectors by arbitrary quaternion rotations.

Table I presents the confusion matrices and F-scores of respective neural networks measured by test data obtained from both of the two PolSAR images. From the confusion matrices, it can be seen that the water and grass, and the town and forest are misclassified each other, respectively. Note that, for the PolSAR pseudo-color features, we find much misclassification between the water and the grass. The F-scores show that our proposed QCNN especially succeeded in discriminating between town and forest compared to the other two networks. Our proposed QCNN shows the best average F-score.

B. The classification results for the normalized Stokes vectors

Fig. 9 shows the classification results of Ishikari for the normalized Stokes vectors. The sea area in the upper left of the classification results is accurately classified compared to that of the results for the PolSAR pseudo-color features. Scattering waves at water surface have low coherence due to ripples while possible wind-swaying grass is invisible by an L-band electromagnetic wave. Therefore, the DoP values for these two areas become different. The neural networks are able to differentiate between the sea and grass by learning partially polarized state from the normalized Stokes vectors.

Fig. 10 shows the classification results of Tomakomai for the normalized Stokes vectors. The town area in the upper right is significantly misclassified into forest. The town and forest have so similar scattering characteristics in pixels that they need to be discriminated by spatial texture. The fine textures

of PolSAR images are included mainly in intensity features. However, the Stokes vectors are normalized by their intensity, so that the textures are lost, coupled with the impact of multi-look processing. Therefore, the neural networks cannot learn spatial textures, and fail to discriminate between town and forest.

Table II presents the confusion matrices and F-scores of respective neural networks. Compared to Table I for the PolSAR pseudo-color features, all the neural networks succeeded in discriminating between water and grass while they fail to distinguish between town and forest. From these two experiments for the two different features, it is found that the classification results largely depend on learned features.

C. Our proposed merging results

The previous results show that the PolSAR pseudo-color features are suitable for discrimination between town and forest, but not between sea and grass, while the normalized Stokes vectors are opposite. Fig. 11 shows the merged results of Ishikari using the two features. The sea area seems to be accurately classified. Fig. 12 presents the classification results of sea area in Ishikari shown so far for respective neural networks and features. Compared to the classification results for the PolSAR pseudo-color features, the sea area quality of merged results is largely improved. In addition, in Fig. 13, the forest area in the lower left is more accurately classified compared to the classification results for each feature.

The merging results of Tomakomai area are shown in Fig. 14. The forest area in the lower left and the town area in the upper right are accurately classified. Fig. 15 shows the classification results of the forest area in Tomakomai presented so far for respective neural networks and features. The forest area quality of the merged results is improved compared to the result for each feature respectively. Fig. 16 summarizes the classification results of the town area in the upper right in Tomakomai. The merged results of this town area are worse than the results for the PolSAR pseudo-color features while they are better than the those for the normalized Stokes vectors.

Table III presents the confusion matrices and F-scores of respective neural networks. Compared to Tables I and II, the overall classification results are improved by the merging method. Table IV summarizes the average F-scores of respective neural networks and features presented so far. The results of our proposed QCNN shows the best classification performance in all the situations. In addition, the merging method significantly improved the classification results, demonstrating its effectiveness.

V. CONCLUSION

We proposed full-learning quaternion convolutional neural networks which learn data by updating all four parameters of a quaternion. The proposed QCNNs showed better classification performance than the conventional RVCNNs and fixed-axis QCNNs. This is because the full-learning QCNN can learn the relationship between the components of input vectors with arbitrary quaternion rotations. In addition, we showed

TABLE I: The classification results of RVCNN, fixed-axis QCNN and QCNN for PolSAR pseudo-color features.

Neural network	Class	water	town	grass	forest	F-score	avg F-score
RVCNN	water	1009	0	191	0	0.846	0.866
	town	0	1078	5	117	0.939	
	grass	177	1	890	132	0.779	
	forest	0	16	0	1184	0.899	
fixed-axis QCNN	water	943	0	257	0	0.826	0.845
	town	0	1020	12	168	0.903	
	grass	139	3	931	127	0.775	
	forest	0	35	3	1162	0.875	
QCNN	water	994	0	206	0	0.849	0.874
	town	0	1117	9	74	0.952	
	grass	148	1	919	132	0.787	
	forest	0	28	0	1172	0.909	

TABLE II: The classification results of RVCNN, fixed-axis QCNN and QCNN for normalized Stokes vectors.

Neural network	Class	water	town	grass	forest	F-score	avg F-score
RVCNN	water	1192	2	6	0	0.975	0.884
	town	0	942	46	212	0.804	
	grass	53	177	939	31	0.856	
	forest	0	21	2	1177	0.898	
fixed-axis QCNN	water	1184	7	9	0	0.977	0.881
	town	0	979	49	172	0.808	
	grass	39	210	905	46	0.835	
	forest	0	27	4	1169	0.904	
QCNN	water	1185	10	5	0	0.964	0.890
	town	0	955	66	179	0.821	
	grass	74	149	965	12	0.859	
	forest	0	13	12	1175	0.916	

TABLE III: The merged results of RVCNN, fixed-axis QCNN and QCNN.

Neural network	Class	water	town	grass	forest	F-score	avg F-score
RVCNN	water	1196	0	4	0	0.970	0.921
	town	0	1052	9	139	0.925	
	grass	69	11	982	138	0.895	
	forest	0	11	0	1189	0.892	
fixed-axis QCNN	water	1176	0	24	0	0.964	0.916
	town	0	1044	8	148	0.919	
	grass	63	17	988	132	0.889	
	forest	0	10	2	1188	0.891	
QCNN	water	1196	0	4	0	0.970	0.928
	town	0	1096	14	90	0.946	
	grass	69	13	980	138	0.890	
	forest	0	9	4	1187	0.908	

TABLE IV: The summary of the average F-scores for each neural network and each feature.

Feature	RVCNN	fixed-axis QCNN	QCNN
PolSAR pseudo-color	0.865	0.845	0.874
Normalized Stokes vectors	0.883	0.881	0.890
Merge	0.921	0.916	0.928

that the classification results largely depend on input features. Specifically, the PolSAR pseudo-color features are suitable for discrimination between town and forest, but not between water

and grass, while the normalized Stokes vectors are opposite. We merged the classification results for respective features based on their cross-entropy values to highly improve the

final classification by compensating for the shortcomings of respective features successfully.

REFERENCES

- [1] J.-S. Lee, M. Grunes, and E. Pottier, "Quantitative comparison of classification capability: fully polarimetric versus dual and single-polarization SAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 11, pp. 2343–2351, 2001.
- [2] S. R. Cloude and E. Pottier, "An entropy based classification scheme for land applications of polarimetric sar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 1, pp. 68–78, 1997.
- [3] A. Freeman and S. L. Durden, "A three-component scattering model for polarimetric SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 3, pp. 963–973, 1998.
- [4] Y. Yamaguchi, T. Moriyama, M. Ishido, and H. Yamada, "Four-component scattering model for polarimetric SAR image decomposition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 8, pp. 1699–1706, Aug 2005.
- [5] Y. Yamaguchi, A. Sato, W.-M. Boerner, R. Sato, and H. Yamada, "Four-component scattering power decomposition with rotation of coherency matrix," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2251–2258, 2011.
- [6] J. S. LEE, M. R. GRUNES, and R. KWOK, "Classification of multi-look polarimetric SAR imagery based on complex Wishart distribution," *International Journal of Remote Sensing*, vol. 15, no. 11, pp. 2299–2311, 1994. [Online]. Available: <https://doi.org/10.1080/01431169408954244>
- [7] Jong-Sen Lee and Grunes, M.R. and Ainsworth, T.L. and Li-Jen Du and Schuler, D.L. and Cloude, S.R., "Unsupervised classification using polarimetric decomposition and the complex Wishart classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2249–2258, 1999.
- [8] C. Lardeux, P.-L. Frison, C. Tison, J.-C. Souyris, B. Stoll, B. Fruneau, and J.-P. Rudant, "Support Vector Machine for Multifrequency SAR Polarimetric Data Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 12, pp. 4143–4152, 2009.
- [9] X. Niu and Y. Ban, "Multi-temporal RADARSAT-2 polarimetric SAR data for urban land-cover classification using an object-based support vector machine and a rule-based approach," *International Journal of Remote Sensing*, vol. 34, no. 1, pp. 1–26, 2013. [Online]. Available: <https://doi.org/10.1080/01431161.2012.700133>
- [10] L. Loosvelt, J. Peters, H. Skriver, B. De Baets, and N. E. C. Verhoest, "Impact of Reducing Polarimetric SAR Input on the Uncertainty of Crop Classifications Based on the Random Forests Algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 4185–4200, 2012.
- [11] P. Du, A. Samat, B. Waske, S. Liu, and Z. Li, "Random Forest and Rotation Forest for fully polarized SAR image classification using polarimetric and spatial features," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 38–53, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271615000611>
- [12] Y. Zhou, H. Wang, F. Xu, and Y. Jin, "Polarimetric sar image classification using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1935–1939, 2016.
- [13] Z. Zhang, H. Wang, F. Xu, and Y. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7177–7188, 2017.
- [14] S. De, L. Bruzzone, A. Bhattacharya, F. Bovolo, and S. Chaudhuri, "A novel technique based on deep learning and a synthetic target database for classification of urban areas in PolSAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 1, pp. 154–170, 2018.
- [15] Chen, Si-Wei and Tao, Chen-Song, "PolSAR image classification using polarimetric-feature-driven deep convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 4, pp. 627–631, 2018.
- [16] H. Parikh, S. Patel, and V. Patel, "Classification of SAR and PolSAR images using deep learning: a review," *International Journal of Image and Data Fusion*, vol. 11, no. 1, pp. 1–32, 2020. [Online]. Available: <https://doi.org/10.1080/19479832.2019.1655489>
- [17] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, "Quaternion neural network with geometrical operators," *Journal of Intelligent & Fuzzy Systems*, vol. 15, no. 3, 4, pp. 149–164, 2004.
- [18] F. Shang and A. Hirose, "Quaternion Neural-Network-Based PolSAR Land Classification in Poincare-Sphere-Parameter Space," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 9, pp. 5693–5703, 2014.
- [19] H. Kim and A. Hirose, "Unsupervised fine land classification using quaternion autoencoder-based polarization feature extraction and self-organizing mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 3, pp. 1839–1851, 2017.
- [20] K. Kinugawa, F. Shang, N. Usami, and A. Hirose, "Isotropization of quaternion-neural-network-based polsar adaptive land classification in poincare-sphere parameter space," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 8, pp. 1234–1238, 2018.
- [21] Zhu, Xuanyu and Xu, Yi and Xu, Hongteng and Chen, Changjian, "Quaternion Convolutional Neural Networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [22] Zhang, Xinzhen and Xia, Jili and Tan, Xiaoheng and Zhou, Xichuan and Wang, Tao, "PolSAR Image Classification via Learned Superpixels and QCNN Integrating Color Features," *Remote Sensing*, vol. 11, no. 15, 2019. [Online]. Available: <https://www.mdpi.com/2072-4292/11/15/1831>
- [23] S. Cloude and E. Pottier, "A review of target decomposition theorems in radar polarimetry," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 498–518, 1996.
- [24] S. Uhlmann and S. Kiranyaz, "Integrating Color Features in Polarimetric SAR Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2197–2216, 2014.
- [25] J. Cheng, Y. Ji, and H. Liu, "Segmentation-Based PolSAR Image Classification Using Visual Features: RHLBP and Color Features," *Remote Sensing*, vol. 7, no. 5, pp. 6079–6106, 2015. [Online]. Available: <https://www.mdpi.com/2072-4292/7/5/6079>

Fig. 6: Classification results of Ishikari scene for PolSAR pseudo-color features by (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image for reference.

Fig. 7: Classification results of Ishikari scene for PolSAR pseudo-color features by (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image for reference.

Fig. 8: Classification results of Ishikari town area for PolSAR pseudo-color features by (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image for reference.

Fig. 9: Classification results of Ishikari scene for normalized Stokes vectors by (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image.

Fig. 10: Classification results of Tomakomai scene for normalized Stokes vectors by (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image.

Fig. 11: Merged classification results of Ishikari scene by (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image.

Fig. 12: The classification results of sea area in Ishikari by respective network and features as well as the corresponding area of a Sentinel-2 optical image.

Fig. 13: The classification results of forest area in Ishikari by respective networks and features as well as the corresponding area of a Sentinel-2 optical image.

Fig. 14: Merged classification results of Tomakomai scene of (a)RVCNN, (b)fixed-axis QCNN and (c)QCNN as well as (d)Sentinel-2 optical image.

Fig. 15: The classification results of forest area in Tomakomai by respective networks and features as well as the corresponding area of a Sentinel-2 optical image.

Fig. 16: The classification results of town area in Tomakomai by respective networks and features as well as the corresponding area of a Sentinel-2 optical image.