Color image classification via quaternion principal component analysis network

Rui Zeng, Jiasong Wu, Zhuhong Shao, Yang Chen, Beijing Chen, Lotfi Senhadji, Huazhong Shu

Color Image Classification via Quaternion Principal Component Analysis Network

Rui Zeng[1, 3], Jiasong Wu[1, 2, 3], Zhuhong Shao[4], Yang Chen[1, 2, 3], Beijing Chen[5], Lotfi Senhadji [2, 3], Huazhong Shu[1, 2, 3]

[1]*LIST, the Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 210096, China*

[2]*INSERM, U 1099, Université de Rennes 1, LTSI, Rennes 35000, France*

[3]*Centre de Recherche en Information Biomédicale Sino-français, Nanjing 210096, China*

[4]*College of Information Engineering, Capital Normal University, Beijing 100048, China*

[5]*China-USA Computer Science Research Center, Nanjing University of Information Science and Technology, 210044 Nanjing, China.*

Abstract: The principal component analysis network (PCANet), which is one of the recently proposed deep learning architectures, achieves the state-of-the-art classification accuracy in various datasets and reveals a simple baseline for deep learning networks. However, the performance of PCANet may be degraded when dealing with color images due to the fact that the architecture of PCANet cannot properly utilize the spatial relationship between each color channel in three dimensional color image. In this paper, a quaternion principal component analysis network (QPCANet), which extends PCANet by using quaternion theory, is proposed for color image classification. Comparing with PCANet, the proposed QPCANet takes into account the spatial distribution information of RGB channels in color images and ensures larger amount of intra-class invariance by using quaternion domain representation for color images. Experiments conducted on different color image datasets such as UC Merced Land Use, Georgia Tech face, CURet and Caltech-101 have revealed that the proposed QPCANet generally achieves higher classification accuracy than PCANet in color image classification task. The experimental results also verify that QPCANet has much better rotation invariance than PCANet when color image dataset contains lots of rotation information and demonstrate even a simple one layer

QPCANet may obtain satisfied accuracy when compared with two layer PCANet.

Keywords: Deep learning, convolutional neural network, quaternion, QPCANet, PCANet, color image classification

1. Introduction

In recent years, image classification becomes a more and more active research topic in the field of pattern recognition and computer vision. It is a very challenging task since image content recognition is subject to various changes of image in terms of illumination, rotation, scaling or more complex deformation. To effectively counter the intra-class variability of image content, numerous approaches have been proposed in the past decades. These methods can be generally divided into two groups: manually extracting intrinsic feature [1-3] and learning unsupervised/supervised feature from data of interest [4-5].

For the first group, the most commonly used methods are scale-invariant feature transforms (SIFT) [1], Gabor features [2], and local binary pattern (LBP) [3]. Both of them achieve great success in different kinds of image classification tasks, such as object recognition, texture classification, and face recognition. However, the limitation of this sort of methods is obviously that we need to manually select the proper method when dealing with new image classification task, where usually requires new domain knowledge.

To remedy the limitation of above hand-crafted methods, the second group, which is well known as deep learning approaches [4-9], has become a very active research field in recent years. The objective of deep learning [7] is to extract intrinsic features from data by a multi-level architecture, with the hope that higher-level features represent more concise semantics of the data. Many approaches have been used to build deep learning architectures, which can be roughly classified into four classes by Bengio et al. [8]: (1) deep learning architectures based on probabilistic models, for example, deep belief network (DBN) [4] and deep boltzmann machines

(DBM) [10], etc. (2) deep learning architectures based on reconstruction algorithms, for example, deep autoencoder [5] and sparse coding [11], etc. (3) deep learning architectures based on manifold learning algorithms, for example, local coordinate coding (LCC) algorithm [12, 13], etc. (4) deep learning architectures based on multilayer neural networks, in which convolutional neural network (CNN) [14-24], first proposed by LeCun et al. [14, 15], gradually becomes the mainstream deep learning structure for image classification after that AlexNet [16] won the champion of ImageNet [17] large scale visual recognition challenge of 2012. Then, Overfeat [18], VGGNet [19], GoogleNet [20], and ResNet [21] were further proposed and obtained much lower error rates than AlexNet [16] in the ImageNet challenge in the year from 2013 to 2015. Generally speaking, multilayer CNN [16, 18-21] with many parameters is very suitable for the classification of large scale image datasets, for example, ImageNet database [17]. However, when the size of image datasets is not very large, two or three-layer convolutional networks with few parameters may be enough for a good classification performance. For example, Mallat and Bruna [25, 26] proposed a mathematical justified wavelet scattering network (ScatNet) whose convolutional layer, nonlinear layer, pooling layer are constructed by prefixed complex wavelets, modulus operator, and average operator, respectively. More recently, Chan et al. [27] proposed a new deep learning architecture named principal components analysis network (PCANet) whose convolutional layer, nonlinear layer, pooling layer are constructed by principal components analysis (PCA) filters, binary hashing, and block-wise histograms, respectively. Although PCANet uses the most basic and simple operations, it is quite on par with and often better than the state-of-the-art techniques of feature selection for most image classification tasks, including face images, hand-written digits, texture images, and object images. Meanwhile, Chan et al. [27] proposed the linear discriminant analysis network (LDANet) as a variant of PCANet. The construction method of PCANet attracts the attentions of many researchers. Gan et al. [28] proposed a deep graph embedding network (GENet) for face recognition and Feng et al. [29] presented a discriminative locality alignment network (DLANet) for scene classification. Jia et al.

[30] proposed two-dimensional PCANet (2DPCANet) for dayside aurora classification. Qin et al. [31] combined PCANet and spatial pyramid pooling for underwater live fish recognition. Zhao et al. [32] proposed multi-level modified finite radon transform network (MMFRTN) for image upsampling and Lei et al. [33] proposed stacked image descriptor for face recognition. Li et al. [34] proposed SAE-PCA network for human gesture recognition in RGBD (Red, Green, Blue, Depth) images. Zeng et al. [35] proposed a multilinear discriminant analysis network (MLDANet) for tensor object classification.

On the other hand, color image processing also attracts the attention of many researchers since color images can provide much larger amount of information of the real-world objects when compared with gray images [36-40]. The classical representation of color images usually combines the values of Red (R), Green (G), Blue (B) - channels into one vector. Under this simple representation, the relationships between each color channel pixel of color image are destroyed, and the dimension of images is three times of that of gray-scale images. As a result, it is crucial to seek a way to represent color images more properly by taking the spatial relationships between R, G, B channels into consideration. Quaternion [41, 42] is shown as powerful mathematical tool for representing color images [43-50] and also rotation operation [51, 52]. In this way, a color image can be represented as a compact quaternion, which preserves intrinsic structure of color image and the spatial relationships between R, G, B channels. Bihan and Sangwine [43] as well as Pei et al. [44] proposed a new quaternion feature extraction method for color image, called quaternion principal components analysis (QPCA), which is able to extract more robust and informative features from color image than classical PCA. These studies emphasize that QPCA is superior to PCA in color image representation.

At the present stage, most of the above feature extraction algorithms (SIFT, LBP, PCANet, LDANet, etc.) were only designed for gray image classification and they will face many issues when simply be applied to color image classification, such as degrading performance, dimension curse and so on [53]. Thus, similar researches that are suitable for color image classification need

to be studied.

In this paper, to address the above issues, we propose a new color image feature extraction algorithm, namely, quaternion principal component analysis network (QPCANet) which is a quaternion deep learning architecture and extends the principals of PCANet approach from real domain to quaternion domain. The input of QPCANet utilize the quaternion expression of color image, where the values of R, G, B, channels are put into three imaginary parts of quaternion, respectively. In the processing stage of QPCANet, we first make use of quaternion convolutional layer with QPCA filter to extract higher level semantic quaternion feature of color image. We then use quaternion binary hashing operation to construct the nonlinear layer. For the last stage, i.e. pooling layer, the block histogram operation for four components of quaternion feature, which is extracted from the above layer, are processed to generate the feature vector for classification. The performance of QPCANet feature on various classifiers, for example, k-nearest neighbor (KNN), large scale linear support vector machine (large scale SVM) [54-56], are also studied in detail in this paper. Moreover, QPCANet are then evaluated and compared with PCANet on various color image databases for many classification tasks, including face recognition, object recognition, texture classification and land use classification.

The contributions of the paper are as follows. First, we propose a novel feature extraction algorithm, namely, QPCANet, which shows better performance on color image classification than that of RGB PCANet [27], RGB LDANet [27], RGB SIFT [36], and RGB LBP [3]; Second, we show that, despite its simplicity and generality of quaternion architecture, one layer QPCANet achieve better result than two layer PCANet. Last but not least, to our best knowledge, we first extend simple deep learning network to quaternion domain. Crucially, this work gives a totally new perspective of deep learning neural networks on quaternion domain. We empirically show that even a low-level hierarchy convolutional networks in quaternion domain may outperform high-level convolutional networks in real domain.

The rest of the paper is organized as follows. The quaternion algebra and the QPCA are briefly

introduced in Section 2. Section 3 describes the architecture of QPCANet in details and also shows how it works. Section 4 introduces similar methods proposed by existing literature and their parameter setting in experiments. The parameter model of QPCANet is also discussed in this section. Classification performances of QPCANet and PCANet are evaluated and compared on various color image datasets in Section 5. Section 6 concludes the paper.

2. Quaternion algebra and QPCA

In this section, we briefly review the quaternion algebra and QPCA.

2.1 Quaternion algebra

A quaternion number x is a hypercomplex number, which consists of one real part and three imaginary parts:

$$x = S(x) + I(x)\boldsymbol{i} + J(x)\boldsymbol{j} + K(x)\boldsymbol{k} \in \mathbb{Q} \tag{1}$$

where $\mathbb{Q}$ denotes the quaternion number field, $S(x), I(x), J(x), K(x) \in \mathbb{R}$ ($\mathbb{R}$ denotes the real number field) and $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are three imaginary units obeying the following rules:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}, \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j} \tag{2}$$

, which shows that the quaternion multiplication is non-commutative. We call x a pure quaternion when $S(x) = 0$.

The conjugate and l_2 norm of a quaternion x is defined as, respectively,

$$x^* = S(x) - I(x)\boldsymbol{i} - J(x)\boldsymbol{j} - K(x)\boldsymbol{k} \tag{3}$$

$$\|x\| = \sqrt{S^2(x) + I^2(x) + J^2(x) + K^2(x)} \tag{4}$$

If $\|x\| = 1$, we call x a unit quaternion. For a complete review of the properties of quaternion, please refer to [42].

2.2 QPCA

A color image can be represented as a pure quaternion matrix $\mathbf{Q} \in \mathbb{Q}^{m \times n}$ whose elements are as

follows [45]:

$$\mathbf{Q}(s,t) = \mathbf{Q}_R(s,t)\boldsymbol{i} + \mathbf{Q}_G(s,t)\boldsymbol{j} + \mathbf{Q}_B(s,t)\boldsymbol{k}, \quad 1 \le s \le m, 1 \le t \le n \tag{5}$$

Where $\mathbf{Q}_R(s,t)$, $\mathbf{Q}_G(s,t)$, and $\mathbf{Q}_B(s,t)$ are represented as red, green, and blue components of the pixel at position (s, t), respectively.

Suppose we have a set of quaternion images $\{\mathbf{Q}_i \in \mathbb{Q}^{m\times n}, i = 1,..., N\}$. We vectorize all quaternion images and denote them as $\{\bar{\mathbf{Q}}_i \in \mathbb{Q}^{mn}, i = 1,..., N\}$, which are then concatenated to obtain

$$\bar{\mathbf{Q}} = [\bar{\mathbf{Q}}_1, \bar{\mathbf{Q}}_2, \ldots, \bar{\mathbf{Q}}_N] \in \mathbb{Q}^{mn\times N} \tag{6}$$

Then, the covariance matrix is given by

$$\mathbf{C} = \frac{1}{N}\bar{\mathbf{Q}}'\bar{\mathbf{Q}}'^{H} \tag{7}$$

where $\bar{\mathbf{Q}}' = \bar{\mathbf{Q}} - E[\bar{\mathbf{Q}}]$, $E[\bar{\mathbf{Q}}]$ is a matrix whose columns are centered from $\bar{\mathbf{Q}}$ and the superscript H is the conjugate transposition operator. Then, the QPCA of color images is given by:

$$\mathbf{C} = \mathbf{W}\boldsymbol{\Omega}\mathbf{W}^{H} \tag{8}$$

Note that $\mathbf{C}$ is a quaternionic Hermitian matrix, $\boldsymbol{\Omega}$ is a real diagonal matrix [43] and each column of $\mathbf{W} \in \mathbb{Q}^{mn\times mn}$ is an eigenvector of $\mathbf{C}$ according to quaternion eigenvalue decomposition [44]. Then quaternion feature of color images can be written as:

$$\mathbf{Q}_f = \mathbf{W}^{H}\bar{\mathbf{Q}} \tag{9}$$

Each column of $\mathbf{Q}_f$ is a quaternion feature corresponds to each color image.

One may note that QPCA is invariant to spatial rotation of color image shown in [43]. However, if you concatenate the values of R, G, B channels, and then use the ordinary PCA, the process is not invariant to spatial rotation of color images. That is why QPCA is superior to the ordinary PCA when dealing with color images.

3. Quaternion principal component analysis network (QPCANet)

The architecture of the proposed QPCANet is depicted in Fig. 1. In this section, we analyze the structure of two-layer QPCANet (QPCANet-2) for color image classification, and we show how to build a multi-layer QPCANet.

3.1 The first QPCA layer

Suppose that we have N quaternion images $\{\mathbf{Q}_i \in \mathbb{Q}^{m\times n}, i = 1,...,N\}$ and their corresponding labels for training. For simplicity, we assume that the patch size is $k_1 \times k_2$, k_1 and k_2 are positive odd integers. We collect all $(m-k_1+1) \times (n-k_2+1)$ quaternion patches around each pixel of the ith quaternion image $\mathbf{Q}_i$. Then each quaternion patch is centered by subtracting its mean. Thus we get zero-mean quaternion patches for the ith quaternion image. We note these quaternion features $\mathbf{q}_i = [\mathbf{q}_{i,1}, \mathbf{q}_{i,2}, \ldots, \mathbf{q}_{i,(m-k_1+1)\times(n-k_2+1)}]$, and each column of $\mathbf{q}_i$ belongs to $\mathbb{Q}^{k_1k_2}$. Repeating the above process, we can get all quaternion patches of N input patterns for training. By constructing the same matrix for all quaternion patches and by putting them together, we obtain

$$\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N] \in \mathbb{Q}^{k_1k_2 \times N(m-k_1+1)(n-k_2+1)} \tag{10}$$

The covariance matrix of $\boldsymbol{q}$ is computed as:

$$\mathbf{C}^1 = \frac{\mathbf{q}\mathbf{q}^H}{N(m-k_1+1)(n-k_2+1)} \tag{11}$$

The above matrix admits a quaternion eigenvalue decomposition:

$$\mathbf{C}^1 = \bar{\mathbf{W}}^1 \mathbf{\Omega} \bar{\mathbf{W}}^{1H} \tag{12}$$

where $\bar{\mathbf{W}}^1 \in \mathbb{Q}^{k_1k_2 \times k_1k_2}$ is a unitary matrix that contains the eigenvectors of $\mathbf{C}^1$ and $\mathbf{\Omega} \in \mathbb{R}^{k_1k_2 \times k_1k_2}$ is a diagonal matrix with eigenvalues on its diagonal. The values on the diagonal of $\mathbf{\Omega}$ are arranged in decreasing magnitude order and the corresponding eigenvectors, i.e. the principal components of $\mathbf{q}$, are arranged accordingly in $\bar{\mathbf{W}}^1$. The bigger the eigenvalue is, the more important the quaternion principal component will be. These principal component vectors are also called QPCA filter bank.

Let L_i be the desired number of QPCA filters in the i th layer. We choose then the first L_1 eigenvectors of $\bar{\mathbf{W}}^1$ to form a new matrix $\mathbf{W}^1 \in \mathbb{Q}^{k_1 k_2 \times L}$ where each of its columns is seen as a filter $\mathbf{W}_l^1$:

$$\mathbf{W}_l^1 \in \mathbb{Q}^{k_1 \times k_2}, l = 1, 2, \ldots, L \tag{13}$$

These filters form the filter bank of quaternion image $\{\mathbf{Q}_i, i = 1, ..., N\}$. The filter bank captures the main variations of all quaternion patches.

Then the first QPCA layer executes a convolution using the QPCA filter bank $\mathbf{W}^1$ to get a set of quaternion feature maps $\{\mathbf{F}_i^l, l = 1, ..., L_1\}$ of $\mathbf{Q}_i$ as follows:

$$\mathbf{F}_i^l = \mathbf{Q}_i * \mathbf{W}_l^1, i = 1, 2, \ldots, N, l = 1, 2, \ldots, L_1 \tag{14}$$

where $\{\mathbf{F}_i^l, l = 1, ..., L_1\}$ is the lth feature map of $\mathbf{Q}_i$ .The boundary of $\mathbf{Q}_i$ is padded by zero quaternion (i.e. zero-padding), whose real part and three imaginary parts are zeros, so as to ensure that $\mathbf{F}_i^l$ and $\mathbf{Q}_i$ have the size $m \times n$. $\mathbf{Q}_i$ can then be transformed into several quaternion feature maps $\{\mathbf{F}_i^l, l = 1, ..., L_1\}$ according to the numbers of QPCA filters l . It turns out that each element of $\{\mathbf{F}_i^l, l = 1, ..., L_1\}$ can also be used as a quaternion input pattern. By repeating the above process, one can expect to derive high-level features.

3.2 The second QPCA layer

Repeating the same process as the first stage, we collect all the quaternion patches which have the same size as the first layer from all the quaternion feature maps of all images. We denote these quaternion patch as a patch set $\{\boldsymbol{r}_i^{l,a}, l = 1, ..., L_1; a = 1, ..., (m - k_1 + 1)(n - k_2 + 1)\}$, where $\boldsymbol{r}_i^{l,a}$ is the ath quaternion patch of $\mathbf{F}_i^l$. The set $\{\boldsymbol{r}_i^{l,a}, l = 1, ..., L_1; a = 1, ..., (m - k_1 + 1)(n - k_2 + 1)\}$ are then sent to QPCA algorithm to get the filter bank $\bar{\mathbf{W}}^2$ in the second layer.

For the filter bank $\bar{\mathbf{W}}^2$, we only pick the first L_2 column to form a new filter bank

$\{\mathbf{W}_\ell^2, \ell = 1,...,L_2\}$, which are then convoluted by $\mathbf{F}_i^l$ to obtain $L_1 \times L_2$ quaternion maps $\mathbf{G}_i^{11},\ldots,\mathbf{G}_i^{1L_2},\mathbf{G}_i^{21},\ldots,\mathbf{G}_i^{2L_2},\mathbf{G}_i^{L_1 1},\ldots,\mathbf{G}_i^{L_1 L_2} \in \mathbb{Q}^{m\times n}$ as follows:

$$\mathbf{G}_i^{l,\ell} = \{\mathbf{F}_i^l * \mathbf{W}_\ell^2, l = 1,...,L_1; \ell = 1,...,L_2\} \tag{15}$$

One or more additional stages can be stacked as above. But in our experiments, we found that QPCANet-2 is enough to get a good accuracy in color image classification and thus a deep architecture is not necessarily required.

3.3 Quaternion feature maps coding

Each quaternion feature maps extracted by the second QPCA layer should be binarized, weighted, and then summed in order to reduce the complexity of quaternion feature maps and make the extracted feature more separable. Thus, each quaternion feature map $\mathbf{G}_i^{l,\ell} = S(\mathbf{G}_i^{l,\ell}) + I(\mathbf{G}_i^{l,\ell})\boldsymbol{i} + J(\mathbf{G}_i^{l,\ell})\boldsymbol{j} + K(\mathbf{G}_i^{l,\ell})\boldsymbol{k}$, is binarized by applying the Heaviside step function $H(\cdot)$ to its four parts (the value of $H(\cdot)$ is one for positive entries and zero otherwise). The binarized quaternion feature map is denoted by $\tilde{\mathbf{G}}_i^{l,\ell} = S(\tilde{\mathbf{G}}_i^{l,\ell}) + I(\tilde{\mathbf{G}}_i^{l,\ell})\boldsymbol{i} + J(\tilde{\mathbf{G}}_i^{l,\ell})\boldsymbol{j} + K(\tilde{\mathbf{G}}_i^{l,\ell})\boldsymbol{k}$, and the resulting maps are weighted to form a new quaternion pattern:

$$\mathbf{T}_i^l = \sum_{\ell=1}^{L_2} 2^{\ell-1}\tilde{\mathbf{G}}_i^{l,\ell} = S\left(\mathbf{T}_i^l\right) + I\left(\mathbf{T}_i^l\right)\boldsymbol{i} + J\left(\mathbf{T}_i^l\right)\boldsymbol{j} + K\left(\mathbf{T}_i^l\right)\boldsymbol{k}. \tag{16}$$

Note that the pixel values of $S\left(\mathbf{T}_i^l\right), I\left(\mathbf{T}_i^l\right), J\left(\mathbf{T}_i^l\right), K\left(\mathbf{T}_i^l\right)$ are integers belonging to the interval $\left[0, 2^{L_2} - 1\right]$.

Next, we discuss how to code these binarized quaternion feature maps $\mathbf{T}_i^l$. Since $\mathbf{T}_i^l$ has four parts $S\left(\mathbf{T}_i^l\right), I\left(\mathbf{T}_i^l\right), J\left(\mathbf{T}_i^l\right), K\left(\mathbf{T}_i^l\right)$, So we need coding each part of $\mathbf{T}_i^l$ separately. Let us first consider $S(\mathbf{T}_i^l)$ for example. We divide $S(\mathbf{T}_i^l)$ into B blocks. For each block, we simply compute the histogram (with $2^{L_2} - 1$ bins) of the decimal values in each block. After this coding process, we concatenate the histograms of the B blocks into one vector, that is,

$Bhist(S(\mathbf{T}_i^l)) \in \mathbb{R}^{2^{L_2}B}$. After performing the same coding operation on another three parts $I\left(\mathbf{T}_i^l\right), J\left(\mathbf{T}_i^l\right), K\left(\mathbf{T}_i^l\right)$, we obtain four vectors, which are then concatenated to derive the feature vector of the quaternion image $\mathbf{Q}_i$, which can be written as follows:

$$\mathbf{f}_i = \left[Bhist\left(S\left(\mathbf{T}_i^l\right)\right), Bhist\left(I\left(\mathbf{T}_i^l\right)\right), Bhist\left(J\left(\mathbf{T}_i^l\right)\right), Bhist\left(K\left(\mathbf{T}_i^l\right)\right)\right] \in \mathbb{R}^{4\times(2^{L_2})L_1B} \tag{12}$$

Since the feature of quaternion image is computed by histogram from a large amount of blocks which picked from several feature maps, the feature vector of $\mathbf{Q}_i$ is sparse and very long. Thus we feed these feature vectors to classifies, for example, KNN, SVM [54-56], for classification. Chan et al. [27] showed that large scale linear SVM [54] can obtain higher accuracy than other classifiers when dealing with the sparse and long feature vector, and they also suggested that, when using PCANet, non-overlapping blocks are suitable for face images, whereas the overlapping blocks are appropriate for hand-written digits, textures, and object images. We found that the recommendation is also suitable for QPCANet.

4. Experimental Setting and Parameters model selecting

4.1 Introduction of the experimental datasets and the preprocessing

In this subsection, we introduce the datasets and the preprocessing used in our experiments.

UC Merced Land Use dataset. UC Merced land use dataset [57] is manually extracted from large images from the United States Geological Survey (USGS) National Map Urban Area Imagery collection for various urban areas around the country. The dataset, composed of 256 × 256 pixels RGB images with pixel resolution of one foot, is manually partitioned into 21 classes. Each class contains 100 images. Fig. 2 shows 21 images, each of which represents a distinct class. Some representative images from the dataset are also shown in Fig. 3, from which we can see that the dataset contains abundant rotation information due to a different point of view of aerial photography. Thus, the land objects appear in one class are mainly the same, the only difference is the angles of view. Obviously, one may see that there is a large intra-class variability, which

will increase the difficulty of color image classification. Every images are scaled to 32 × 32 pixels to reduce the computational cost in our experiment.

Georgia Tech face database. Georgia Tech face database [58] contains images of 50 people and for each individual, 15 color images were collected at the Center for Signal and Image Processing at Georgia Institute of Technology. All images in the database are 640 × 480 pixels and the average size of faces in these images is 150 × 150 pixels. Most of the images were taken in two different sessions to take into account the variations in illumination conditions, facial expression, and appearance. In our experiment, the face images are cropped with the dimension 128 × 160 to guarantee classification accuracy. Some examples are shown in Fig. 4. To reduce the computational complexity, we resize all cropped images to 64 × 64 pixels.

CURet texture database. The CURet texture dataset [59] contains images of 61 materials that broadly span the range of different surfaces that we commonly see in our environment. Every class is composed of 92 images of the size of 200 × 200 pixels. All images are linearly scaled to 32 × 32 pixels for experimental simplicity.

Caltech-101 object dataset. Caltech-101 dataset [60] contains a total of 9146 images including color and gray-level images, belonging to 101 distinct objects, including faces, watches, ants, pianos, etc. In our experiment, we only consider 8733 color images in Caltech-101 dataset and discard all gray-level images. All color images were resized to be 32 × 32 pixels in order to reduce the computational time.

4.2 Introduction of our compared methods

In this paper, similar with PCANet, we only compare the results of single descriptor but not compare the methods that are composed of many different descriptors. Specifically, we compare our proposed QPCANet method with original gray PCANet [27], and its two variants, that is, RGB PCANet [27], and RGB LDANet [27], and also with some widely used feature extractors, including RGB SIFT [36] and RGB LBP [3]. The compared methods are described as follows.

With respect to RGB SIFT, we first perform conventional SIFT [1] to R, G, B channels

separately to obtain three feature vectors which are then concentrated together to get one feature vector. RGB SIFT shows better performance than conventional SIFT when dealing with color images. The conventional SIFT descriptor is extracted from a 16×16 block and the grid spacing is set to 6. Following the method proposed by Yang et al. [61], we computer sparse coding for each SIFT feature vector of all images and then histogram by three pyramid levels. For RGB LBP, conventional LBP are separately performed with respect to R, G, B channels to obtain three vectors, which are then combined to form one feature vector. The number of filter size in LBP is set to 8 and the radius is set to 1. When dealing with RGB PCANet, we use the same method as [27] to preprocess color image, i.e. we gather three color channels together to form a long vector representation of color image and then regard it as a gray image to process it. Because RGB LDANet has the same architecture as RGB PCANet, so we directly use the same parameter model and preprocessing method. The postfixes -1 -2 appeared in later sections denote the number of layers in network architecture (e.g. RGB LDANet-1, RGB LDANet-2).

4.3. The impact of classifiers and various parameters

In this subsection, we focus on the problem of classifier selection and parameter models setting for QPCANet by using UC Merced land use dataset.

The impact of classifiers. We first discuss which classifier is more suitable for the feature extracted by the proposed QPCANet. We test three kinds of classifier, including KNN with Euclidean distance, KNN with cosine similarity, and large scale linear SVM. The parameters of two-layer QPCANet (QPCANet-2) and two-layer RGB PCANet (RGB PCANet-2) are set as follows: the patch size, the block size, the overlapping ratio, and the number of filters are 3×3, 8×8, 0.6, and $L_1 = L_2 = 8$, respectively. The recognition accuracy of QPCANet-2 and RGB PCANet by using three classifiers is shown in Table 1. One can see that the performance of RGB PCANet-2 is slightly better than that of QPCANet-2 by using KNN classifier, however, the performance of both QPCANet-2 and RGB PCANet-2 is not very good in KNN classifier in

general.

The reason of experimental results is that KNN with cosine similarity outperform that with Euclidean distance in high dimensional vector space, especially for sparse entries [62, 63]. One may note that a feature vector generated by QPCANet is a combination of histograms computed statistically from blocks, thereby possessing lots of sparse entries. Considering Euclidean distance of two vectors are computed from the difference of each entry within both vectors, this sort of measure cannot be avoided to assign same weight to every entry of vector. The sparse entries in a high dimensional vector should be assigned smaller weight because they do not contain abundant information as much as other entries possessed normal value. Thus, the capacity of classification will decay when apply to high dimensional feature vector. On the contrary, cosine similarity pays more attention to the estimation of difference between two angles of two feature vectors, it means that cosine similarity typically focuses on principal components of feature vector to discriminate two vectors, rather than trivial entries.

The performance of QPNCANet-2 by using large scale linear SVM is much better than that of RGB PCANet. Therefore, a proper classifier significantly promotes the performance of QPCANet-2. Why large scale linear SVM is more suitable for QPCANet-2 feature and RGB PCANet-2 feature than KNN? Because the length of RGB PCANet-2 feature is three times of that of gray PCANet-2 and the length of QPCANet-2 feature is four times of that of gray PCANet-2. Both QPCANet-2 and RGB PCANet-2 will obtain a very long but very sparse feature vector from color image, which lead to very worse performance by using KNN classifier but good performance by using large scale linear SVM. Therefore, we use large scale linear SVM as classifier in the following experiments.

The impact of the number of filters L_i. We then discuss the effect of the number of filters L_i on the recognition accuracy of QPCANet, whose patch size, block size, and overlapping ratio are set to 3×3, 8×8, and 0.6, respectively. Then we use a greedy algorithm to find the optimal number of QPCA filters in every stages of QPCANet. Fig. 5 shows the result of the recognition accuracy of

one-staged QPCANet (QPCANet-1) in relation with the number of filters L_1, whose values vary from 2 to 9. One can see that QPCANet-1 achieves the best result in $L_1 = 9$. But we still choose $L_1 = 8$ when taking the octave filters into consideration as that of gray PCANet [27]. We then draw in Fig. 4 the result of the recognition accuracy of two-staged QPCANet (QPCANet-2) in relation with the number of filters L_2, whose values vary from 2 to 9, meanwhile, L_1 is set to 8. It can be seen from Fig. 4 that QPCANet-2 achieves a good performance for $L_2 = 8$, however, when $L_2 \geq 8$, the recognition rate begins decreasing. Considering the simplicity of applications of QPCANet, we should make the parameter model as simple as possible. Thus, although some fine-tuned L_1 and L_2 values could lead to performance improvement, we decided to set $L_1 = L_2 = 8$.

The impact of the patch size and the block size. We then study the effect of the patch size and the block size on the recognition accuracy of QPCANet, whose number of filters is set to $L_1 = L_2 = 8$ and overlapping ratio is set to 0.6, respectively. Similar to gray PCANet, the patch size is set to 3×3, 5×5, 7×7. We set the block size to 4×4, 8×8, 16×16, which are the submultiple divisors of the original image size. The results are shown in Table 2. One can see that the best recognition rate is obtained when the patch size is 3×3 and the block size is 8×8. We can also note that the accuracy in UC Merced land Use dataset declines slowly with the increase of the patch size, which means that appropriate patch size and block size should be chosen in QPCANet. Generally speaking, the smaller the patch size is, the finer the scale of the image feature is. Furthermore, modest block size can capture enough energy to represent image in the coding layer of QPCANet.

We then study the effect of the patch size and the block size on the recognition accuracy of QPCANet, whose number of filters is set to $L_1 = L_2 = 8$ and overlapping ratio is set to 0.6, respectively. Similar to gray PCANet, the patch size is set to 3×3, 5×5, 7×7. We set the block size to 4×4, 8×8, 16×16, which are the submultiple divisors of the original image size.

We then research the effect of the overlapping ratio on the recognition accuracy of QPCANet, whose patch size, block size, and number of filters are set to 3×3, 8×8, and $L_1 = L_2 = 8$, respectively. The overlapping ratio varies from 0.1 to 0.9. Fig. 6 demonstrates the recognition rate

in this experiments. One can see that the recognition rates of different overlapping ratio fluctuate at 78%. We choose overlapping ratio 0.6 to get the best performance in UC Merced Land Use. The parameter model is concluded in Table 3.

5. Experimental Results and Discussion for different image datasets

5.1. Testing on UC Merced land use database

For the UC Merced land use database [57], we randomly selected 80 training images per class, and the remaining ones were used for testing. The QPCANet is trained with the number of filters $L_1 = L_2 = 8$, the block size is set to 8×8 and the overlapping ratio is fixed to 0.6. We changed the patch size $k_1 \times k_2$ from 3×3 to 7×7. The same parameters are used for RGB PCANet-2 and Gray PCANet. The results are reported in Table 4. Note that with the increase of patch size, the recognition accuracy of all networks decreases. The best performance for various patch sizes is highlighted in bold. In UC Merced Land use dataset, one-staged QPCANet (QPCANet-1) outperforms all the other one-staged networks. It is even better than some two-staged networks. Compared to RGB PCANet-2, the proposed QPCANet-2 increases the recognition accuracy by more than 6% in the case of patch size 3×3. The performance of Gray PCANet is worse than that of RGB PCANet and QPCANet in all cases.

The best performance of one-staged RGB LDANet (RGB LDANet-1), two-staged RGB LDANet (RGB LDANet-2), RGB SIFT, and RGB LBP are also given in Table 5. It turns out that QPCANet performs much better than RGB PCANet, gray PCANet, RGB LDANet, RGB SIFT and RGB LBP in UC Merced Land Use database. The reason may be due to the nature of UC Merced land use database which contains a large amount of rotation type images. Quaternion representation gives a concise and all-in-one representation for rotation, and is known for its successful application in computer graphics, computer vision, and orbital mechanics of satellites [43-52], etc. We suspect that the quaternion model of color image gives more rotation invariance to QPCANet when compared to RGB PCANet. In the next section, we further experimentally verify this conjecture.

5.2. Face recognition on Georgia Tech face database

For the Georgia Tech face database, we randomly pick up 10 images in each class as training images, and the others are used for testing images. We find that one-layer networks provide excellent results. Therefore, two-layer networks are not considered here. The number of filters L_1 in networks is set to 8 and the non-overlapping block is of size 8 × 8. The average accuracy of classifications over 10 times experiments are listed in Table 6.

We also compare the best performance of different methods in Table 7. We can see from the table that QPCANet performs better than RGB PCANet, gray PCANet, RGB LDANet, RGB SIFT and RGB LBP in Georgia Tech face database.

Next, we experimentally verify the hypotheses which we find in the experiment of UC Merced Land use database: QPCANet outperforms RGB PCANet when many rotated color images are contained in the database.

Firstly, we choose only one cropped front face image from 20 individuals. These images are zero padded to form a square image of the size 220 × 220 pixels. A representative padded image is shown in Fig. 7(a). The zero-padding process ensures that the rotation operation will not induce any border effect. For each individual (i.e. class), we rotate the padded front image from 0° to 360° by a step of 10° to obtain 36 images for each class. Some rotated images are shown in Fig. 7(b). We randomly select 18 images from each class to train networks and the remaining ones are used in the testing procedure. All images are then linearly scaled to 64 × 64 pixels. For QPCANet and PCANet, the number of filters, and the (non-overlapping) block size are set to $L_1 = L_2 = 8$, and 8×8, respectively. The recognition rates of QPCANet, RGB PCANet, and gray PCANet in terms of different parameter models are shown in Table 8.

The results of best performance of different methods shown in section 4.3 are listed in Table 9. Surprisingly, the accuracy of all networks decreases with the increase of stage. When the patch size is 5 × 5, the recognition accuracy of QPCANet-1 reaches 84.94%, which outperforms all the other methods. We can conclude that, compared with RGB PCANet, QPCANet allows to extract

features that are less sensitive to the rotation of color images. The performance of LDANet, RGB LBP, and RGB SIFT are not as good as QPCANet and RGB PCANet.

5.3. Texture discrimination in CURet Dataset

In our experiments, CURet database is randomly split into a training set and a testing set, with 23 training images for each class. The QPCANet is trained with the number of filters $L_1 = L_2 = 8$, the overlapping ratio is set to 0.5, and the block size is fixed to 8 × 8. The RGB PCANet and Gray PCANet use the same parameters as that of QPCANet.

The average recognition rates of QPCANet and PCANet over 5 different random splits are given in Table 10 and the performance of other methods whose parameter model shown in section 4.3 are given in Table 11. We can see that QPCANet outperforms other networks whether the stage is one or two. When the patch size is set to 7 × 7, the performance of QPCANet-1 approximately equals that of RGB PCANet-2 and outperform that of Gray PCANet-2. We also notice that RGB PCANet-1 outperforms Gray PCANet-1. The performance of RGB PCANet is much higher than that of Gray PCANet due to the use of color information and boundary information of texture dataset at the same time. The performance of RGB LDANet is neither good nor bad. RGB SIFT perform not well in texture database due to inconspicuous key points. RGB LBP outperform RGB SIFT slightly on this texture database.

5.4. Performance in Caltech-101

In Caltech-101 dataset, we randomly select 15 color images as training images per class and the remaining images are used for testing. With respect to the parameters setting of QPCANet, the overlapping ratio of block is set to 0.5 to preserve appropriate feature invariance for objects in Caltech-101 database.

For fair comparisons, appropriate parameters of RGB PCANet and gray PCANet are required. For both RGB PCANet and Gray PCANet, we enlarged patch size from 3×3 to 7×7. The block

size was fixed to 7×7 and overlapping ratio is set to 0.5. The average accuracy over 5 drawing of the training set is listed in Table 12 and the best accuracy rates of different methods whose use the showing parameter model is given in Table 13. We see that the best performance (63.78%) is achieved by QPCANet-2 when patch size is 3 × 3. With the increase of the patch size the accuracy decreases. This illustrates that too big patch size is not appropriate for color object recognition with QPCANet. We think there are at least two reasons: (1) generally speaking, the smaller patch size is, the finer scale of feature is. When the patch size increases, the invariance increases, but it comes with a partial loss of information [26], that is, a good choice of the patch size is the balances between the invariance enhancement and information loss. What's more important for the performance of QPCANet is that it should choose the small patch size to capture more details from color images, however, we do not worry too much about invariance since QPCA has enough rotation invariance for color images [43, 44]; (2) the convolutional layer of QPCANet is constructed by QPCA, which can capture enough energy to represent and classify color images even with very small patch sizes [26, 43, 44].

The performance improvement from QPCANet-1 to QPCANet-2 is not as large as that of RGB PCANet. But we note that the accuracy of QPCANet-1 is much better than other methods. In fact, the behavior of QPCANet-1 is very close to that of Gray PCANet-2 and RGB PCANet-2. This implies that we can achieve an acceptable result by just apply a simple QPCANet-1. Thus, QPCANet with only one layer still has competitive performance when comparing with other methods.

By comparing Gray PCANet with RGB PCANet, a strange phenomenon draws our attention. Why the accuracy of RGB PCANet-1 is lower than that of Gray PCANet-1, but RGB PCANet-2 outperforms Gray PCANet-2? First, let us consider some pictures extracted from Caltech-101 and reported in Fig. 8. We note that every horizontal pair of objects has the same main colors, but the objects are not belonging the same class. Thus, we cannot do a right classification only with color information. For color image classification, color information and boundary information seem

equally important. To some extent, color information is harmful to color image classification when an algorithm has not enough representation ability. That is, color is a burden for some classification algorithms. It seems that RGB PCANet-1 is such an algorithm, which is not able to utilize simultaneously color information and boundary information. For Gray PCANet-1, we changed color images to gray-scale ones and thus only the boundary information, which is effectively utilized by Gray PCANet-1, is preserved. Therefore, it is not surprising that Gray PCANet-1 outperforms RGB PCANet-1.

Surprisingly, a significant improvement of performance happens when upgrade RGB PCANet-1 to RGB PCANet-2 by adding one additional layer. RGB PCANet-2 seems to have enough ability to deal with both color information and boundary information. When compared with RGB PCANet-1, the representation ability of RGB PCANet-2 is significantly higher. RGB PCANet-2 also performs better than Gray PCANet-2 by utilizing color information of objects. Gray PCANet-2 does not perform significantly better than Gray PCANet-1 in this database. We can conclude that the accuracy of color object recognition is difficult to improve by only utilizing boundary information.

Why QPCANet outperforms the others methods (RGB PCANet, Gray PCANet, RGB LDANet, RGB SIFT and RGB LBP) in color image classification? We think that the reason is the utilization of quaternion model of color image. The quaternion representation of color image preserves the underlying spatial structures and relationships between R, G, B channels. QPCA is shown to be more suitable for color images representation, and allows enhancing the robustness of color images features [43, 44]. The shortcoming of RGB PCANet is that it neglects the structure and unity of color images.

To conceptualize the learned QPCANet filters of patch size 3 × 3, we draw them in Fig. 9. In this figure, the first four rows correspond to the real part and three imaginary parts of the first stage QPCA filters, and the others correspond to the four parts of the second stage QPCA filters. The learned RGB PCANet filters of patch size 3 × 3 are shown in Fig. 10. The first three rows

represent the one stage RGB PCA filters where each row corresponds to R, G, B channels, respectively. The last row represents the PCA filters of the second stage.

All the filters depicted in Fig. 9 and 10 are scaled within [0, 1]. Some learned filters exhibit similar shapes however; their numerical values are not the same. We notice that some similar QPCA filters occur several times in the different parts of quaternion. Such a filter redundancy may improve the intra-class invariance of feature maps. For the first three rows in RGB PCA filters, we notice that the last seven RGB PCA filters of each row have the same shape although their numerical values are not the same. It means that the similar filtering operations are performed in R, G, B channels.

6. Conclusion

In this paper, we presented a novel quaternion deep learning architecture named QPCANet for color image classification. The QPCANet uses the quaternion representation of color image as the input and thus keeps the spatial information of pixels. Moreover, through powerful properties of quaternion expression, even a trivial one layer QPCANet can achieve a satisfied classification results when compared with other two-layer deep learning networks in real domain.

The constructing of QPCANet relies upon five steps: quaternion representation of a color image, QPCA filter bank, binary operation, quaternion feature maps weighting and summing, pooling. In the first step, we represent color image as a compact quaternion to avoid dimensional curse and keep the spatial information in that. In QPCA filter bank step, a set of quaternion filters are trained from a set of color images by QPCA. Subsequently, binary operation, which builds a nonlinear layer, is performed for each pixel of the output of QPCA layer. In the quaternion features weighting and summing step, quaternion features are weighted and summed to obtain a discriminant features. In the pooling layer, histogram is performed to obtain the final feature vector of QPCANet. The network outputs are then used for classification purpose by employing the large scale linear SVM technique.

The conducted experiments showed that QPCANet outperforms RGB PCANet, Gray PCANet, RGB LDANet, RGB SIFT, and RGB LBP in various color image classification tasks, such as texture classification, face recognition, and object recognition. Additionally, we experimentally verified that QPCANet performs well in color images databases, which contains many rotated images.

In summary, the proposed QPCANet is a powerful tool for color image classification. It provides a new perspective of constructing deep learning networks via quaternion theory. Furthermore, the QPCANet can be also a valuable baseline for studying quaternion extension of other advanced deep learning networks.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61201344, 61271312, 61401085, 61572258, 11301074), and by the Project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, by the Qing Lan Project and the '333' project (No. BRA2015288), and by the Open Fund of China-USA Computer Science Research Center (KJR16026). The authors are also grateful to the anonymous reviewers for their constructive comments and suggestions to greatly improve the quality of this work and the clarity of the presentation.

References

[1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (60) (2004) 91-110.
[2] A.K. Jain, N.K. Ratha, S. Lakshmanan, Object detection using Gabor filters, Pattern Recognition 30 (2) (1997) 295-309.
[3]T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971-987.
[4] G. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural computation 18 (7) (2006) 1527-1554.
[5] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504-507.
[6] Y. LeCun, Y. Bengio, G.E. Hinton, Deep learning, Nature 521 (2015) 436-444.
[7] Y. Bengio, Learning deep architectures for AI, Foundations and trends® in Machine Learning. 2 (1) (2009) 1-127.
[8] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798-1828.
[9] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27-48.
[10] R. Salakhutdinov, G.E. Hinton, Deep boltzmann machines, Journal of Machine Learning Research 5 (2) (2009) 1967-2006.
[11] B.A. Olshausen, D.J. Field, Emergence of Simple-Cell Receptive Field Properties by Learning a

Sparse Code for Natural Images, Nature 381 (6583) (1996) 607-609.
[12] K. Yu, T. Zhang, Y. Gong, Nonlinear Learning Using Local Coordinate Coding, in: Proceedings of the NIPS, 2009.
[13] Y. Lin, Z. Tong, S. Zhu, K. Yu, Deep Coding Network, in: Proceedings of the NIPS, 2010.
[14] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Computation 1 (4) (1989) 541-551.
[15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278-2324.
[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of NIPS, 2012.
[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, *et al.*, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015) 211-252.
[18] P. Sermanet, et al, Overfeat: Integrated recognition, localization and detection using convolutional networks, arXiv preprint arXiv:1312.6229, 2013.
[19] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.
[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.*, Going deeper with convolutions, in: Proceedings of the CVPR, 2015.
[21] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, arXiv preprint arXiv:1512.03385, 2015.
[22] J. Xu, X. Luo, G. Wang, H. Gilmore, A. Madabhushi, A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images, Neurocomputing 191 (2016) 214-223.
[23] P. Qin, W. Xu, J. Guo, An empirical convolutional neural network approach for semantic relation classification, Neurocomputing 190 (2016) 1-9.
[24] Q. Guo, F. Wang, J. Lei, D. Tu, G. Li, Convolutional feature learning and Hybrid CNN-HMM for scene number recognition, Neurocomputing 184 (2015) 78-90.
[25] S. Mallat, Group invariant scattering, Communications on Pure and Applied Mathematics 65 (10) (2012) 1331-1398.
[26] J. Bruna, S. Mallat, Invariant scattering convolution networks, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1872-1886.
[27] T.H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, PCANet: A simple deep learning baseline for image classification? IEEE Trans. Image Process. 24 (12) (2015) 5017-5032.
[28] Y. Gan, T. Yang, C. He, A deep graph embedding network model for face recognition, in: Proceedings of the ICSP, 2014.
[29] Z. Feng, L. Jin, D. Tao, S. Huang, DLANet: A manifold-learning-based discriminative feature learning network for scene classification, Neurocomputing 157 (2015) 11-21.
[30] Z. Jia, B. Han, X. Gao, 2DPCANet: Dayside aurora classification based on deep learning, in: Proceedings of the CCCV, 2015.
[31] H. Qin, X. Li, J. Liang, Y. Peng, C. Zhang, DeepFish: Accurate underwater live fish recognition with a deep architecture, Neurocomputing 187 (2016) 49-58.
[32] Y. Zhao, R. Wang, W. Wang, W. Gao, Multi-level modified finite radon transform network for image upsampling, IEEE Trans. Circuits Syst. Video Technol. (2015).
[33] Z. Lei, D. Yi, Z. Li Stan, Learning stacked image descriptor for face recognition, IEEE Trans. Circuits Syst. Video Technol. (2015).
[34] S.Z. Li, B. Yu, W. Wu, S.Z. Su, R.R. Ji, Feature learning based on SAE-PCA network for human gesture recognition in RGBD images, Neurocomputing 151 (2015) 565-573.
[35] R. Zeng, J. Wu, L. Senhadji, H. Shu, Tensor object classification via multilinear discriminant analysis network, in: Proceedings of the ICASSP, 2015.
[36] K.E. Van De Sande, T. Gevers, C.G. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582-1596.
[37] Z. Zhang, M.B. Zhao, B. Li, P. Tang, F.Z. Li, Simple yet effective color principal and discriminant feature extraction for representing and recognizing color images, Neurocomputing 149 (2015) 1058-1073.
[38] R. Khemchandani, P. Saigal, Color image classification and retrieval through ternary decision structure based multi-category TWSVM, Neurocomputing 165 (2015) 444-455.
[39] Y. Zheng, J. Byeungwoo, D. Xu, Q.M.J. Wu, H. Zhang, Image segmentation by generalized

hierarchical fuzzy C-means algorithm, Journal of Intelligent and Fuzzy Systems 28 (2) (2015) 4024-4028.
[40] X. Wen, L. Shao, Y. Xue, W. Fang, A rapid learning algorithm for vehicle classification, Information Sciences 295 (1) (2015) 395-406.
[41] W.R. Hamilton, On quaternions. Proc Roy Irish Acad. 3 (1847) 1-16.
[42] F. Zhang, Quaternions and matrices of quaternions, Linear algebra and its applications 251 (1997) 21-57.
[43] N.L. Bihan, S.J. Sangwine, Quaternion principal component analysis of color images, in: Proceedings of the ICIP, 2003.
[44] S.C. Pei, J.H. Chang, J.J. Ding, Quaternion matrix singular value decomposition and its applications for color image processing, in: Proceedings of the ICIP, 2003.
[45] T.A. Ell, S.J. Sangwine, Hypercomplex Fourier transforms of color images, IEEE Trans. Image Process. 16 (1) (2007) 22-35.
[46] W.L. Chan, H. Choi, R.G. Baraniuk, Coherent multiscale image processing using dual-tree quaternion wavelets, IEEE Trans. Image Process. 17 (7) (2008) 1069-1082.
[47] L. Guo, M. Dai, M. Zhu, Multifocus color image fusion based on quaternion curvelet transform. Optics Expresses 20 (19) (2012) 18846-18860.
[48] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, J.L. Coatrieux, Color image analysis by quaternion-type moments, Journal of Mathematical Imaging and Vision 51 (1) (2014) 124-144.
[49] L.Q. Guo, M. Zhu, Quaternion Fourier–Mellin moments for color images, Pattern Recognition 44 (2) (2011) 187-195.
[50] S. Gai, New banknote defect detection algorithm using quaternion wavelet transform, Neurocomputing 196 (2016) 133-139.
[51] J.B. Kuipers, Quaternions And Rotation Sequences: A Primer With Applications To Orbits, Aerospace And Virtual Reality, Princeton University Press, Princeton, New Jersey, 1999.
[52] S. Liwicki, M.T. Pham, S. Zafeiriou, et al., Full-angle quaternions for robustly matching vectors of 3D rotations, in: Proceedings of the CVPR, 2014.
[53] G. Shakhnarovich, B. Moghaddam, Face recognition in subspaces, Handbook of Face Recognition (2011) 19–49.
[54] R.E. Fan, K.W. Chang, C.J. Hsieh, et al., LIBLINEAR: A library for large linear classification, The Journal of Machine Learning Research 9 (2008) 1871-1874.
[55] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, S. Li, Incremental Support Vector Learning for Ordinal Regression, IEEE Trans. Neural Netw. Learn. Syst. 26 (7) (2015) 1403-1416.
[56] B. Gu, V.S. Sheng, A Robust Regularization Path Algorithm for ν-Support Vector Classification, IEEE Trans. Neural Netw. Learn. Syst. (2016) 1-8.
[57] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: Proceedings of ICAGIS, 2010.
[58] A. Nefian, Georgia tech face database, 2013.
[59] K.J. Dana, B. Van Ginneken, S.K. Nayar, J.J. Koenderink, Reflectance and texture of real-world surfaces, ACM Trans. Graphics 18 (1) (1999) 151-157
[60] F.F. Li, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Computer Vision and Image Understanding 106 (1) (2007) 59-70.
[61] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the CVPR, 2009.
[62] E. Levent, S. Michael, K. Vipin, Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, in: Proceedings of the SDM, 2003.
[63] E. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, in: Proceedings of the AAAI workshop, 2000.

Pictures and Tables

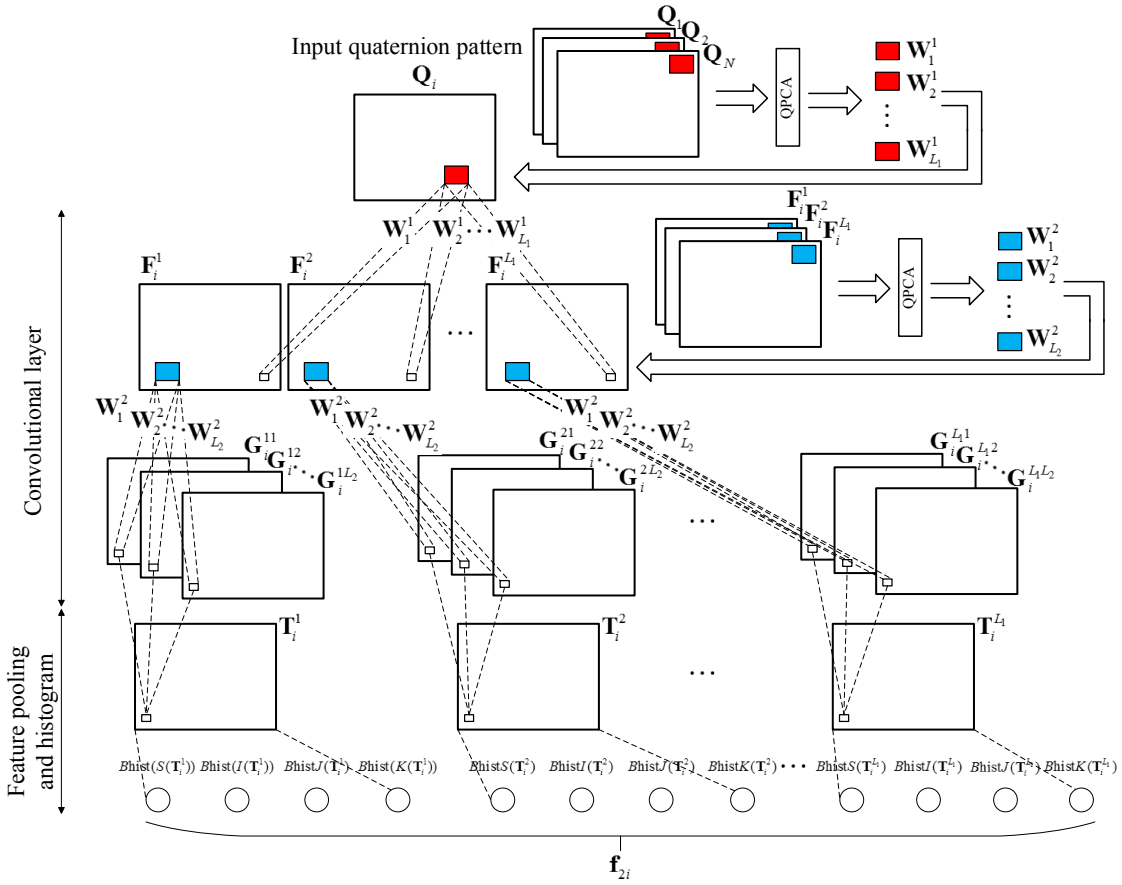

Fig. 1. Architecture of two-stage QPCANet

Fig. 2. Some representative examples of classes in the UC Merced land use database

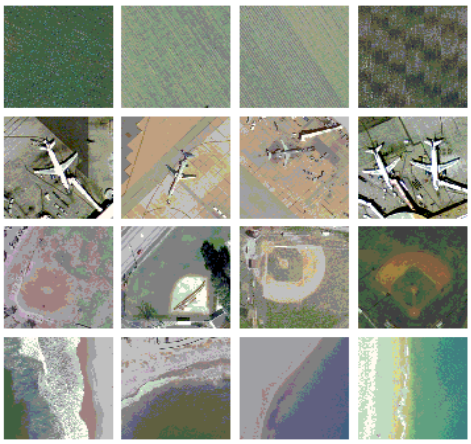

Fig. 3. Images contains rotation information in UC Merced land use database

Fig. 4. Sixteen individuals in Georgia Tech face database

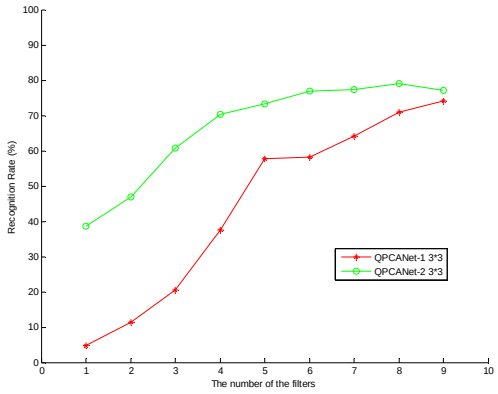

Fig. 5. Recognition accuracy of QPCANet on UC Merced land use database for different number of QPCA filters for fixed patch size of 3 × 3. The red line: The number of QPCA filters L_1 varies from 2 to 9 in the first stage. The green line: The number of QPCA filters L_2 varies from 2 to 9 in the second stage with L_1=8.

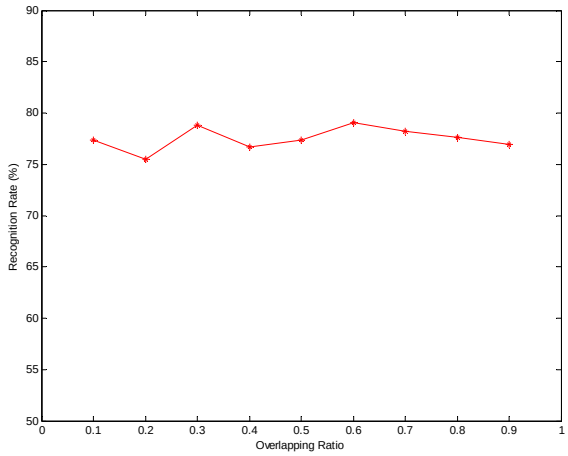

Fig. 6. The overlapping ratio of QPCANet changes from 0.1 to 0.9

(a) (b)

Fig. 7. Original representative padded front face image and some rotated images. (a) Representative padded front face image. (b) Some rotated face images in the new database.

Fig. 8. Some pictures have the same main colors but are not belonging to the same category of objects

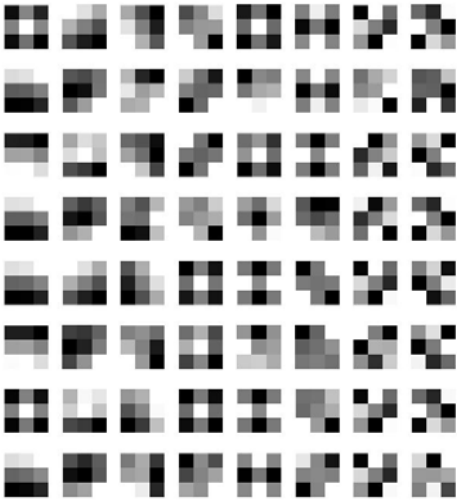

Fig. 9. QPCA filters in the first stage (rows 1 to 4) and second stage (rows 5 to 8)

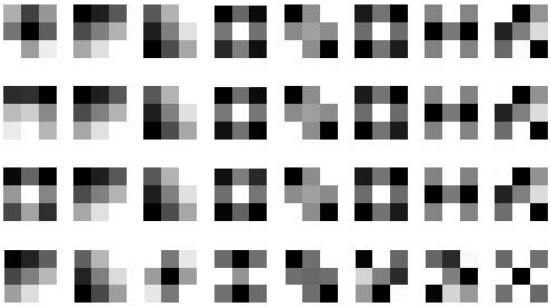

Fig. 10. RGB PCA filters in the first stage and second stage

Table 1
The accuracy (%) of color image classification in UC Merced Land use database in terms of different classifiers.

Method	KNN-cosine	KNN-Eclidean	Large scale SVM
QPCANet-2	63.81	50.00	**79.05**
RGB PCANet-2	**64.05**	**53.57**	73.33

Table 2
The recognition rate (%) of UC Merced Land Use dataset in relation with the different block sizes and the different patch sizes.

Block size \ patch size	3×3	5×5	7×7
4×4	70.00	66.43	60.00
8×8	79.05	70.01	67.62
16×16	80.95	75.00	74.29

Table 3
The parameter model of QPCA

QPCANet Paramter Model
Patch Size
Number of Filters in Each Stage
Block Size
Overlapping Ratio

Table 4
Recognition rate (%) of land use images on UC Merced Land Use database

Patch Size	QPCANet-1	QPCANet-2	RGB PCANet-1	RGB PCANet-2	Gray PCANet-1	Gray PCANet-2
3 × 3	70.24	**79.05**	65.24	73.33	65.00	66.67
5 × 5	68.81	**70.71**	64.29	67.14	57.14	58.81
7 × 7	**68.10**	67.62	62.14	63.10	55.24	56.90

Table 5

Comparison of recognition rates (%) of the various methods on UC Merced Land Use

Method	Recognition rates
RGB SIFT	55.47
RGB LBP	43.33
QPCANet-1	68.10
QPCANet-2	79.05
RGB PCANet-1	65.24
RGB PCANet-2	73.33
Gray PCANet-1	65.00
Gray PCANet-2	66.67
RGB LDANet-1	67.86
RGB LDANet-2	67.14

Table 6

Face recognition rates (%) of different networks on Georgia Tech face database

Patch Size	QPCANet-1	RGB PCANet-1	Gray PCANet-1
3 × 3	**100**	99.2	99.2
5 × 5	**100**	98.8	99.2
7 × 7	**100**	99.6	99.2

Table 7

Comparison of accuracy (%) of the methods on Georgia Tech face database

Method	Recognition rates
RGB SIFT	97.70
RGB LBP	80.00
QPCANet-1	100.00
RGB PCANet-1	99.60
Gray PCANet-1	99.20
RGB LDANet-1	99.60

Table 8

Comparison of face recognition rates (%) of rotation on Georgia Tech face database

Patch Size	QPCANet-1	QPCANet-2	RGB PCANet-1	RGB PCANet-2	Gray PCANet-1	Gray PCANet-2
3 × 3	**82.39**	77.12	80.56	74.17	60.22	47.77
5 × 5	**84.94**	79.17	71.49	68.61	69.44	61.66
7 × 7	**79.77**	76.94	70.83	65.56	70.56	63.61

Table 9
The accuracy (%) of different methods on Rotated Georgia Tech face database dataset

Method	Recognition rates
RGB SIFT	45.83
RGB LBP	42.22
QPCANet-1	84.94
QPCANet-2	79.17
RGB PCANet-1	80.56
RGB PCANet-2	74.17
Gray PCANet-1	69.44
Gray PCANet-2	63.61
RGB LDANet-1	86.94
RGB LDANet-2	64.72

Table 10
The accuracy (%) of texture classification on CURet dataset

Patch Size	QPCANet-1	QPCANet-2	RGB PCANet-1	RGB PCANet-2	Gray PCANet-1	Gray PCANet-2
3 × 3	96.94	**98.66**	92.28	96.36	90.71	93.70
5 × 5	97.81	**98.57**	95.60	98.08	95.60	96.56
7 × 7	98.05	**98.40**	96.29	98.03	96.03	96.20

Table 11
The accuracy (%) of texture classification on CURet dataset

Method	Recognition rates
RGB SIFT	60.91
RGB LBP	74.25
QPCANet-1	98.05
QPCANet-2	98.66
RGB PCANet-1	96.29
RGB PCANet-2	98.08
Gray PCANet-1	96.03
Gray PCANet-2	96.56
RGB LDANet-1	94.61
RGB LDANet-2	95.58

Table 12
The accuracy (%) of color image classification in Caltech-101 database

Patch Size	QPCANet-1	QPCANet-2	RGB PCANet-1	RGB PCANet-2	Gray PCANet-1	Gray PCANet-2
3 × 3	58.59	**63.78**	50.33	63.50	55.69	59.90
5 × 5	57.55	60.28	47.27	**63.31**	55.43	60.09
7 × 7	56.12	57.40	47.08	**59.38**	54.06	54.58

Table 13
Comparison of verification rates (%) on Caltech-101 dataset

Method	Recognition rates
RGB SIFT	37.98
RGB LBP	23.48
QPCANet-1	58.59
QPCANet-2	63.78
RGB PCANet-1	50.23
RGB PCANet-2	56.50
Gray PCANet-1	55.69
Gray PCANet-2	60.09
RGB LDANet-1	44.30
RGB LDANet-2	61.76