Regional Principal Component Analysis Network With the Rolling Guidance Filter for Classifying the Hyperspectral Images

XINPENG WANG¹, BINGO WING-KUEN LING², (Senior Member, IEEE), SHUANGYUAN YANG³, (Member, IEEE), AND XIULING LI⁴
¹Faculty of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China
²Information Engineering Faculty, Guangdong University of Technology, Guangzhou 510006, China
³School of Informatics, Xiamen University, Xiamen 361005, China
⁴School of Physics and Optoelectronic Engineering, Guangdong University of Technology, Guangzhou 510006, China
Corresponding author: Bingo Wing-Kuen Ling (yonquanzhang@gdut.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant U1701266, Grant 61372173, and Grant 61671163; in part by the Team Project of the Education Ministry of the Guangdong Province under Grant 2017KCXDT011; in part by the Guangdong Higher Education Engineering Technology Research Center for Big Data on Manufacturing Knowledge Patent under Grant 501130144; in part by the Guangdong Province Intellectual Property Key Laboratory Project under Grant 2018B03322016; in part by the Hong Kong Innovation and Technology Commission, Enterprise Support Scheme, under Grant S/E07017; and in part by the Guangdong University Ideological and Political Education Project under Grant 2019GXSZ013.

ABSTRACT Because conventional PCANET approach is that the conventional PCANET performs the PCA for all the segments of all the training pixel vectors, and this does not capture the difference between different segments of the same training pixel vectors, classification accuracy is not high. This paper proposes to employ a regional principal component analysis network with the rolling guidance filter (RPCANET_RGF) for performing the hyperspectral image (HSI) classification with few training samples. Regional principal component analysis network (RPCANET) proposed in this paper performs the PCA for each segment of all training pixel vectors. Besides, the rolling guidance filter (RGF) is used to remove the spatial noise and to enhance the edges of the HSIs. Different from the conventional convolutional neural networks (CNNs), the coefficients of the filters are obtained by performing the principal component analysis (PCA) on the regional segments of HSIs. This approach is also different from the conventional principal component analysis network (PCANET). Here, different segments of the same pixel image are processed by different Filters. Since the RPCANET_RGF is a general learning method that obtains the filter coefficients directly from the HSIs, the back propagation based training is not required. Hence, the RPCANET_RGF requires less computational power for performing the training compared to the CNN. Besides, as the RPCANET_RGF can make use of both the spectral information and the spatial information for performing the classification, the computer numerical simulation results show that the classification accuracy achieved by the RPCANET_RGF is higher than that by the conventional PCANET and other state of the art methods.

INDEX TERMS Regional principal component analysis network, rolling guidance filter, hyperspectral image, classification.

I. INTRODUCTION

The classifications of the hyperspectral images play an important role for the economic and social developments [3]–[6]. However, the hyperspectral images consist of many color bands. Extracting the only useful information from these color bands for performing the classification is crucial. This is because it can reduce the required computational power.

However, it may result to the reduction of the classification accuracy [13], [14] especially when the total number of the training samples is small [15].

To address the above difficulties, the multinomial logistic regression was proposed for performing the classification of the HSIs [7]. The SMLR-SpATV classifier using the SpATV regularization with the spectral information was also proposed [19]. Besides, the discriminative spectral spatial margin (DSSM) based method [33] and the spatial spectral regularized local discriminant embedding (SSRLDE) based
method [34] made a full use of the spatial information and the priori knowledge of the HSIs [18]. Moreover, by using a spatial hypergraph model, the spatial neighbors of the HSIs were combined and the fused spatial spectral features were extracted using a hypergraph embedding method. Furthermore, based on the graph framework, a marginal Fisher analysis (MFA) based method [29], a linear discriminant analysis (LDA) based method [22], a locally linear discriminant analysis (LLDA) based method [30], a locally linear embedding based method [23], a neighborhood preserving embedding (NPE) based method [25], a regularized local discriminant embedding (RLDE) method [34], a Laplacian Eigen map based method [24] and a locality preserving projection (LPP) based method [28] were proposed.

Recently, [26] propose a sparse-adaptive hypergraph discriminant analysis (SAHDA) method to obtain the embedding features of the HSI in this letter. [27] propose a hybrid-graph learning method (EHGDL) to reveal the complex high-order relationships of the HSI. A convolutional neural network (CNN) [1] was used for performing the classification of the hyperspectral images. It is worth noting that constructing and training an appropriate model for the CNN to perform the classification requires a huge training set. Nevertheless, obtaining a large amount of data for some specific applications is challenging. As a result, the developments of the deep learning applications are restricted. Moreover, a robust PCANET was used for performing the detection of the change of the HSIs [48]. The principal component analysis network (PCANET) is a very simple deep learning network which consists of only very basic data processing modules such as the cascaded PCA module, the binary hashing module and the segment-wise histogram module. Since it can achieve an excellent classification performance and its structure is simple as well as it is easy to perform the optimization of the hyper-parameters [2], the PCANET is useful for performing the image classification, the target recognition and other processing. As a result, many classification applications such as the character recognition [10] and the scene character recognition [11] have been developed. However, the PCANET is rarely used for performing the HSI classification.

In this paper, a new scheme based on the regional PCANET and the rolling guidance filter called the regional PCANET_RGF (RPCANET_RGF) is proposed for performing the HSI classification. In the RPCANET_RGF, the weights are computed based on the regional PCA instead of the back propagation algorithm. Thus, the required time for computing the weight is reduced. Moreover, the RPCANET_RGF exploits the spatial information of the HSIs via the rolling guidance filter. In particular, different segments of the same image are processed by different Filters, in other words, the RPCANET_RGF proposed in this paper performs the PCA for each segment of all training pixel vectors, while the conventional PCANET performs the PCA for all the segments of all the training pixel vectors.

The rest of this paper is organized as follows. In Section II, the related works on the PCANET are reviewed. Section III explains the details of the proposed HSI classification approach based on RPCANET_RGF. In Section IV, the corresponding computer numerical simulations are presented using three widely used HSI datasets. Finally, Section V concludes some remarks and discusses some possible future research directions.

II. RELATED WORKS
A. PCANET
PCA is used for the dimension reduction and fault diagnosis and fault tolerant control [12]. It projects the data from the high dimensional space to the low dimensional space. Then, keep the data in the low dimensional space such that the variance of the data in the low dimensional space is as large as possible. In this case, it facilitates the classification of the data. From here, it is required to find the most appropriate projection. To achieve this objective, the PCA algorithm calculates the eigenvalues and the eigenvectors of the covariance matrix of the data and chooses the eigenvectors of the covariance matrix with the largest corresponding eigenvalues to form the projection matrix. On the other hand, the PCANET employs the PCA to learn the convolution kernel of the convolutional neural network. Then, the binarization and the hash reset the two dimensional signal. In this paper, the regional principal component analysis network with the rolling guidance filter (RPCANET_RGF) is applied to perform the classification of the hyperspectral image.

1) THE INPUT LAYER
The PCANET model is discussed in details in [17]. Suppose that the dimensions of the hyperspectral image (HSIs) are \( m \times n \times h \). Here, \( m \) is the length of the HSI, \( n \) is the width of the HSI and \( h \) is the total number of the color bands of the HSI. This is also known as the dimension of each pixel vector. Therefore, the hyperspectral image is composed of \( m \times n \) pixel vectors. Suppose that there are \( N \) training pixel vectors taken from a HSI. They are denoted as \( T_j \) for \( j = 1, \ldots, N \). Obviously, \( N \leq mn \). Let the segment size or the length of the one dimensional filter be \( k_1 \) for all the layers of the PCANET. Here, only the PCA is employed to operate on the training pixel vectors \( T_i \) for \( i = 1, \ldots, N \). In the following sections, the training procedures of the conventional PCANET are presented in details.

2) THE FIRST HIDDEN LAYER
To design the first layer of the PCANET, each training pixel vector is divided into the segments with the lengths equal to \( k_1 \times 1 \). More precisely, two consecutive segments of each training pixel vector are related to be shifted by one point from each other. That is, the first segment in \( T_i \) is taken out from its 1\(^{st} \) point to its \( k_1^{th} \) point. The second segment in \( T_i \) is taken out from its 2\(^{nd} \) point to its \( (k_1+1)^{th} \) point. The \( j^{th} \) segment in \( T_i \) is denoted as \( \alpha_{i,j} = \begin{bmatrix} x_{i,j} & \cdots & x_{i,k_1+j-1} \end{bmatrix}^T \in \mathbb{R}^{k_1} \).
$R_i^{k_i \times 1}$ for $i = 1, \cdots, N$ and for $j = 1, \cdots, h - k_1 + 1$, where $x_{i,j}$ is the $j$th value in $T_i$. Then, $\alpha_{i,j} = \left[ x_{i,j} \cdots x_{i,k_1+j-1} \right]^T$ is obtained by subtracting $\alpha_{i,j}$ from its mean. Denote the matrix containing all the zero mean segments of $T_i$ as $\bar{X}_i = \left[ \bar{X}_{i,1} \cdots \bar{X}_{i,(h-k_1+1)} \right] \in R^{k_i \times (h-k_1+1)}$. The above procedures are repeated for all the training pixel vectors. Finally, denote the matrix containing all these zero mean segments of all the training pixel vectors as

$$X = [ \bar{X}_1 \cdots \bar{X}_N ] \in R^{k_i \times (h-k_1+1)N}. \quad (1)$$

Assume that the total number of the orthonormal filters in the first layer of the PCANET is $L_1$. Let these orthonormal filters be

$$W_i^1 = V_i \in R^{k_i \times 1} \quad \text{for } i = 1, 2, \cdots, L_1. \quad (2)$$

Put $V_i$ into a matrix and let this matrix be $V$. The PCA minimizes the reconstruction error using these orthonormal filters. Therefore, the design of these filters can be formulated as the following optimization problem:

$$\min_{V \in R^{k_i \times L_1}} \left\| X - VV^T X \right\|^2_F, \quad (3a)$$

subject to $V^T V = I_{L_1}, \quad (3b)$

where $I_{L_1}$ is the $L_1 \times L_1$ identity matrix. The solution of this optimization problem is the matrix with its column vectors being the first $L_1$ eigenvectors of $XX^T$. Here, the corresponding eigenvalues are sorted in the descending order. It is worth noting that these leading principal eigenvectors capture the main variations of the processed segments of the training pixel vectors.

Let the output of the $l$th filter in the first layer of the PCANET be $\Gamma_i^1$ for $i = 1, \cdots, N$ and for $l = 1, \cdots, L_1$. $\Gamma_i^1$ is the result of the convolution between the $l$th filter $W_i^1$ and the $l$th pixel vector $\Gamma_i \in \mathbb{R}^{k_1 \times 1}$. That is,

$$\Gamma_i^1 = \bar{X}_i^T W_i^1 \quad (4)$$

for $i = 1, \cdots, N$ and for $l = 1, \cdots, L_1$. Therefore, the matrix containing the outputs of the first layer of the PCANET is:

$$\left[ \Gamma_1^1 \cdots \Gamma_{L_1}^1 \cdots \Gamma_N^1 \cdots \Gamma_{L_1}^N \right] \in R^{(h+k_1-1) \times L_1 N}. \quad (5)$$

3) THE SECOND HIDDEN LAYER

The second layer of the PCANET is almost the same as its first layer. If the total number of the filters in the second layer is $L_2$, then there are $L_2$ outputs for each input $\Gamma_i^1$ in the second layer of the PCANET. Denote $W_i^2$ as the $l$th filter in the second layer of the PCANET. Each $W_i^2$ performs the convolution with $W_i^1$ for $\ell = 1, 2, \cdots, L_2$. Denote $O_{i,l}^2$ as the output of the second layer of the PCANET. $O_{i,l}^2$ can be obtained as follow:

$$O_{i,l}^2 = \Gamma_i^1 * W_i^2 \quad \text{for } i = 1, 2, \cdots, N, \quad \text{for } l = 1, 2, \cdots, L_1 \quad \text{and for } \ell = 1, 2, \cdots, L_2. \quad (6)$$
III. RPCANET_RGF

A. REGIONAL SEGMENTED PCA

Because conventional PCANET approach is that the conventional PCANET performs the PCA for all the segments of all the training pixel vectors, and this does not capture the difference between different segments of the same training pixel vectors, classification accuracy is not high. This paper proposes to employ a regional principal component analysis network with the rolling guidance filter (RPCANET_RGF) for performing the hyperspectral image (HSI) classification with few training samples. The RPCANET proposed in this paper performs the PCA for each segment of all training pixel vectors. A main difference between our proposed method and the conventional PCANET approach is that the conventional PCANET performs the PCA for all the segments of all the training pixel vectors. On the other hand, the RPCANET_RGF proposed in this paper performs the PCA for each segment of all training pixel vectors. The training procedures of the RPCANET_RGF are presented in details below.

Assume that the $j$th segment of the $i$th training pixel vectors is $\alpha_{i,j}$ for $j = 1, \ldots, h - k_1 + 1, i = 1, \ldots, N$. The $j$th segments of all the training pixel vectors are put side by side to form the $k_1 \times N$ matrix $Y_i$. The same operation applies to all other segments of the training pixel vectors. Eventually, we have:

\[
\begin{align*}
\hat{Y}_1 &= \begin{bmatrix} \sigma_{1,1} & \sigma_{2,1} & \cdots & \sigma_{N,1} \end{bmatrix} \in \mathbb{R}^{k_1 \times N}, \\
\hat{Y}_2 &= \begin{bmatrix} \sigma_{1,2} & \sigma_{2,2} & \cdots & \sigma_{N,2} \end{bmatrix} \in \mathbb{R}^{k_1 \times N}, \\
\vdots \\
\hat{Y}_{h-k_1+1} &= \begin{bmatrix} \sigma_{1,h-k_1+1} & \sigma_{2,h-k_1+1} & \cdots & \sigma_{N,h-k_1+1} \end{bmatrix} \in \mathbb{R}^{k_1 \times N}.
\end{align*}
\]

Here, $\sigma_{i,j}$ is obtained by subtracting $\alpha_{i,j}$ from its mean. The filters for the $i$th segment are the first $L_1$ eigenvectors of covariance matrix $\hat{Y}_i \hat{Y}_i^T$. Hence, the total number of the filters obtained at the first layer is $L_1(h - k_1 + 1)$. On the other hand, the total number of the filters obtained at the first layer of the conventional PCANET is $L_1$. FIGURE 1 shows the flowchart of filter parameter acquisition of regional principal component analysis network. The test samples are input into the network with these parameters to obtain the classification accuracy of the test samples. Denote PCANET as conventional PCANET. DenoteRegional_PCA as the method described in the preceding paragraph, and RPCANET as conventional PCANET withRegional_PCA at the first layer. Denote PCANET_RGF as conventional PCANET with RGF, and DenoteRPCANET_RGF as RPCANET with RGF.

B. SPATIAL AND SPECTRAL WEIGHTS IN RGF

This paper uses the method of RGF to solve the problem of noise. The RGF procedures are briefly summarized below:

1) STEP 1: GAUSSIAN FILTERING

Gaussian filter is a 2-dimensional convolution operator using Gaussian kernel for image blurring (removing details and noise). Assuming that the Gaussian Filter (window) size is $(2k + 1) \times (2k + 1)$, the formula of its elements is:

\[
h_{i,j} = \frac{1}{2\pi \sigma^2} e^{-\frac{(i-k)^2 + (j-k)^2}{2\sigma^2}} \text{ for } i,j = 1,2,\ldots,2k+1
\]

(9)

where $\sigma$ is the variance and $k$ determines the dimension of the kernel matrix, and define $M = \sum_{i,j=1}^{2k+1} h_{i,j}$.

(10) normalizes each element to get the element value of the Gaussian kernel $H_{i,j}$:

\[
H_{i,j} = \frac{1}{M} h_{i,j}
\]

(10)

The pixel value $R_{i,j}$ is obtained by the Gaussian filter and is expressed by (11):

\[
R_{i,j} = \sum_{i,j=1}^{2k+1} \sum_{i,j=1}^{2k+1} m_{i,j} H_{i,j}
\]

(11)

$m_{i,j}$ is value of neighboring pixels of $R_{i,j}$ of HSI.

In order to increase the diversity of the classification data, the larger the spatial window value, the larger the $\sigma$ of the Gaussian filter. The basis is the characteristic of the Gaussian distribution. The probability of the numerical distribution in $(\mu - 3\sigma, \mu + 3\sigma)$ is 0.9974. If the kernel matrix is larger, the corresponding $\sigma$ is also larger. On the contrary, if Sigma is larger, then the nuclear matrix coverage is also larger. The relationship between $k$ and $\sigma$ with reference to OpenCv is determined by (12):

\[
\sigma = k \times 0.3 + 0.8
\]

(12)
2) STEP 2, EDGE RECOVERY

The iterative edge recovery step forms the key function in RGF. In this process, an image \( H \) is iteratively updated. We denote \( R_{t+1} \) as the result in the \( t \)-th iteration. Initially, \( R_1 \) is the output of Gaussian filtering. The value of \( R_{t+1} \) in the \( t \)-th iteration is obtained in a joint bilateral filtering form given the original input image \( H \).

\[
R_{t+1}(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\left(-\frac{∥p - q∥^2}{2σ_s^2} - \frac{∥R_t(p) - R_t(q)∥^2}{2σ_r^2}\right) H(q)
\]  
(13)

\[
K_p = \sum_{q \in N(p)} \exp\left(-\frac{∥p - q∥^2}{2σ_s^2} - \frac{∥R_t(p) - R_t(q)∥^2}{2σ_r^2}\right)
\]  
(14)

H remains unchanged throughout all iterations, \( ∥R_t(p) - R_t(q)∥^2 \) is the Euclidean distance between point \( p \) and point \( q \). \( σ_s \) and \( σ_r \) control the spatial and spectral weights respectively.

In (13) and (14), if \( σ_r \) is generally set larger, the slope of the Gaussian function is more gradual, so that the variance between the pixels after step 1 become smaller, and noise pixels are cleared more thoroughly; In general, the value of \( σ_r \) is as small as possible, the slope of the Gaussian function is steeper, and after the \( t \)-th iteration, even if the values of \( R_t(p) \) and \( R_t(q) \) are close, the value of \( \exp\left(-\frac{∥p - q∥^2}{2σ_s^2} - \frac{∥R_t(p) - R_t(q)∥^2}{2σ_r^2}\right) \) increases exponentially. When \( ∥R_t(p) - R_t(q)∥^2 \gg ∥p - q∥^2 \), we can get

\[
\exp\left(-\frac{∥p - q∥^2}{2σ_s^2} - \frac{∥R_t(p) - R_t(q)∥^2}{2σ_r^2}\right) \approx \exp\left(-\frac{∥R_t(p) - R_t(q)∥^2}{2σ_r^2}\right),
\]  
(15)

this can better recover the image boundary, but if \( σ_r \) is too smaller, it is easier to restore noise data. This article uses adaptive \( σ_r \) by (16) and (17). Through experiments, \( k \) is better between 1/20 and 1/35.

\[
σ_r = km_r
\]  
(16)

\[
m_r = \sqrt{\frac{\sum_{s \in N(p)} ∥R_t(s) - R_t(p)∥^2}{n}}
\]  
(17)

The standard deviation between the edge pixel and its surrounding pixels is large, that is advantageous for separating different types of images. In general, Step 1 of RGF eliminates the effect of noise, but it blurs the boundaries of Hyperspectral image segmentation; and step 2 of RGF enhances Hyperspectral image detail and restores edges.

C. FLOW DIAGRAM OF RPCANET_RGF

FIGURE 2 shows the detailed segment diagram of the proposed (two-stage) RPCANET_RGF. Input pixel points pass through RGF, segment mean removal, 2 layers net, output layer and SVM, we will get the classification result.

The conventional PCANET performs the PCA for all the segments of all the training pixel vectors, so \( V_{ij} = V_{ij} = V_{hj} \). While the RPCANET_RGF performs the PCA for each segment of all training pixel vectors, so \( V_{ij} \neq V_{ij} \neq V_{hj} \).

IV. COMPUTER NUMERICAL SIMULATION RESULTS AND DISCUSSIONS

This paper reduces the noise and enhance boundary by RGF filter. Then a small number training pixels of HIS are randomly selected from the original data to obtain the Filter of two layers. Binarization processing is performed on each output matrix of the second layer, and the obtained result only contains 1 and 0. The result is hashed by squaring, and is divided into some equal length segments, and then we calculate the histogram information for each segment. The histogram information of the segment is then cascaded in the histogram features of each segment, and finally the segment extended histogram feature vector is obtained, and these vectors are passed through the libsvm classifier to obtain the classification result of the train data and the parameters of the SVM classifier. Next, we apply these Filters and trained SVM models to the test pixels of HIS to get classification accuracy. This operation was performed 10 times at different train data set, and the average value obtained was used as the final classification result.

In the subsequent experiments, we used the following settings: In the noise removal phase, select 8 times RGF for different spatial window range of pixels, the first spatial window range is \( 3 \times 3 \), the second time is \( 5 \times 5 \), and so on. The eighth time is \( 17 \times 17 \). For each spatial window range, 3 cycles of RGF operation, that is, 1 times Gaussian filter, 2 times RGF iterative filters, and every time one \( m \times n \times h \) image is get; after 8 cycles, eight images of \( m \times n \times h \) were obtained, and these images were connected in series to obtain an image of size \( m \times n \times 8h \) for subsequent RPCANET_RGF experiments. Size of segments The first and second layer are \( 40 \times 1 \), the first and second layers have a number of filters of 8, and the histogram is segmented using a non-overlapping method with a segment size of \( 8 \times 1 \) in the histogram generation phase.

In the follow part of this section, we provide an experimental evaluation for RPCANET_RGF using 3 real HSIs. In our experiments, the classification results are compared visually...
and quantitatively, where the quantitative comparisons are based on the class-specific accuracy, overall accuracy (OA), average accuracy (AA) and the k coefficient [1], all experiments are performed using MATLAB R2015a on an Intel Core(TM) i7 7700 2.60-GHz machine with 8 GB RAM.

Indian Pines was the first test data for hyperspectral image classification. Indian pine was imaged in 1992 by an airborne visual infrared imaging spectrometer (AVIRIS) and then intercepted in a size of 145 × 145. These areas are labeled for use as a hyperspectral image classification test. The AVIRIS imaging spectrometer has an imaging wavelength range of 0.4-2.5µm, which continuously images the ground objects in 220 consecutive bands. The remaining 200 bands after eliminating the 20 invalid bands are the subject of research. The spatial resolution of the image formed by the spectral imager is about 20m, so it is easier to produce mixed pixels, which makes classification difficult. The number of pixel is 145 × 145 = 21025. 10776 pixels is background which is painted blue, and 10249 pixels represent classification data, labeled with different colors. Schematic of 20 training samples randomly selected for each class and the ground-truth image are shown in FIGURE 3.

The second dataset is The Pavia University data set (Pavia_U) [40]. Pavia_U consists of data over the Pavia, Italy, acquired by the ROSIS instrument in 2001. The image scene intercepted in a size of 610 × 340 pixels and the remaining 103 bands after eliminating the 12 invalid bands are the subject of research. There are totally nine classes in the HSI data set. FIGURE 4 shows Schematic of 20 samples randomly selected for each category and the ground-truth map.

The third dataset is Salinas; Salinas consists of data over the Salinas Valley, California, characterized by high spatial resolution. The image scene intercepted in a size of 512 × 217 pixels and the remaining 204 bands after eliminating the 20 invalid bands are the subject of research. There are totally 16 classes in the HSI data set. Schematic of 20 samples randomly selected for each category and the ground-truth map are shown in FIGURE 5.

### A. COMPARATIVE EXPERIMENTS USING THE INDIAN PINES DATASET

In the experiments from FIGURE 6. To FIGURE 9, we randomly select 10% labeled samples per each class for training and the rest for testing, the training data is taken randomly 10 times, and the scales of the x axis of these figures are the segment length, while the scales of the y axis of these figures are the average classification accuracy of the 10 randomly selected test data. The comparison on the average classification accuracy between the conventional PCANET and the RPCANET under the different segment lengths of the Indian pine data is shown in FIGURE 6. It can be seen from FIGURE 6 that the overall rate of the classification accuracy of the RPCANET is 6% higher than that of the conventional PCANET without Regional_PCA. The filters in the first layer of the network in this experiment are the first eight eigenvectors corresponding to the largest eight eigenvalues. The results verify that the RPCANET can significantly improve the classification accuracy comparing with PCANET. This because RPCANET can captures the details of the difference between different segments of the same training pixel vectors, and then the classification accuracy can been improved.

When the segment length is small, the variance of the data projected to these eigenvectors is close to 0. These data are not conducive to the classification. On the other hand, when the segment length is large, it will lose a lot of the detailed features. If the paper does not specify, to tradeoff between these two effects the rest of this paper chooses the segment length as 40. The methods for obtaining the results shown in FIGURE 6 have not considered the spatial information. Hence, the obtained accuracies are relatively low.

The experimental results of FIGURE 6 verify RPCANET with RGF can also significantly improve the classification accuracy.

The classification accuracy of Indian pine of between RPCANET and RPCANET_RGF is shown in the FIGURE 7, and RPCANET is PCANET without RGF.
It can be seen from the FIGURE 7 that the RGF can significantly improve the classification accuracy.

In the paper, RGF is used in the data initialization phase. The number of RGF reverse iterations is set to 2, and the space window is set to $3 \times 3, 5 \times 5, 7 \times 3, 9 \times 9, 11 \times 11, 13 \times 13, 15 \times 15$ and $17 \times 17$ respectively. Superimposing the data obtained from these different spatial windows, we length of each vector of original HIS pixel is changed from 200 to 1600.

When the segment lengths of the first and second layers of RPCANET_RGF are 15, 20, 25, 30, 35, 40, the average classification accuracy is shown in FIGURE 8:

In FIGURE 8, the explanation for the decrease of classification accuracy when the segment length is 35: the training sample has certain randomness. Because the units of the ordinate are very small, the difference is very small, and the average classification accuracy is over 99%, when segment lengths are between 30 and 40, so the algorithm is also very stable.

Adding RGF, FIGURE 9 shows the comparison of overall classification accuracy between PCANET_RGF and RPCANET_RGF. In FIGURE 9, sections in the first layer (blue line) denote RPCANET_RGF, and no sections in the first layer (green line) denote PCANET_RGF. Comparing with FIGURE 6, the classification accuracy is significantly improved after adding RGF. As the segment length increases, although the classification accuracy rate of RPCANET_RGF increases first and then decreases, the variety of classification accuracy rate was not obvious, and it was around 99%. As the segment length PCANET_RGF is increased, the classification accuracy rate increases, and then rapidly reduces. Indicating that the conventional PCANET is greatly influenced by the segment length, while RPCANET_RGF is very little influenced by the segment length, the algorithm is more robust. The reason for this phenomenon is that when the conventional PCANET performs singular value decomposition algorithm, it uses the set of all segments of all training pixel vectors. As the length of the data segment increases, for example, 100 spectral bands are taken as a segment, however we only take the first 8 singular values, the difference between the 8 singular values is small, and so the difference between the data to be classified is small. When the segment length is small, the difference between the singular values is large, and it leads to a large difference between the data to be classified. However for RPCANET_RGF, since PCA is performed separately for each segment, the effect of segment length is less influence, and the classification accuracy is higher.
We randomly select 10% labeled samples per each class for training and the rest for testing. The proposed approaches are benchmarked with seven state-of-the-art methods including kernel-based LORSAL (KLORSAL) [21], BELM [42], NLELM [20], KELM [20], the ASML.KELM, SVM [20], SVM with CK (SVM-CK) [20], LORSAL [21], SMLR.SpATV (KLORSAL with the weighted Markov random field) [35], [36], PCRC [39], SMLR [40], SR [41]. The quantitative results is average of 10 times for each methods, 0 shows the OAs of different methods. 0 shows that method of RPCANET.RGF can achieve 99.18 OA, and perform better than other methods. Because RPCANET.RGF makes different segments of the same pixel vector be processed with different filters, and the same filters are used for segments of the same sequence number of vectors of different pixels.

Then we randomly select 20 labeled samples per each class for training and the rest are test samples. The paper compares RPCANET_RGF with seven state-of-the-art methods including SVM [1], SVM with CK (SVM-CK) [20], LORSAL [21], SMLR.SpATV (KLORSAL with the weighted Markov random field) [35], [36], PCRC [39], SMLR [40], SR [41]. The LIBSVM [37] software is used for the implementation of the SVM and the SVM-CK. The result is average of 10 times for each method. The class-specific accuracies, overall accuracies (OAs), average accuracies (AAs) and k coefficients (k) of different methods are showed in TABLE 2. FIGURE 10 illustrates full classification maps obtained by different methods. We randomly pick a result from the results of 10 runs to form Figure 10. Obviously, RPCANET_RGF works best, because different segments of the same pixel vector are processed with different Filters, and RPCANET.RGF can better distinguish between different segments. We can also see that the maps obtained by K-SVM, PCRC, SMLR, SR, LORSAL and SVM-CK have heavy noisy appearance. Although SMLR.SpTV gets better results, their classification maps still divide samples from one category into another.

When we randomly select 5,10,15,20 labeled samples from each class to form the training set, and the rest forms the test set, FIGURE 11 shows the average accuracy of different methods. The AA is average of 10 times for each method.
TABLE 2. Class-specific classification accuracies (in percentage), OA (in percentage), AA (in percentage) and kappa coefficient for the AVIRIS Indian Pines dataset. K-SVM; PCRC; SMLR; SR; LORSAL; SVM-CK; SMLR_SpTV; RPCANET_RGF. (The best results are highlighted in bold typeface).

| Class Name          | K-SVM   | PCRC    | SMLR   | SR     | LORSAL | SMLR_SpTV | SVM-CK | RPCANET_RGF |
|---------------------|---------|---------|--------|--------|--------|-----------|--------|-------------|
| Alfalfa             | 91.47±7.26 | 82.35±11.35 | 80.29±6.8 | 69.41±7.49 | 87.35±6.36 | 87.35±1.52 | 98.82±2.06 | 99.71±0.93 |
| Corn-notill         | 66.53±5.33 | 52.48±12.26 | 61.84±6.55 | 51.95±13.13 | 66.49±4.89 | 66.49±6.59 | 78.47±6.2 | 82.42±9.4 |
| Corn-mintill        | 68.72±2.75 | 40.97±13.76 | 40.97±6.94 | 49.03±12.98 | 61.28±4.57 | 61.28±8.05 | 86.68±7.46 | 91.24±3.07 |
| Corn                | 87.1±5.47 | 75.7±6.64 | 71.45±7.06 | 71.96±10.17 | 82.71±4.45 | 82.71±0.25 | 92.2±2.61 | 99.95±0.15 |
| Grass-plant         | 86.14±5.31 | 66.25±12.62 | 81.7±4.35 | 82.37±5.95 | 81.22±8 | 81.22±10.26 | 84.88±4.35 | 91.55±5.24 |
| Grass-trees         | 89.71±3.86 | 95.12±1.93 | 91.39±1.94 | 90.84±9.66 | 93.01±2.19 | 93.01±0.84 | 93.47±3.17 | 99.02±1 |
| Grass-plant-mowed   | 95.38±3.97 | 63.85±21.77 | 91.54±4.37 | 42.31±14.16 | 93.85±7.94 | 93.85±0 | 97.69±5.19 | 100±0 |
| Hay-windrowed       | 94.69±2.59 | 89.04±15.55 | 94.12±1.39 | 94.48±5.44 | 96.27±1.78 | 96.27±0.09 | 96.57±1.99 | 100±0 |
| Oats                | 99±3.16 | 96±5.16 | 83±13.37 | 56±14.3 | 99±3.16 | 99±0 | 96±8.43 | 100±0 |
| Soybean-notill      | 74.28±4.19 | 63.12±15.81 | 61.12±3.95 | 43.93±8.6 | 61.64±6.59 | 61.64±5.97 | 85.27±4.58 | 90.27±3.58 |
| Soybean-mintill     | 55.47±7.05 | 33.68±14.44 | 40.21±4.29 | 60.9±13.41 | 50.72±6.42 | 50.72±7.5 | 73.13±5.76 | 84.04±4.89 |
| Soybean-clean       | 78.69±7.59 | 62.95±15.39 | 65.91±4.7 | 39.87±17.62 | 80.35±8.02 | 80.35±5.08 | 82.34±5.45 | 94.29±2.2 |
| Wheat               | 99.11±0.7 | 100±0 | 99.11±0.7 | 97.66±3.25 | 99.69±0.27 | 99.69±0.16 | 99.17±0.5 | 99.95±0.16 |
| Woods               | 89.02±6.66 | 97.2±1.73 | 84.26±4.77 | 94.14±4.3 | 88.97±4.91 | 88.97±4.33 | 90.28±4.98 | 95.76±5.6 |
| Buildings-Grass-Trees-Drives | 67.72±9.19 | 42.33±10.51 | 72.5±9.46 | 51.31±8.53 | 67.5±6.55 | 67.5±4.87 | 88.61±7.5 | 97.75±3.94 |
| Stone-Steel-Towers  | 93.2±4 | 98.93±1.51 | 90.8±2.77 | 98.13±2.28 | 91.73±6.88 | 91.73±1.13 | 97.6±2.93 | 100±0 |
| OA                  | 73.68±1.93 | 61.24±2.89 | 64.12±1.83 | 65.66±2.78 | 70.78±1.26 | 86.44±1.55 | 83.73±0.86 | 90.59±1.15 |
| AA                  | 83.52±0.84 | 72.50±1.96 | 75.64±1.15 | 68.39±2.10 | 81.36±1.00 | 92.90±0.97 | 90.07±0.76 | 95.37±0.77 |
| k                   | 70.39±2.12 | 56.87±2.94 | 59.85±1.95 | 61.05±3.29 | 67.2±1.35 | 84.67±1.71 | 81.62±0.97 | 89.31±1.29 |

As the number of training samples increases, AAs increase basically. When training samples is limited, for example, 15 samples from each class, RPCANET_RGF can obtain an AA over 90%, which is around 7% higher than the SMLR_SpTV which is the second best method. In addition, RPCANET_RGF has performed well when the number of training samples in each category was more than or equal to 10. This Experiments show that RPCANET_RGF can achieve higher average accuracy with less training samples.

B. EXPERIMENTS WITH THE Pavia UNIVERSITY DATA SET

For the Pavia University data set, we randomly select 20 labeled samples per each class for training and the rest are test samples. The paper compares RPCANET_RGF with the same seven state-of-the-art methods including used by the first database. The results is average of 10 times for each methods, and the class-specific accuracies, overall accuracies(OAs), average accuracies(AAs) and kappa coefficients(k) of different methods is shown in TABLE 3. We randomly pick a result from the results of 10 runs to form FIGURE 12, which shows RPCANET_RGF also gives best results. We can also see that the maps obtained by K-SVM, PCRC, SMLR, SR, LORSAL and SVM-CK have heavy noisy appearance. Although SMLR_SpTV gets better results, their classification maps still divide samples from one category into another.

When we randomly select 5, 10, 15, 20 labeled samples from each class to form the training set, and the rest forms the test set, FIGURE 13 shows the average accuracy of different methods. The AA is average of 10 times for each method. As the number of training samples increases, AAs increase basically. When training samples is limited, for example, 15 training samples from each class, RPCANET_RGF can obtain an AA over 94%, which is around 7% higher than the SMLR_SpTV which is the second best method. In addition, RPCANET_RGF has performed well when the number of training samples in each category was more than or equal to 10. This experiments show that RPCANET_RGF can achieve higher average accuracy with less training samples.
TABLE 3. Class-specific classification accuracies (in percentage), OA (in percentage), AA (in percentage) and kappa coefficient for the Pavia university data set. K-SVM; PCRC; SMLR; SR; LORSAL; SVM-CK; SMLR_SpTV; RPCANET_RGF. (the best results are highlighted in bold typeface).

| Class Name      | K-SVM  | PCRC   | SMLR   | SR     | LORSAL | SMLR_SpTV | SVM-CK | RPCANET |
|-----------------|--------|--------|--------|--------|--------|-----------|--------|---------|
| Asphalt         | 75.11±6.07 | 25.51±6.24 | 54.66±5.84 | 7.05±18.45 | 56.42±5.2 | 85.73±5.96 | 81.09±6.95 | 86.28±3.23 |
| Meadows         | 72.2±4.33   | 81.57±5.16  | 69.97±6.1  | 50.26±11.11 | 71.82±4.65 | 85.18±5.43 | 84.6±6.58 | 94.31±3.06 |
| Gravel          | 78.48±7.03  | 88.11±3.33  | 67.57±6.32 | 39.56±23.8 | 73.48±6.55 | 87.48±8.58 | 81.2±3.53 | 91.52±4.47 |
| Trees           | 90.69±2.94  | 96.25±1.75  | 88.67±5.91 | 85.34±19.94 | 90.61±3.28 | 89.87±4.46 | 90.45±1.5 | 96.46±1.68 |
| Painted metal sheets | 99.22±0.29  | 99.98±0.03  | 98.79±0.54 | 99.71±0.12 | 98.59±0.72 | 99.81±1.05 | 99.53±0.21 | 100±0 |
| Bare Soil       | 76.29±9.7   | 65.83±2.26  | 68.53±2.33 | 41.26±11.34 | 70.77±1.68 | 86.51±2.03 | 84.74±3.48 | 93.25±1.58 |
| Bitumen         | 83.40±1.57  | 74.40±1.20  | 74.40±1.20 | 38.85±2.75 | 76.50±2.04 | 88.41±2.47 | 87.50±1.61 | 94.74±1.15 |
| Self-Blocking Bricks | 70.44±2.36  | 55.83±2.53  | 60.16±2.42 | 28.91±8.84 | 63.09±1.93 | 82.45±2.65 | 80.27±4.27 | 91.14±2.03 |

TABLE 4. Class-specific classification accuracies (in percentage), OA (in percentage), AA (in percentage) and kappa coefficient for the Pavia university data set. K-SVM; PCRC; SMLR; SR; LORSAL; SVM-CK; SMLR_SpTV; RPCANET_RGF. (the best results are highlighted in bold typeface).

| Class Name      | K-SVM  | PCRC   | SMLR   | SR     | LORSAL | SMLR_SpTV | SVM-CK | RPCANET_RGF |
|-----------------|--------|--------|--------|--------|--------|-----------|--------|-------------|
| Asphalt         | 75.11±6.07 | 25.51±6.24 | 54.66±5.84 | 7.05±18.45 | 56.42±5.2 | 85.73±5.96 | 81.09±6.95 | 86.28±3.23 |
| Meadows         | 72.2±4.33   | 81.57±5.16  | 69.97±6.1  | 50.26±11.11 | 71.82±4.65 | 85.18±5.43 | 84.6±6.58 | 94.31±3.06 |
| Gravel          | 78.48±7.03  | 88.11±3.33  | 67.57±6.32 | 39.56±23.8 | 73.48±6.55 | 87.48±8.58 | 81.2±3.53 | 91.52±4.47 |
| Trees           | 90.69±2.94  | 96.25±1.75  | 88.67±5.91 | 85.34±19.94 | 90.61±3.28 | 89.87±4.46 | 90.45±1.5 | 96.46±1.68 |
| Painted metal sheets | 99.22±0.29  | 99.98±0.03  | 98.79±0.54 | 99.71±0.12 | 98.59±0.72 | 99.81±1.05 | 99.53±0.21 | 100±0 |
| Bare Soil       | 76.29±9.7   | 65.83±2.26  | 68.53±2.33 | 41.26±11.34 | 70.77±1.68 | 86.51±2.03 | 84.74±3.48 | 93.25±1.58 |
| Bitumen         | 83.40±1.57  | 74.40±1.20  | 74.40±1.20 | 38.85±2.75 | 76.50±2.04 | 88.41±2.47 | 87.50±1.61 | 94.74±1.15 |
| Self-Blocking Bricks | 70.44±2.36  | 55.83±2.53  | 60.16±2.42 | 28.91±8.84 | 63.09±1.93 | 82.45±2.65 | 80.27±4.27 | 91.14±2.03 |

can achieve higher average accuracy with less training samples.

C. EXPERIMENTS WITH THE SALINAS DATA SET

For the Salinas data set, we randomly select 20 labeled samples per each class for training and the rest are test samples. The paper compares RPCANET_RGF with the same seven state-of-the-art methods including used by the first database. The results is average of 10 times for each method, and the class-specific accuracies, overall accuracies(OAs), average accuracies(AAs) and k coefficients(k) of different methods is shown in TABLE 3. We randomly pick a result from the results of 10 runs to form FIGURE 14, which shows RPCANET_RGF also gives best results. We can also see that the maps obtained by K-SVM, PCRC, SMLR, SR, LORSAL and SVM-CK have heavy noisy appearance. Although SMLR_SpTV gets better results, their classification maps still divide samples from one category into another.

Next, when we randomly select 5, 10, 15, 20 labeled samples from each class to form the training set, and the rest forms the test set, FIGURE 15 shows the average accuracy of different methods. The AA is average of 10 times for each method. As the number of training samples increases, AAs increase basically. When training samples is limited, for example, 15 training samples from each class, RPCANET_RGF can obtain an AA over 96%, which is around 4% higher than the SMLR_SpTV which is the second best method. In addition,
TABLE 5. Class-specific classification accuracies (in percentage), OA (in percentage), AA (in percentage) and kappa coefficient for the salinas data set. K-SVM; PCRC; SMLR; SR; LORSAL; SVM-CK; SMLR_SpTV; RPCANET_RGF. (the best results are highlighted in bold typeface).

| Class Name                  | K-SVM | PCRC | SMLR | SR  | LORSAL | SMLR_SpTV | SVM-CK | RPCANET_RGF |
|-----------------------------|-------|------|------|-----|--------|-----------|--------|-------------|
| Broccoli_green_weeds_1      | 98.14±0.93 | 99.95±0.05 | 96.71±1.57 | 99.79±0.21 | 87.35±6.36 | 99.96±0.03 | 98.14±0.93 | 100±0  |
| Broccoli_green_weeds_2      | 98.08±1.7 | 96.23±1.48 | 99.29±0.1 | 97.48±1.77 | 66.49±4.89 | 100±0 | 98.08±1.7 | 99.84±0.14 |
| Fallow                     | 96.11±3.8 | 74.32±6.65 | 95.26±2.04 | 66.21±12.33 | 61.28±4.57 | 94.87±6.44 | 96.11±3.8 | 100±0  |
| Fallow_rough_low           | 98.9±0.68 | 99.13±0.71 | 97.91±1.36 | 97.35±1.17 | 82.71±4.45 | 99.03±0.39 | 98.9±0.68 | 98.4±1.17 |
| Fallow_smooth              | 96.09±1.94 | 98.25±0.12 | 95.92±1.41 | 95.82±3.72 | 81.22±8 | 99.3±0.6 | 96.09±1.94 | 99.92±0.13 |
| Stubble                    | 98.96±1.29 | 99.91±0.02 | 99.32±0.16 | 99.88±0.03 | 93.01±2.19 | 100±0.01 | 98.96±1.29 | 100±0  |
| Celery                     | 99.38±0.09 | 99.81±0.05 | 99.12±0.42 | 98.3±1.91 | 93.85±7.94 | 99.98±0.01 | 99.38±0.09 | 99.75±0.21 |
| Grapes_untrained           | 72.38±8.51 | 79.84±20.26 | 68.55±10.3 | 63.1±19.14 | 96.27±1.78 | 81.37±10.7 | 72.38±8.51 | 90.63±9.36 |
| Soil_vineyard_develop      | 98.55±1.06 | 99.99±0.03 | 99.29±0.33 | 99.72±0.21 | 99±3.16 | 100±0 | 98.55±1.06 | 100±0  |
| Corn_senesced_green_weeds  | 88.62±2.12 | 87.26±2.27 | 87.91±2.54 | 87.85±3.2 | 61.64±6.59 | 95.74±1.37 | 88.62±2.12 | 98.68±1.2 |
| Lettuce_romeaine_4wk       | 94.14±2.85 | 96.35±0.59 | 95±1.64 | 94.62±2.89 | 50.72±6.42 | 99.9±0.1 | 94.14±2.85 | 99.9±0.1 |
| Lettuce_romeaine_5wk       | 97.97±1.42 | 60.36±4.37 | 95.78±2.43 | 58.37±24.83 | 80.35±8.02 | 99.96±0.04 | 97.97±1.42 | 100±0  |
| Lettuce_romeaine_6wk       | 98±1.45 | 93.34±4.55 | 97.13±1.84 | 86.27±6.66 | 99.69±0.27 | 99.06±0.69 | 98±1.45 | 100±0  |
| Lettuce_romeaine_7wk       | 94.37±3.43 | 94.06±1.11 | 91.9±2.64 | 91.63±1.74 | 88.97±4.91 | 99.74±0.21 | 94.37±3.43 | 100±0  |
| Vinyard_untrained          | 65.37±8.33 | 47.19±18.65 | 61.85±4.68 | 45.34±21.05 | 67.5±6.55 | 78.68±15.6 | 65.37±8.33 | 85.67±13.05 |
| Vinyard_vertical_trellis   | 96.12±2.4 | 97.52±0.75 | 95.71±1.24 | 91.99±2.35 | 91.73±6.88 | 99.66±0.39 | 96.12±2.4 | 99.83±0.15 |
| OA                          | 87.57±1.99 | 84.79±2.32 | 86.17±1.92 | 80.11±1.37 | 87.49±1.10 | 92.69±0.92 | 87.57±1.99 | 99.45±1.37 |
| AA                          | 93.20±0.68 | 88.97±0.96 | 92.29±0.77 | 85.86±1.63 | 93.41±0.47 | 96.70±0.48 | 93.20±0.68 | 98.29±0.68 |
| $k$                         | 86.18±1.30 | 83.01±2.50 | 84.63±2.10 | 77.88±1.46 | 86.11±1.19 | 91.87±1.02 | 86.18±1.30 | 95.49±1.54 |

RPCANET_RGF has performed well when the number of training samples in each category was more than or equal to 10. This experiment also proves that RPCANET_RGF can achieve higher average accuracy with less training samples than the conventional PCANET and other state of the art methods.

In order to increase comparability, we choose 2% for each category from The Pavia University data when selecting training data. Full classification maps obtained by different methods are illustrated in FIGURE 16. Each map is constructed from a randomly selected result from ten results. As can be seen from FIGURE 16, RPCANET_RGF gives satisfactory results on smooth homogeneous regions by making use of the Regional segment spectral and stacking the joint spatial information. We can also see that the maps obtained by K-SVM, PCRC, SMLR, SR and SVM-CK have heavy noisy appearance. SVM-CK, LORSAL and K_SVM got similar classification accuracy, but the map of SMLR-CV and K_SVM has more sesame and pepper noise information than LORSAL. One of possible reasons for this may be that the spectral-based methods cannot make use of Regional segment spectral and stacking the joint spatial information of the HSI. Although SMLR SpTV show improvements on the classification results from TABLE 5 is slightly higher than RPCANET_RGF in some ground categories, their classification maps still divide more samples from one category into another. The results in TABLE 5 and FIGURE 16 further prove the proposed RPCANET_RGF not only reduces noise, but also provides higher OA, AA, kappa and CA than other methods.

Last, when we randomly select 1%, 2%, 3%, 4% labeled samples from each class to form the training set, and the rest forms the test set, FIGURE 17 shows the average accuracy of different methods. The AA is average of 10 times for each method. When the percentage of training samples increases, AAs increase basically. When training samples is limited, for example, 1% samples from each class, RPCANET_RGF can obtain an AA over 98%, which is around 2% higher than the SMLR_SpTV which is the second best method. In addition, RPCANET_RGF has performed well when the
number of training samples in each category was more than or equal to 1%. These experiments once again show that RPCANET_RGF can achieve higher average accuracy with less training samples.

V. CONCLUSION AND FUTURE WORKS

Inspired by PCANET and spatial relationship between neighboring pixels, we develop the RPCANET_RGF. RPCANET_RGF applies regional PCA to every segments of the same pixel picture, so that different filters processes different segments of the same pixel vector and the same Filters are used for segments of the same number of vectors of all pixels. RGF can effectively eliminate the spatial noise information of HSIs. The experiments have shown that RPCANET_RGF outperforms some state-of-the-art HSI classification methods even with a small number of training samples.

In our future work, first, tensor methods can extract different features, and we can design some new learning methods to learn spectral-spatial features of each segment and mix these methods using Optimization Methods. Second, for our structure, more features such as band selection will be explored in the future. Last, we will try to use parallel computing to further improve its computational efficiency.

REFERENCES

[1] H. Pu, Z. Chen, B. Wang, and G.-M. Jiang, “A novel spatial-spectral similarity measure for dimensionality reduction and classification of hyperspectral imagery,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 11, pp. 7008–7022, Nov. 2014.
[2] F. Hu, M. Zhou, P. Yan, K. Bian, and R. Dai, “PCANet: A common solution for laser-induced fluorescence spectral classification,” IEEE Access, vol. 7, pp. 107129–107141, 2019.
[3] H. Li, G. Xiao, T. Xia, Y. Y. Tang, and L. Li, “Hyperspectral image classification using functional data analysis,” IEEE Trans. Cybern., vol. 44, no. 9, pp. 1544–1555, Sep. 2014.
[4] S. Jia, L. Shen, J. Zhu, and Q. Li, “A 3-D Gabor phase-based coding and matching framework for hyperspectral imagery classification,” IEEE Trans. Cybern., vol. 48, no. 4, pp. 1176–1188, Apr. 2018.
[5] Y. Yuan, J. Lin, and Q. Wang, “Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization,” IEEE Trans. Cybern., vol. 46, no. 12, pp. 2966–2977, Dec. 2016.
[6] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, “Multiple feature learning for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 3, pp. 1592–1606, Mar. 2015.
[7] J. Li, J. M. Bioucas-Dias, and A. Plaza, “Semi supervised hyperspectral image classification using soft sparse multinomial logistic regression,” IEEE Geosci. Remote Sens. Lett., vol. 10, no. 2, pp. 318–322, Mar. 2013.
[8] F. Wang, R. Zhang, and Q. Wu, “Hyperspectral image classification based on PCA network,” in Proc. 8th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS), Aug. 2016, pp. 1–4.
[9] M. Faupel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, “Advances in spectral-spatial classification of hyperspectral images,” Proc. IEEE, vol. 101, no. 3, pp. 652–675, Mar. 2013.
[10] R. Liu and T. Lu, “Character recognition based on PCANet,” in Proc. 15th Int. Symp. Parallel Distrib. Comput. (ISPDC), 2016, pp. 364–367.
[11] C. Chen, D. Wang, and H. and Wang, “Scene Character Recognition Using PCANET,” in Proc. ICIMCS 2015, pp. 65–65–4.
[12] Y. Wu, B. Jiang, and Y. Wang, “Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains,” ISA Trans., vol. 99, pp. 488–495, Apr. 2020.
[13] H. Yuan and Y. Y. Tang, “Spectral–Spatial shared linear regression for hyperspectral image classification,” IEEE Trans. Cybern., vol. 47, no. 4, pp. 934–945, Apr. 2017.
[14] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, “Compression of hyperspectral remote sensing images by tensor approach,” Neurocomputing, vol. 147, pp. 358–363, Jan. 2015.
[15] B. Kumar and O. Diksht, “Spectral contextual classification of hyperspectral imagery with probablistic relaxation labeling,” IEEE Trans. Cybern., vol. 47, no. 12, pp. 4380–4391, Dec. 2017.
[16] Q. Zhang, X. Shen, L. Xu, and J. Jia (2015). Rolling Guidance Filter. [Online]. Available: http://www.cse. cuhk.edu.hk/leojia/projects/rollguidance
[17] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, “PCANet: A simple deep learning baseline for image classification?” IEEE Trans. Image Process., vol. 24, no. 12, pp. 5017–5032, Dec. 2015.
[18] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, “Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image,” IEEE Trans. Cybern., vol. 49, no. 7, pp. 2406–2419, Jul. 2019.
[19] Y. Wei, Y. Zhou, and H. Li, “Spectral-spatial response for hyperspectral image classification,” Remote Sens., vol. 9, no. 3, p. 203, Feb. 2017.
[20] Y. Zhou, J. Peng, and C. L. P. Chen, “Extreme learning machine with composite kernels for hyperspectral image classification,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 8, no. 6, pp. 2351–2360, Jun. 2015.
[21] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei, “Supervised spectral– spatial hyperspectral image classification with weighted Markov random fields,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 3, pp. 1490–1503, Mar. 2015.
[22] H. Wang, X. Lu, Z. Hu, and W. Zheng, “Fisher discriminant analysis with L1-norm,” IEEE Trans. Cybern., vol. 44, no. 6, pp. 828–842, Jun. 2014.
[23] S. T. Roweis, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[24] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” Neural Comput., vol. 15, no. 6, pp. 1373–1396, Jun. 2003.
[25] X. F. He, D. Cai, S. C. Yan, and H.-J. Zhang, “Neighborhood preserving embedding,” in Proc. 10th Int. Conf. Comput. Vis., Beijing, China, Oct. 2005, pp. 1208–1213.

[26] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng, and T. Yin, “Sparse-adaptive hypergraph discriminant analysis for hyperspectral image classification,” IEEE Geosci. Remote Sens. Lett., vol. 17, no. 6, pp. 1082–1086, Jun. 2020.

[27] F. Luo, L. Zhang, B. Du, and L. Zhang, “Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 8, pp. 5336–5353, Aug. 2020, doi: 10.1109/TGRS.2020.2963848.

[28] Z. Wang and B. He, “Locality perserving projections algorithm for hyperspectral image dimensionality reduction,” in Proc. 19th Int. Conf. Geoinformatics, Jun. 2011, pp. 1–4.

[29] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, “Local geometric structure feature for dimensionality reduction of hyperspectral imagery,” Remote Sens., vol. 9, no. 8, p. 790, Aug. 2017.

[30] M. Sugiyama, “Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis,” J. Mach. Learn. Res., vol. 8, pp. 1027–1061, May 2007.

[31] X. Kang, S. Li, and J. Atli Benediktsson, “Spectral–spatial hyperspectral image classification with edge-preserving filtering,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 5, pp. 2666–2677, May 2014.

[32] F. Cao, Z. Yang, J. Ren, W.-K. Ling, H. Zhao, M. Sun, and M. Sugiyama, “Dimensionality reduction of multimodal labeled data,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 5, pp. 2578–2592, May 2014.

[33] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng, and T. Yin, “Sparse-adaptive hypergraph discriminant learning for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 8, pp. 5336–5353, Aug. 2020, doi: 10.1109/TGRS.2020.2963848.

[34] Z. Feng, S. Yang, S. Wang, and L. Jiao, “Discriminative spectral–spatial margin-based semisupervised dimensionality reduction of hyperspectral data,” IEEE Geosci. Remote Sens. Lett., vol. 12, no. 2, pp. 224–228, Feb. 2015.

[35] Y. Zhou, J. Peng, and C. L. P. Chen, “Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 2, pp. 1082–1095, Feb. 2015.

[36] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations,” Exploring Artif. Intell. New Millennium, vol. 8, pp. 236–239, Jan. 2003.

[37] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” IEEE Trans. Inf. Theory, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[38] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 27:1–27:27, 2011.

[39] C. R. Hunter and K. Lange, “A tutorial on MM algorithms,” Amer. Statistician, vol. 58, no. 1, pp. 30–37, Feb. 2004.

[40] S. Cai, L. Zhang, W. Zuo, and X. Feng, “A probabilistic collaborative representation based approach for pattern classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2950–2959.

[41] J. Li, J. M. Bioucas-Dias, and A. Plaza, “Semisupervised hyperspectral image segmentation using multimodal logistic regression with active learning,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 11, pp. 4085–4098, Nov. 2010.

[42] U. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” Proc. IEEE, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[43] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 2, pp. 519–524, Apr. 2012.

[44] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in Proc. 30th ICML, 2013, pp. 1–9.

[45] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[46] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. 2009, pp. 2146–2153.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. NIPS, 2012, pp. 1097–1105.