


A Multi-Kernel Mode Using a Local Binary Pattern and Random Patch Convolution for Hyperspectral Image Classification

Wei Huang , Member, IEEE, Yao Huang, Zebin Wu, Senior Member, IEEE, Junru Yin, and Qiqiang Chen

Abstract—With the development of deep learning technology, more and more scholars have applied it to hyperspectral image (HSI) classification to improve classification accuracy. However, these deep-learning methods not only take a lot of time in the pre-training phase, but also have relatively limited classification performance when there are fewer labeled samples. In order to improve classification performance while reducing costs, this article proposes a multikernel method based on a local binary pattern and random patches (LBPRP-MK), which integrates a local binary pattern (LBP) and deep learning into a multiple-kernel framework. First, we use LBP and hierarchical convolutional neural networks to extract local textural features and multilayer convolutional features, respectively. The convolution kernel for the convolution operation is obtained from the original image using a random strategy without training. Then, we input local textural features, multilayer convolutional features, and spectral features obtained from the original image into the radial basis function to obtain three kernel functions. Finally, the three kernel functions are merged into a multikernel function according to their optimal weights under the composite kernel strategy. This multikernel function is used as the input for the support vector machine to obtain the classification result map. Experiments show that compared with other HSI classification methods, the proposed method achieves better classification performance on three HSI datasets.

Index Terms—Deep learning, hyperspectral image (HSI) classification, local binary pattern (LBP), multikernel mode, random patches.

I. INTRODUCTION

NOWADAYS, the hyperspectral images (HSIs) with high spectral resolution have attracted much attention in the field of remote sensing [1], [2]. Since these images have hundreds of continuous observation bands across the entire electromagnetic spectrum, more spectral information can be obtained when they are used. Therefore, they are widely used in

atmospheric detection [3], [4], environmental monitoring [5], [6], geological prospecting [7], [8], military reconnaissance [9], [10], and other fields. HSI classification [11]–[14] uses the rich information contained in HSI to assign unique category labels for each pixel, which is an important aspect of HSI applications. However, the well-known Hughes phenomenon [15] brings difficulties to HSI classification.

In order to solve this problem and obtain valuable spectral features, many scholars have researched dimensionality reduction [16]–[18] and band selection [19], [20], both of which have achieved good results. However, the pixel-by-pixel classifiers [21], [22] that use spectral information only have generally low classification accuracies due to the phenomenon of high intra-class spectrum variability and low interclass spectral variability in HSI [23]. Therefore, spatial features play an increasingly important role in the HSI classification. Pesaresi and Benediktsson [24] used a variety of morphological operations to extract the spatial information from images. Shen and Jia [25] designed a set of complex Gabor wavelets with different frequencies and directions to extract the signal variance in the spatial, spectral, and joint spatial-spectral domains. Huang *et al.* [26] used a gray-level co-occurrence matrix to extract the spatial texture information from HSIs. A local binary pattern (LBP) is the simplest and most effective feature description operator of the spatial feature extraction operators. Li *et al.* [27] used LBP for extracting local features from images. 3-D LBP (3-D dense LBP) [28] expands 2-D LBP to 3-D LBP to directly extract HSI features from the spatial-spectral information. In [29], the spatial features were extracted by LBP, then directly were stacked in a vector to realize the joint utilization of spatial-spectral information.

Although the method of manually extracting features has achieved good classification results, it lacks high robustness in the face of complex situations [30]. Therefore, many scholars employ various methods for automatically extracting features. Deep learning can solve this problem and is widely used in the HSI classification. For example, the earliest deep belief networks [31] and stacked autoencoders [32] have achieved good HSI classification results. However, these traditional deep-learning methods also have certain shortcomings. They need to convert 3-D data into 1-D data as input, thus losing a lot of spatial information. CNN takes the original image data as input and does not deform the image data in anyway, which can effectively solve this problem. Many excellent frameworks based on CNN

Manuscript received February 21, 2021; revised April 11, 2021; accepted April 21, 2021. Date of publication April 28, 2021; date of current version May 17, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61971233 and Grant 61872185, and in part by the Henan Province Science and Technology Breakthrough Project under Grant 212102210102 and Grant 212102210105. (Corresponding author: Wei Huang.)

Wei Huang, Yao Huang, Junru Yin, and Qiqiang Chen are with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China (e-mail: hnhw235@163.com; huangyao@zzuli.edu.cn; yinjr@zzuli.edu.cn; chenqq@zzuli.edu.cn).

Zebin Wu is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wuzb@njjust.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3076198

have emerged, for example, in [33], the original hyperspectral data is input into CNN to obtain the classification result graph. Han *et al.* used the two-stream convolutional network and spatial enhancement strategy to combine spatial-spectral features to complete the HSI classification [34]. In [35], a general multimodal deep learning framework was developed, and the five fusion architectures were further unified in this framework. The framework is not only limited to pixel classification tasks, but also suitable for spatial information modeling with CNN, successfully solving the limitations of single-modality applications in deep networks. Subsequently, multiscale densely connected network [36] and multiscale residual network [37] are used for HSI classification and have achieved great success. In addition to the above methods, there are some novel methods that use layering strategies. Gao *et al.* [36] proposed a PCANet classification method. This method uses PCA to learn convolution kernels from a set of patches and then utilizes a layered method to extract convolutional features. Finally, the extracted features are input into SVM to obtain the graph of the classification results. Unlike in [38], a random batch classification method [39] used a random strategy to obtain the convolution kernel in the original HSI without training, which can greatly save running time. To utilize the correlation between local spectral features, Sun *et al.* [40] proposed a random batch and local covariance classification framework that combined the covariance matrix with random patches network (RPNNet) on the basis of [39], thereby greatly improving the classification accuracy.

LBP is combined with CNN to obtain more representative spatial features due to its powerful textural feature extraction capability. However, carrying out the combination is a difficult problem. A kernel function uses simple linear weighting to effectively integrate spatial-spectral information. Camps-Valls *et al.* [41] proposed a composite kernel classification structure using two kernel functions to combine spatial-spectral information. Li *et al.* [42] proposed an AdaBoost framework with weighted ELM by combining composite kernels. To obtain the classification result map, Wang and Duan [43] integrated spatial, spectral, and hierarchical information into the SVM classifier in a multikernel manner. An ideal regularized discrimination multikernel subspace alignment method is proposed in [44] for HSI classification that uses both sample and label similarities to ensure that the generated kernel is very suitable for adaptive tasks. A multiple kernels-based SVM classification model is proposed in [45] that fuses the spatial, spectral, and semantic information in a HSI through a multikernel framework and achieves good classification results. An active multiple kernel Fredholm learning algorithm is proposed in [46] that effectively improves the performance of the classifier in the target domain. Therefore, we use a kernel function to combine CNN and LBP.

In order to alleviate the problem of time-consuming and insufficient labeling of samples in the HSI classification method based on deep learning, we combined the hierarchical deep CNN with LBP and considered a multikernel method based on LBP and random patches. The framework uses spectral features, local textural features, and multilayer convolutional features to complete the HSI classification. Compared with the traditional CNN classification method, the proposed method

can improve the classification performance while reducing the required training time for the CNN model and the number of labeled samples. More specifically, the main contributions of the proposed method in this article are as follows.

- 1) A classification model based on LBP and random patches is proposed. The proposed method takes less time compared with the existing classification frameworks.
- 2) The local textural features extracted by LBP, the multilayer convolutional features extracted by the layered CNN, and the spectral features obtained from the original HSI are input into the multikernel model. This framework effectively combines the textural features of HSI and the features of shallow and deep convolutions, which not only retains the advantages of layered CNN, but also requires fewer labeled samples.
- 3) The convolution kernel is obtained directly from the original HSI without training.

The remaining parts of this article are arranged as follows: Section II introduces the related article. The proposed method in this article is introduced in Section III. Section IV introduces the dataset used in the experiment and undertakes a comparison with the latest methods. Conclusion is drawn in Section V.

II. RELATED WORK

A. SVM Model and Kernel Functions

SVM is a linear classification model, its algorithm goal is to find a hyperplane to segment data points of different categories and maximize the minimum distance between the data points and the hyperplane. Given a set of training data $\{(x_1, y_1), \dots, (x_n, y_n)\}$, where $x_i \in R^N$ is the training data, and $y_i \in \{-1, 1\}$ is the labeled data, SVM mainly solves the following problems:

$$\min_{\omega, \xi_i, b} \left\{ \frac{1}{2} \omega^2 + C \sum_i \xi_i \right\}$$

$$s.t. y_i (\phi(x_i), \omega + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \dots, n, \quad (1)$$

where ω and b represent the linear classifier parameters in the feature space; C is a regularization parameter, which is used to control the classification performance and generalization ability of the classifier; ξ_i is used to deal with some allowable errors; and $\phi(\cdot)$ is a nonlinear mapping that can map pixels to high-dimensional space. For ease of calculation, the Lagrangian multiplier is used to transform the abovementioned original optimization problem into a Lagrangian dual problem, namely

$$\max \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i), \phi(x_j) \right\} \quad (2)$$

where α_i and α_j ($i \in \{1, \dots, n\}$, $j \in \{1, \dots, n\}$) are Lagrange multipliers. Since $\phi(x_i), \phi(x_j)$ requires a great deal of calculation, SVM uses the kernel method, which is a simpler mapping calculation, defined as follows:

$$K(x_i, x_j) = \phi(x_i), \phi(x_j). \quad (3)$$

Then, (3) is substituted into (2) to solve the dual problem and obtain the decision function of the test pixel x , i.e.,

$$f(x) = \sum_{i=1}^n \alpha_i y_i K(x_i, x) + b. \quad (4)$$

The SVM can obtain the inner product of the nonlinear transformation directly through the kernel function without complex operations. However, not all functions are regarded as kernel functions since a kernel function must satisfy Mercer's theorem [47] to be used in SVM. The radial basis function (RBF) kernel is widely used, which is expressed as follows:

$$K(x_i, x_j) = \exp\left(-\frac{x_i - x_j^2}{2\sigma^2}\right) \quad (5)$$

where σ is the width parameter of the kernel function. Let x^{Spec} and x^{Spat} denote the spectral feature and spatial feature of the HSI, respectively. The spectral kernel and spatial kernel are obtained when the spectral feature and spatial feature of the HSI are substituted into (5), i.e.,

$$K_{\text{Spec}}(x_i^{\text{Spec}}, x_j^{\text{Spec}}) = \exp\left(-x_i^{\text{Spec}} - x_j^{\text{Spec}^2}/2\sigma^2\right) \quad (6)$$

$$K_{\text{Spat}}(x_i^{\text{Spat}}, x_j^{\text{Spat}}) = \exp\left(-x_i^{\text{Spat}} - x_j^{\text{Spat}^2}/2\sigma^2\right). \quad (7)$$

A composite kernel function is proposed in [41] to combine the spectral kernel and the spatial kernel and is defined as follows:

$$K_{\text{CK}}(x_i, x_j) = \mu_1 K_{\text{Spec}}(x_i^{\text{Spec}}, x_j^{\text{Spec}}) + \mu_2 K_{\text{Spat}}(x_i^{\text{Spat}}, x_j^{\text{Spat}}) \quad (8)$$

where μ_1 and μ_2 ($\mu_1 + \mu_2 = 1$) are the weights used to balance the spectral kernel and the space kernel. In addition, multiple kernels are an extension of the composite kernel, which obtains the feature information from different angles. It is defined as follows:

$$\begin{aligned} K_{\text{MK}}(x_i, x_j) &= \mu_1 K_{\text{Spec}}(x_i^{\text{Spec}}, x_j^{\text{Spec}}) \\ &+ \mu_2 K_{\text{Spat}}(x_i^{\text{Spat}}, x_j^{\text{Spat}}) + \dots \\ &+ \mu_m K_{\text{Other}}(x_i^{\text{Other}}, x_j^{\text{Other}}) \\ \text{s.t. } \sum_{m=1}^M \mu_m &= 1. \end{aligned} \quad (9)$$

Based on the SVM classifier with multiple kernels, the spectral information, multilayer convolution information, and local texture information are combined to effectively improve the classification accuracy.

B. Local Binary Pattern

The LBP is a local textural feature descriptor with low computational complexity and strong descriptive ability that has been widely used in the fields of texture classification, image retrieval, and target retrieval. In view of the remarkable advantages of LBP,

many improved methods based on LBP have been proposed to solve various practical problems.

The basic idea of LBP is embodied in the two aspects "local" and "binary," where "local" refers to the local neighborhood of the central pixel, and "binary" expresses the relationship between the neighboring pixels and the central pixel. If the gray value of the neighboring pixel is greater than the gray value of the center pixel, the corresponding neighboring pixel is coded as 1; otherwise, it is coded as 0. The gray values of all neighboring pixels are quantized into a binary mode. Then, the resulting binary string is converted to decimal, and the LBP encoding value of the current center pixel is obtained. After calculating the LBP encoding values of all pixels, the frequency histogram of the LBP encoding values is used to represent image features.

Research shows that LBP can achieve a satisfactory classification effect on typical texture databases. Due to the gray level differences of the LBP encoded pixels, it is not affected by the monotonous gray level changes in the image. Mathematically, the values of circular neighborhood pixels and the bilinear interpolation method are used, with the center pixel (x, y) being expressed as follows:

$$\begin{aligned} \text{LBP}_{P,R}(x, y) &= \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \\ s(z) &= \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \end{aligned} \quad (10)$$

where P refers to the number of sampling points, R denotes the radius of the circle, and g_c and g_1, \dots, g_{P-1} represent the gray values of the center pixel and the pixels in the circular neighborhood, respectively.

Although the principle of the LBP operator algorithm is a simple and feasible method, there are still many problems, which can be enumerated as follows.

- 1) The dimensionality of the binary feature coding result is too high.
- 2) It does not adapt to changes in the image topology.
- 3) It is not robust for severe noise and illumination.
- 4) It does not consider the correlation between the local texture of the image and only obtains the textural features between the image neighborhoods; thus, it cannot effectively handle the larger and more complex textural features.
- 5) The stability of the LBP code is generally not good, and subsequent processing cannot be performed.

To solve the problems plaguing LBP, the uniform LBP (ULBP) method is proposed. It is an equivalent mode of LBP. When the cyclic binary number corresponding to a certain LBP has at most two transitions from 0 to 1 or from 1 to 0, the binary corresponding to the LBP is called an equivalent mode class. The specific mathematical descriptor is defined as follows:

$$\begin{aligned} \text{LBP}_{P,R}^{\text{riu}2}(x, y) &= \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U(\text{LBP}_{P,R}(x, y)) \leq 2 \\ P + 1, & \text{otherwise,} \end{cases} \end{aligned} \quad (11)$$

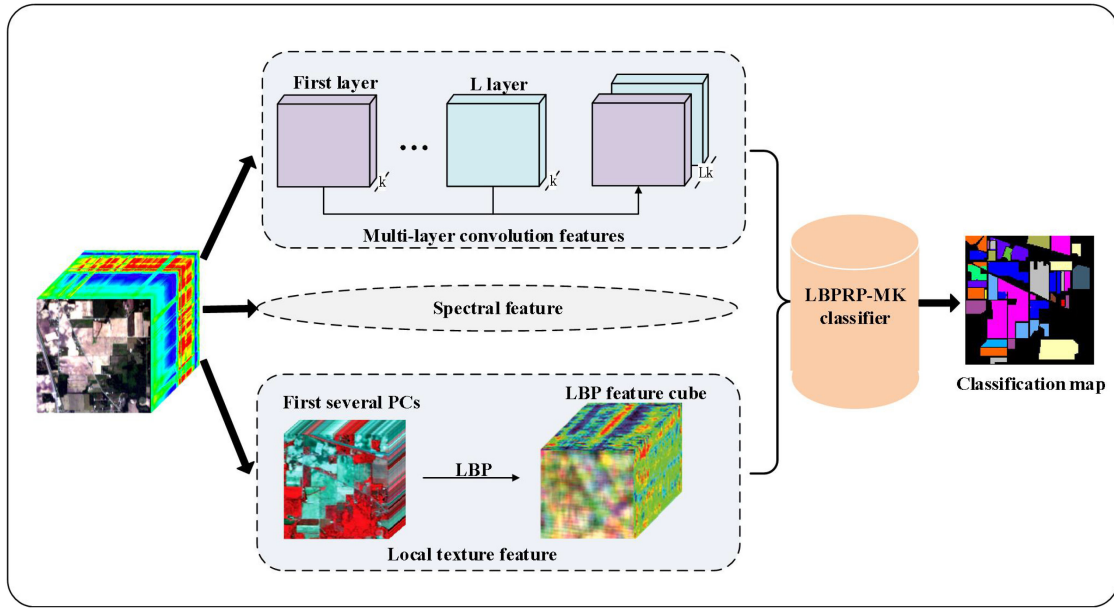


Fig. 1. Proposed LBPRP-MK framework for HSI classification.

where

$$U(\text{LBP}_{P,R}(x, y)) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (12)$$

where $s(\cdot)$ denotes the same function defined in (10).

C. PCA and Whitening

PCA is a data dimensionality reduction algorithm that can greatly improve the speed of unsupervised feature learning that is based on a linear transformation. It transforms the data into a new coordinate system so that the first large variance of any data projection is on the first coordinate (called the first principal component). The second-largest variance is on the second coordinate (the second principal component), etc. More importantly, the PCA algorithm is very helpful in whitening.

Whitening is an important preprocessing technique used in many algorithms to reduce the redundancy of the input data. In addition, the input data has the following properties after whitening.

- 1) The correlation between features is low. The eigenvectors of the first K eigenvalues are selected as the projection direction in PCA. If the size of K is the data dimension n , these K eigenvectors form the selection matrix $U = [u_1, u_2 \dots u_n]$ (u_1 is the main eigenvector, and it corresponds to the largest eigenvalue, u_2 is the secondary eigenvector, and so on). If $K < n$, that is the PCA dimensionality reduction; if $K = n$, the correlation between features is low.
- 2) All features have the same variance. Here, $\frac{1}{\sqrt{\lambda_i}}$ ($\lambda_1, \lambda_2 \dots, \lambda_n$ are the corresponding feature vectors) is used as the scaling factor to scale each feature $x_{\text{rot},i}$.

Then the formula $x_{\text{PCAwhite},i} = \frac{x_{\text{rot},i}}{\sqrt{\lambda_i}}$ is used, and the covariance matrix is the identity matrix I at this time.

PCA and whitening are used in the convolution process to reduce the time spent on the convolution operation in this article.

III. PROPOSED METHOD

In this article, we propose a LBPRP-MK method for HSI classification. First, a PCA transformation is performed on the original HSI to obtain the first few principal components. Then, the LBP operator and convolution operation are used on the first few PCs to obtain local textural features and multilayer convolutional features, respectively. Moreover, three single kernels are used to represent spectral features, local textural features, and multilayer convolutional features, which are then combined in a weighted way to obtain a multikernel function. Finally, the final classification result map is obtained by using the SVM classifier with multiple kernels. The flowchart of the proposed method is shown in Fig. 1.

A. Local Texture Feature Extraction

To reduce the redundant information in the spectral of the HSI, PCA is first used to select the spectral band subset with salient features. Then, the LBP operator is applied to each selected spectral band subset, and the LBP codes are generated by (10). The outputs of the LBP codes reflect the textural direction and smoothness of the local area (of size $w \times w$). After obtaining the LBP codes of all pixels, the histogram is calculated with the local patch centered on the pixel of interest, as shown in Fig. 2. Finally, all the bands of the LBP histogram are connected to form the local textural feature I_{LTF} .

As described in Section II-B, the dimensionality of the binary encoding result is too high, which will greatly increase the time cost. Therefore, ULBP is utilized to simplify the feature vector.

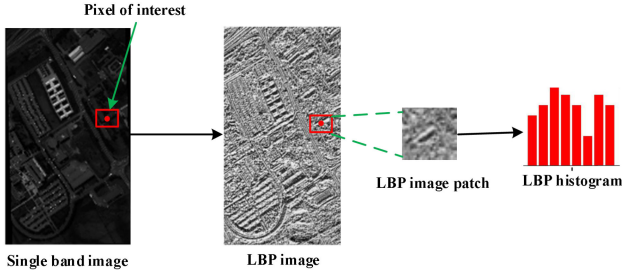


Fig. 2. LBP feature extraction of a single-band image.

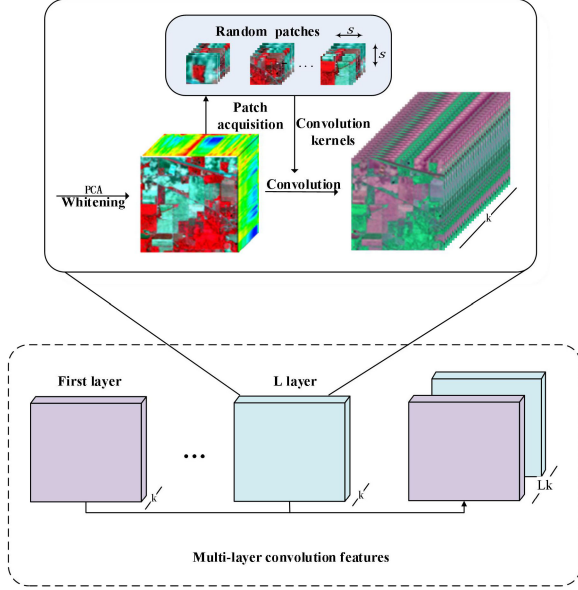


Fig. 3. Procedure used in multilayer convolutional feature extraction.

ULBP can reduce the number of modes from the original 2^p to $p(p-1)+2$, so that the feature vector of a spectral band reduces from 259 to 59. Thus, it creates feature vectors with fewer dimensions and can reduce the impact of high-frequency noise. It is worth noting that the patch size w is a user-defined parameter. In addition, the selection of the optimal patch size will be shown in the experimental section.

B. Multilayer Convolutional Feature Extraction

A cascade structure is utilized to combine the shallow and deep features obtained by the convolution kernels for determining the convolutional features. The convolution kernels are randomly selected from the input data during the convolution operation for each layer. Fig. 3 shows the procedure used in multilayer convolutional feature extraction. The specific steps are given in Algorithm 1.

1) *Random Patch Convolution*: In order to ensure that the selected random block is representative, we first use the K-means algorithm to select the random center, but the algorithm is too complex, so random projection is introduced to reduce the dimensionality. k pixels of the whitening data I_{Whiten} are randomly selected, a patch of size $s \times s \times p$ around each pixel is extracted to obtain a total of k patches. If the randomly

selected pixel is located at the edge of the image, the blank pixels around the pixel are filled by mirroring the image. The obtained k patches are used as the convolution kernels, which convolve the whitened data to obtain the k feature maps, denoted as $T \in R^{r \times c \times k}$. T_i represents the i th feature map, which is calculated as follows:

$$T_i = \sum_{j=1}^p I_{\text{Whiten}}^j * P_i^j, i = 1, 2, \dots, k \quad (13)$$

where I_{Whiten}^j denotes the j th dimension of I_{Whiten} , $P_i^j \in R^{s \times s \times p}$ is the j th dimension of the i th patch, and $*$ refers to 2-D convolution. The step size of the 2-D convolution operation is set to 1. The mirror filling method is still used to fill the blank areas around the pixels located at the edge of the image in the convolution operation.

2) *Nonlinear Activation*: $T \in R^{r \times c \times k}$ is transformed into the 2-D matrix $T \in R^{r \times c \times k}$, which represents the input of the rectified linear units (ReLU) used to obtain the first layer feature $Z^{(1)}$.

3) *Multilayer Convolutional Feature Extraction*: I is replaced with $Z^{(l)}$ ($l = 1, \dots, L$), and the above-mentioned first-layer feature extraction process is used to obtain features for different layers. The obtained convolutional features of each layer are superimposed together to obtain the multilayer convolutional feature I_{MCF} .

Algorithm 1: Multilayer Convolutional Feature Extraction.

Input: The original HSI I , an $L \times 1$ size cell I_{MCF} ;

For $l = 1, \dots, L$

Step 1: Perform PCA and whitening on I to obtain whitening data I_{Whiten} ;

Step 2: k random patches are extracted from the whitening data I_{Whiten} ;

Step 3: These k random patches convolve with the whitening data I_{Whiten} to obtain k feature maps;

Step 4: The activation function is applied to k feature maps in Step 3 to obtain the features of layer l , which are expressed as $Z^{(l)}$;

Step 5: Put $Z^{(l)}$ into I_{MCF} ;

Step 6: If $l < L$, matrix I is replaced by $Z^{(l)}$.

End

Output: Multilayer convolutional feature I_{MCF} .

To facilitate calculation, all layers use the same parameters, including the number of PCs p , the number of image patches k , and the patch size s . The selection of these parameters will be discussed in Section IV.

C. Spectral Feature Extraction

In the framework of LBPRP-MK, we take all the spectral pixels in the original HSI as spectral features and input them into (5) to obtain the spectral kernel function K_{Spec} , which is defined as follows:

$$K_{\text{Spec}}(x_i^{\text{Spec}}, x_j^{\text{Spec}}) = \exp\left(-x_i^{\text{Spec}} - x_j^{\text{Spec}^2}/2\sigma^2\right) \quad (14)$$

where x_i^{Spec} and x_j^{Spec} represent the spectral information of the i th pixel and the j th pixel in HSI, respectively.

D. LBPRP-MK

Three different types of features have been obtained, namely, the spectral features, local textural features, and multilayer convolutional features. To fuse these features, multikernel methods are utilized to construct a new HSI classification framework. Let $x_i \equiv \{x_i^{\text{SPE}}, x_i^{\text{LTF}}, x_i^{\text{MCF}}\}$, then the kernel function in the training phase is expressed as follows:

$$K_{\text{LBPRPMK}}^{\text{SPE-LTF-MCF}}(x_i, x_j) = \mu_{\text{SPE}} K_{\text{SPE}}(x_i^{\text{SPE}}, x_j^{\text{SPE}}) + \mu_{\text{LTF}} K_{\text{LTF}}(x_i^{\text{LTF}}, x_j^{\text{LTF}}) + \mu_{\text{MCF}} K_{\text{MCF}}(x_i^{\text{MCF}}, x_j^{\text{MCF}})$$

$$\text{s.t.}, \mu_{\text{SPE}} + \mu_{\text{LTF}} + \mu_{\text{MCF}} = 1 \quad (15)$$

where μ_{SPE} , μ_{LTF} and μ_{MCF} represent the contributions of the spectral features, local textural features, and multilayer convolutional features in the classification of HSI, respectively. $K_{\text{SPE}}(x_i^{\text{SPE}}, x_j^{\text{SPE}})$, $K_{\text{LTF}}(x_i^{\text{LTF}}, x_j^{\text{LTF}})$ and $K_{\text{MCF}}(x_i^{\text{MCF}}, x_j^{\text{MCF}})$ denote the corresponding kernel functions, which can be obtained with (5).

The main procedures of the proposed LBPRP-MK method are summarized in Algorithm 2.












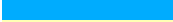



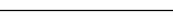
Algorithm 2: LBPRP- MK is Used in HSI Classification.

- Input:** Original HSI I , available training and validation samples;
- Step 1:** PCA is used to reduce the dimensionality of I to obtain the first p PCs, and the LBP operation is performed on the first p PCs to obtain the local textural feature I_{LTF} ;
- Step 2:** Perform Algorithm 1 to obtain the multilayer convolutional feature I_{MCF} ;
- Step 3:** The original data of all spectral bands in I is used as the spectral feature I_{SPE} ;
- Step 4:** A set of training data is randomly selected from the original HSI, the corresponding pixels are extracted from I_{SPE} , I_{LTF} and I_{MCF} according to the location of this set of training data to generate the corresponding three sets of training data;
- Step 5:** The three sets of training data from step 4 are input separately into (5) to obtain three kernel functions, which are the spectral kernel $K_{\text{SPE}}(x_i^{\text{SPE}}, x_j^{\text{SPE}})$, local textural kernel $K_{\text{LTF}}(x_i^{\text{LTF}}, x_j^{\text{LTF}})$, and multilayer convolutional kernel $K_{\text{MCF}}(x_i^{\text{MCF}}, x_j^{\text{MCF}})$;
- Step 6:** Apply the SVM classifier with the multiple kernels in (14) to obtain the classification result map.
- Output:** Classification result map.
-

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, in order to evaluate the performance of the proposed LBPRP-MK method, we conduct a series of experiments on three real datasets. In Section IV-A, we briefly introduce the

TABLE I
STATISTICS FOR THE INDIAN PINES DATASET

Name	Class	Samples		
		Train	Test	Total
Alfalfa		3	43	46
Corn-no till		72	1356	1428
Corn-min till		42	788	830
Corn		12	225	237
Grass/pasture		25	458	483
Grass/tree		37	693	730
Grass/pasture-mowed		2	26	28
Hay-windrowed		24	454	478
Oats		1	19	20
Soybeans-no till		49	923	972
Soybeans-min till		123	2332	2455
Soybeans-clean till		30	563	593
Wheat		11	194	205
Woods		64	1201	1265
Bldg-grass-tree-drives		20	366	386
Stone-steel towers		5	88	93
Total		520	9429	10249

three datasets. In Section IV-B, we discussed the parameters involved in the proposed method. In Section IV-C, we verified the effectiveness of the important parameters in the proposed method. In Section IV-D, we compare the proposed method with seven HSI classification methods on the three datasets, including two traditional classification methods, namely the spatial adaptive total variation method based on sparse multinomial logistic regression classifier (SMLR-SpATV) [48]; the extreme learning machine based on LBP (LBPELM) [27]; three kernel methods, namely, SVM composite kernel (SVMCK) [41], superpixel-based classification via multiple kernels (SC-MK) [49], and the adjacent superpixel-based multiscale spatial-spectral kernel method (ASMGSSK) [50]; two deep learning methods, namely, RpNet [39], and CNN and active learning with Markov random field (CNN-AL-MRF) [51].

All the experiments in this article are implemented in the MATLAB 2019 environment. The experiments use overall accuracy (OA), the evaluation indicators to evaluate the performances of the different classification models used in the experiment. The Kappa coefficient is an index to evaluate the classification performance. The larger its value, the higher the accuracy of the corresponding classification algorithm. Because it considers the number of pixels correctly and incorrectly classified, it can more comprehensively reflect the classification accuracy. All data are average values obtained from 10 run times.

A. Data Description

1) *Indian Pines*: The dataset was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor at the Indian pines test field in Indiana. The data contains 145×145 pixels and 16 classes. There are a total of 220 spectral bands, and the spectral bands cover a range from 0.4 to 2.5 μm . However, some of the bands cannot be reflected by water, so we eliminated these bands and used the remaining 200 bands for the research object. The specific land covers and training and test sets are given in Table I.

TABLE II
STATISTICS FOR THE PAVIA UNIVERSITY DATASET























Class		Samples		
Name	Color	Train	Test	Total
Asphalt		67	6564	6631
Meadows		186	18463	18649
Gravel		21	2078	2099
Trees		31	3033	3064
Metal sheets		14	1331	1345
Bare soil		50	4979	5029
Bitumen		13	1317	1330
Bricks		37	3645	3682
Shadows		10	937	947
Total		429	42347	42776

TABLE III
STATISTICS FOR THE KSC DATASET

Class		Samples		
Name	Color	Training	Test	Total
Scrub		39	722	761
Willow swamp		13	230	243
CP hammock		13	243	256
CP/Oak		13	239	252
Slash pine		9	152	161
Oak/Broadleaf		12	217	229
Hardwood swamp		6	99	105
Graminoid marsh		22	409	431
Spartina marsh		26	494	520
Cattail marsh		21	383	404
Salt marsh		21	398	419
Mud flat		26	477	503
Water		47	880	927
Total		268	4943	5211

2) *Pavia University*: The dataset was obtained by the reflection optical system imaging spectrometer sensor of the University of Pavia. The data contains 9 classes, consisting of 103 spectral bands of 610×340 pixels, with the spectral bands covering a range from 0.43 to $0.86 \mu\text{m}$. The specific land covers and training and test sets are given in Table II.

2) *KSC*: The dataset was obtained with the AVIRIS sensor above KSC in Florida and contains 13 classes. After removing the bands that cannot be reflected by water, there are still 176 spectral bands with 512×614 pixels, and the spectral bands cover a range from 0.4 to $2.5 \mu\text{m}$. The specific land covers and training and test sets are given in Table III.

B. Parameter Settings

In this section, we will discuss the parameter settings involved in LBPRP-MK. The kernel function used in this article is an RBF kernel, and the best kernel bandwidth is obtained through a five-fold cross-validation. On the basis of a large number of experiments, we have selected several parameters that have a greater impact on the experimental results for analysis, namely, patch size s , patch number k , number of PCs p , network depth l , LBP patch size w , spectral kernel weight μ_{SPE} , local texture kernel weight μ_{LTF} , and multilayer convolution kernel weight μ_{MCF} .

1) *Effects of Patch Size and Number*: In order to evaluate the impact of different s and k on classification performance in LBPRP-MK, we set l to 5, w to 21, and the three kernel weights to $\mu_{\text{SPE}} = 0.2$, $\mu_{\text{LTF}} = 0.5$, and $\mu_{\text{MCF}} = 0.3$, respectively. As shown in Fig. 4, with the increase in the number and size of the random patches, the classification accuracy did not always increase, which means that larger sizes and more patches cannot achieve the best classification performance. It can be seen from Fig. 4(a) that when $s = 25$ and $k = 28$, the Indian Pines dataset reaches the best OA. It can be seen from Fig. 4(b) that, unlike the Indian Pines dataset, when $s = 21$ and $k = 12$, the best OA is achieved for the Pavia University dataset. Similarly, it can be seen from Fig. 4(c) that the KSC dataset reaches its best classification performance when $s = 21$ and $k = 12$.

2) *Effect of the Number of PCs*: In order to ensure the uniformity of the experiment, all the PCA operations involved in this article use a uniform p value. As shown in Fig. 5, the number of PCs also affects the accuracy of the experiment. As p increases, the classification accuracy of the three datasets tends to rise and then decrease slightly. At the same time, for all three datasets, the computing time increases significantly. Considering the balance between classification accuracy and time consumption, we set p to 3.

3) *Effect of Network Depth*: In order to judge the influence of the network depth on the experimental results, we update the s , k , and p on the three datasets based on the above experiments, and the other parameter settings remain unchanged. Here, we set network depths from 1 to 10 to analyze the impact on the three datasets. As shown in Fig. 6, it can be observed that when l increases from 1 to 3, the OAs for the three datasets rise too. When l is between 3 and 6, the OAs for the three datasets begin to fluctuate. In the proposed LBPRP-MK, we set the network depth to 6. Although it may not be the best choice for all experimental datasets, a relatively small number is selected to the hardware allowance into consideration.

4) *Effect of LBP Patch Size*: In LBP feature extraction, w denotes patch size that is a very important parameter, and its size directly affects the classification performance. We set w to 7, 11, 15, 19, 23, 27, 31, 35, and 39 for updating parameter l . Meanwhile, the other parameter settings remain unchanged. It can be seen from Fig. 7 that as w increases from 7 to 27, the classification accuracy improves greatly. When w increases from 27 to 39, the classification accuracy decreases. Hence, image patches that are too large or too small are not conducive to the classification task. Patches that are too large may contain pixels from other classes, and features extracted from image patches that are too smalls may not represent the task. Therefore, in this article, we set the image patch size for LBP feature extraction to 27×27 pixels.

5) *Effect of Kernel Weight*: In order to verify the influence of different kernel weights on the experimental results, we first regard μ_{SPE} as a variable, as shown in Fig. 8(a). It can be seen that when μ_{SPE} is between 0.1 and 0.3, the classification accuracy increases. When μ_{SPE} is greater than 0.3, the OAs of the three datasets decrease slightly. When $\mu_{\text{SPE}} = 1$, only the spectral features are used, and the classification effects on the three datasets are very unsatisfactory, indicating that the spatial

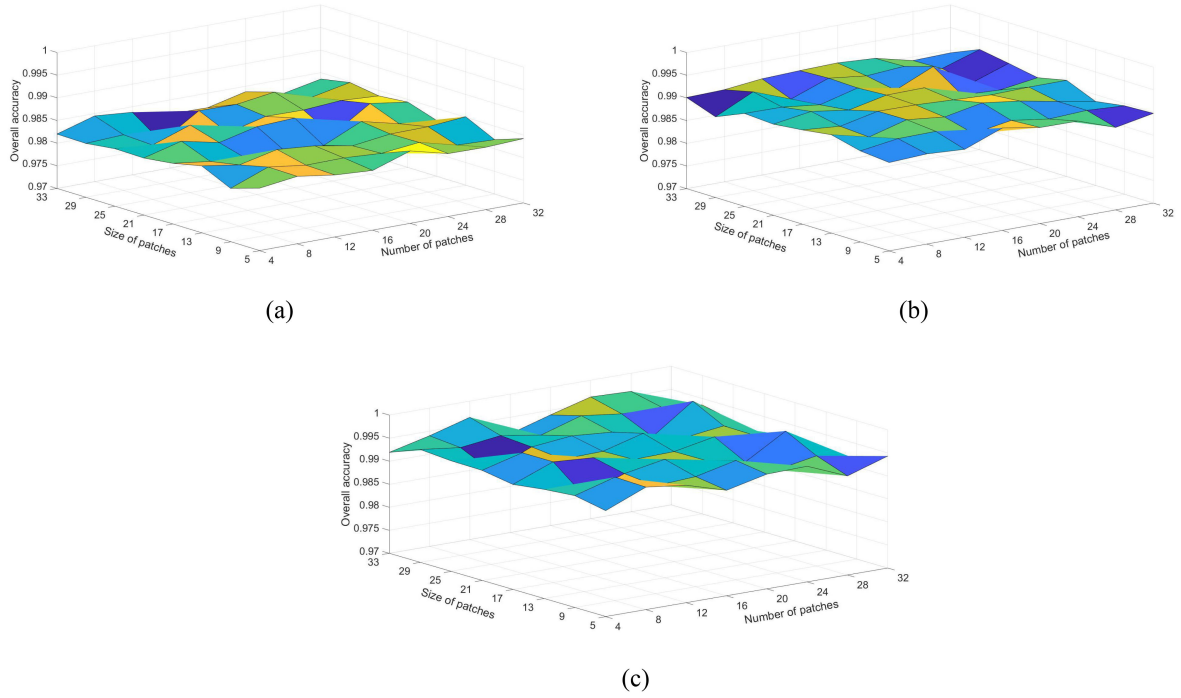


Fig. 4. Classification performance of the LBPRP-MK with different s and k values for the: (a) Indian Pines dataset; (b) Pavia University dataset; and (c) KSC dataset.

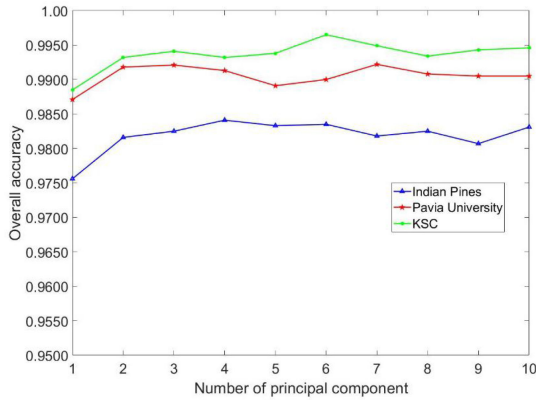


Fig. 5. Influence of the number of principal components.

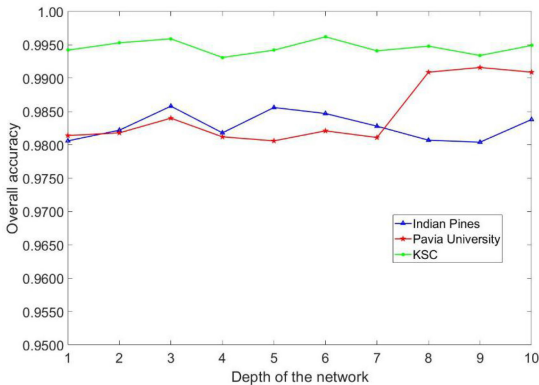


Fig. 6. Influence of network depth.

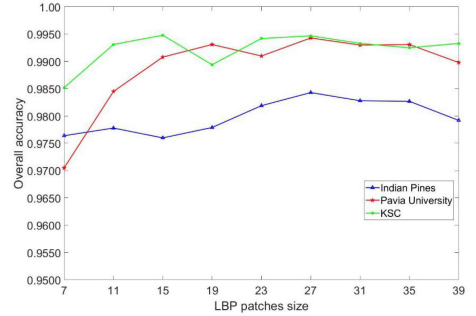


Fig. 7. Effect of LBP patch size.

features play a very important role in classification. Therefore, in this article, we set μ_{SPE} to 0.3. It can be seen from Fig. 8(b) that when the local texture kernel weight μ_{LTF} is less than 0.4 on the Pavia University dataset and KSC dataset, the OAs are increasing, and when μ_{LTF} is greater than 0.4, the OA values begin to decrease slightly. Similarly, the best μ_{LTF} for the Indian Pines dataset is 0.5. Therefore, we set μ_{LTF} to 0.4 and μ_{MCF} to 0.3 for the Pavia University and KSC datasets. For the Indian Pines dataset, we set μ_{MCF} to 0.5, and μ_{MCF} to 0.2.

C. Effectiveness Verification

In this section, we conducted an experiment to evaluate the effectiveness of each technology, including the LBP, multilayer CNN, and multiple kernels. In this experiment, we combine these technologies in turn and propose the following methods.

- 1) *RP-CK*: Combination of CNN and CK.

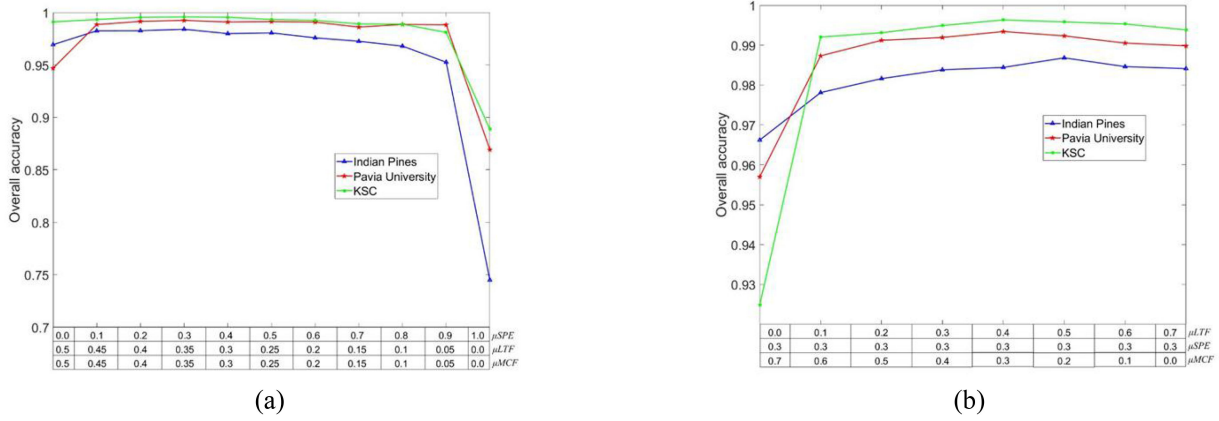


Fig. 8. Impact of different kernel weights.

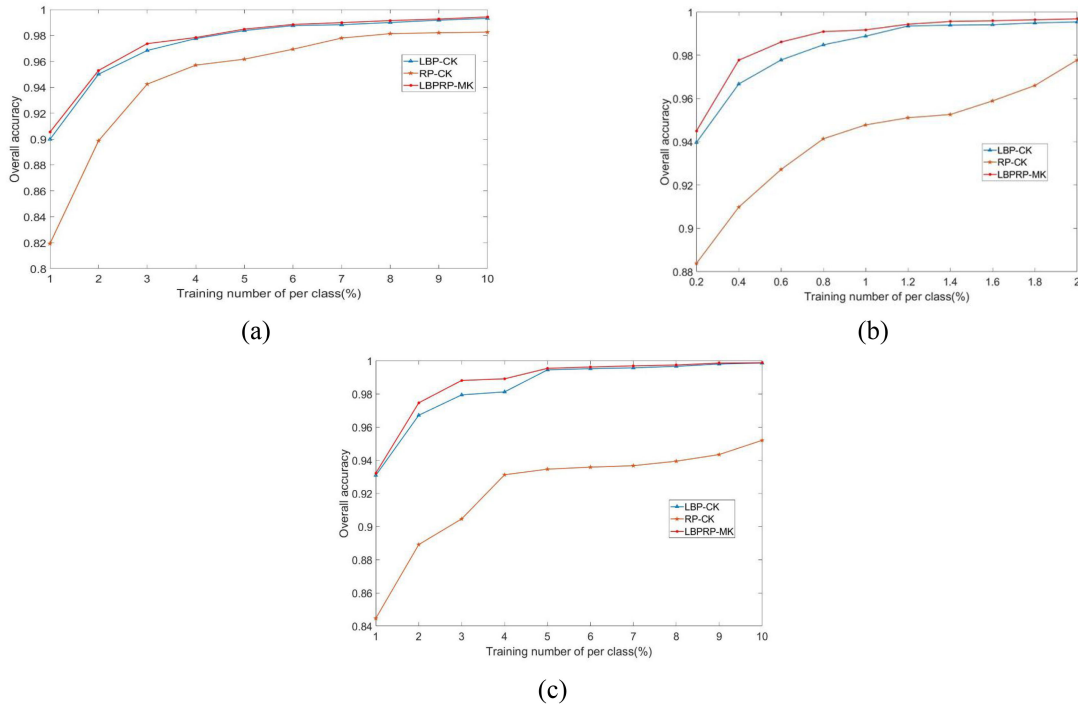


Fig. 9. Effect of training number on the: (a) Indian Pines dataset; (b) Pavia University dataset; and (c) KSC dataset.

- 2) *LBPRP-MK*: Combination of LBP and CK.
- 3) *LBPRP-MK*: The proposed method in this article.

In order to compare the classification performances of these methods, we use the sample number as a variable. Note that in order to ensure the unity of the experiment, the parameters p, s, k, l , and μ_{SPE} in RP-CK are all the same as those in LBPRP-MK, and the parameters p, ω , and μ_{SPE} in LBP-CK are also the same as those in LBPRP-MK. In Section IV-B, we verified through experiments that when the spectral kernel weight μ_{SPE} is 0.3, the classification effect is the best, so in this section, we set μ_{SPE} to 0.3. Since there is no local texture kernel in RP-CK and no multilayer convolution kernel in LBP-CK, and $\mu_{SPE} + \mu_{LTF} + \mu_{MCF} = 1$, the weight of the other kernel is set to 0.7.

It can be seen from Fig. 9 that the classification accuracies of RP-CK, LBP-CK, and LBPRP-MK increase with the increase in the number of training samples. Among them, the classification performance of LBPRP-MK is always better than those of the other two methods. Although the OA of LBP-CK is very close to LBPRP-MK, it consumes more time. Meanwhile, its classification accuracy is not as good as that of LBPRP-MK when there are fewer training samples. It can be seen from Fig. 9 that for the Indian Pines and KSC datasets, when the number of training samples reaches 5% of the total number of samples, the classification accuracy tends to stabilize. Therefore, we can conclude that 5% training samples are sufficient for learning the representative models of the datasets. For the Pavia University dataset, when the number of training samples reaches 1% of

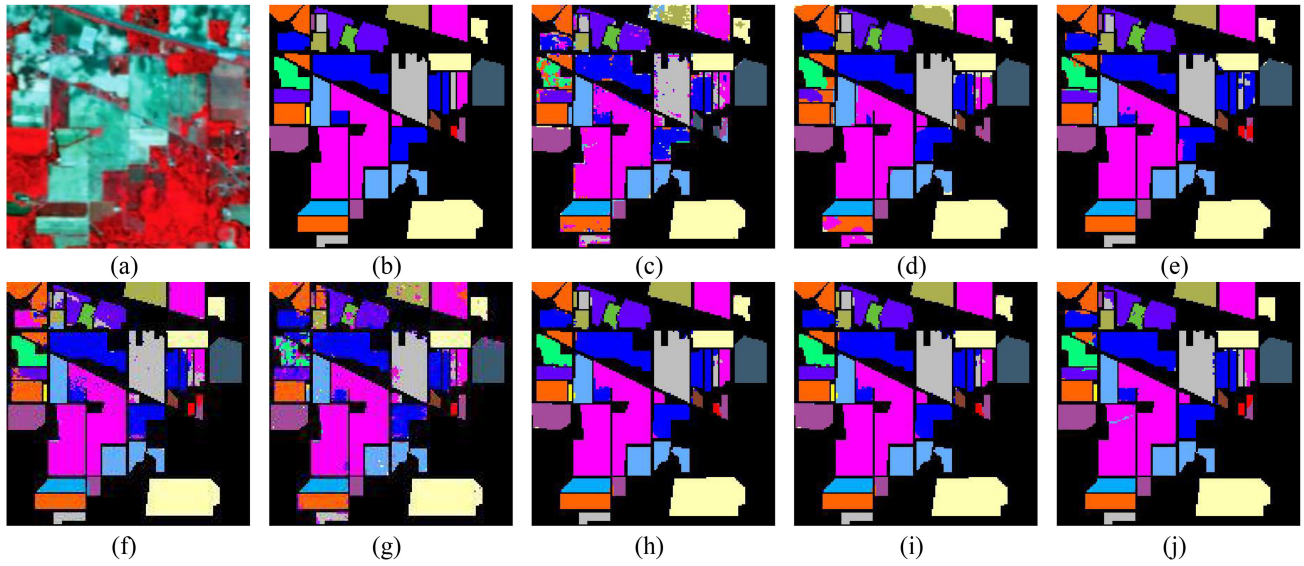


Fig. 10. Indian Pines data classification. (a) False color maps. (b) Real feature marker. (c) SVM composite kernel. (d) Sparse multinomial logistic regression classifier. (e) Learning machine based on LBP. (f) Superpixel-based classification via multiple kernel. (g) Random patches network. (h) Adjacent superpixel-based multiscale spatial-spectral kernel method. (i) CNN and active learning with Markov random field. (j) LBPRP-MK.

TABLE IV
CLASSIFICATION RESULTS FOR INDIAN PINES DATA

Class name	SVMCK	SMLR-SpATV	LBPELM	SC_MK	RPNet	ASMGSSK	CNN-AL-MRF	LBPRP-MK
Alfalfa	0.0980	0.7529	1.0000	0.9767	0.7000	0.9118	0.9271	0.9791
Corn-no till	0.8747	0.9259	0.9527	0.9485	0.9291	0.9753	0.9298	0.9747
Corn-min till	0.8971	0.8547	0.9809	0.9501	0.8906	0.9864	0.8871	0.9737
Corn	0.6041	0.9315	0.9261	0.9662	0.6471	0.9446	0.9770	0.9836
Grass/pasture	0.9337	0.8932	0.9282	0.9118	0.9159	0.9722	0.9290	0.9729
Grass/tree	0.9713	0.9848	0.9821	0.9977	0.9726	0.9893	0.9889	0.9980
Grass/pasture-mowed	0.0000	0.2250	0.9667	0.9423	0.8385	0.9458	0.7674	0.9769
Hay-windrowed	0.9772	0.9998	0.9935	0.9941	0.9685	0.9981	0.9787	1.0000
Oats	0.0000	0.0000	0.8000	1.0000	0.3526	0.9579	0.3889	0.7737
Soybeans-no till	0.8088	0.9017	0.9536	0.9023	0.9042	0.9622	0.9227	0.9703
Soybeans-min till	0.9241	0.9732	0.9845	0.9614	0.9702	0.9846	0.9507	0.9911
Soybeans-clean till	0.8502	0.9096	0.9637	0.8915	0.7853	0.9839	0.9051	0.9799
Wheat	0.9831	0.9950	0.9900	0.9706	0.9825	0.9886	0.9663	0.9948
Woods	0.9700	0.9970	0.9969	0.9998	0.9848	0.9965	0.9928	1.0000
Bldg-grass-tree-drives	0.6761	0.8925	0.9872	0.9358	0.8989	0.9850	0.8840	0.9967
Stone-steel towers	0.8789	0.7922	0.9667	0.9898	0.6705	0.9267	0.9712	0.9648
mean_OA	0.8882	0.9380	0.9732	0.9552	0.9250	0.9813	0.9428	0.9853
STD_OA	0.0066	0.0096	0.0047	0.0088	0.0038	0.0038	0.0031	0.0030
mean_kappa	0.8723	0.9291	0.9695	0.9489	0.9140	0.9787	0.9379	0.9832
STD_kappa	0.0074	0.0110	0.0053	0.0100	0.0043	0.0044	0.0042	0.0040

the total number of samples, the classification accuracy tends to stabilize, so 1% training samples are used for the experiments on this dataset. It can also be seen from Fig. 9 that even if the training samples are only 1% on the Indian Pines dataset and the KSC dataset, the overall classification effect can reach more than 90%. On the Pavia University dataset, only 0.2% of the training samples can be achieved. Classification accuracy of more than 90%. It proves the effectiveness and robustness of the proposed method under the condition of small samples.

D. Comparison With Other State-of-the-Art Methods

To evaluate the classification accuracy of the proposed LBPRP-MK method, we introduce some the latest HSI

classification methods for comparison, including SVMCK, SMLR-SpATV, LBPELM, SC_MK, RPNet, ASMGSSK, and CNN-AL-MRF. The classification maps of different methods are shown in Figs. 10–12 and Tables IV and VI shows the quantitative assessment.

1) *Indian Pines Dataset*: We randomly select 5% of the data in each class as training samples, resulting in 520 training samples in total, and the remaining samples are used for testing in Indian Pines dataset. Table IV gives the experimental results of these methods. It can be seen that the STD_OA of the LBPRP-MK method is 0.3% and the mean_OA is 98.53%. In these methods, the mean_OA of the SVMCK method is the smallest, but the STD_OA of SMLR-SpATV method is the most unstable. Compared with the OA of the ASMGSSK method with the

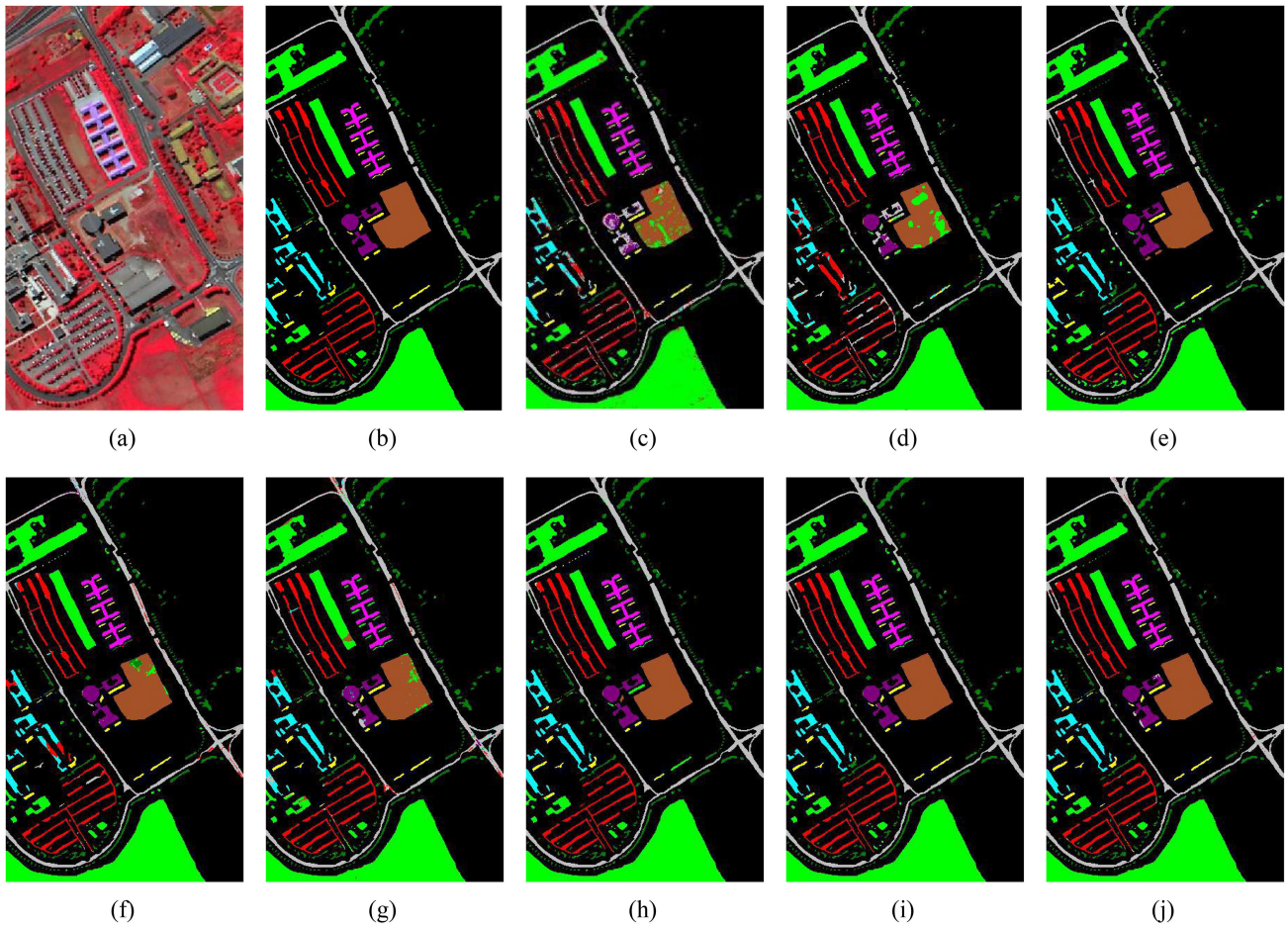


Fig. 11. Pavia University data classification. (a) False color maps. (b)Real feature marker. (c) SVM composite kernel. (d) Sparse multinomial logistic regression classifier. (e) Learning machine based on LBP. (f) Superpixel-based classification via multiple kernel. (g) Random patches network. (h) Adjacent superpixel-based multiscale spatial-spectral kernel method. (i) CNN and active learning with Markov random field. (j) LBPRP-MK.

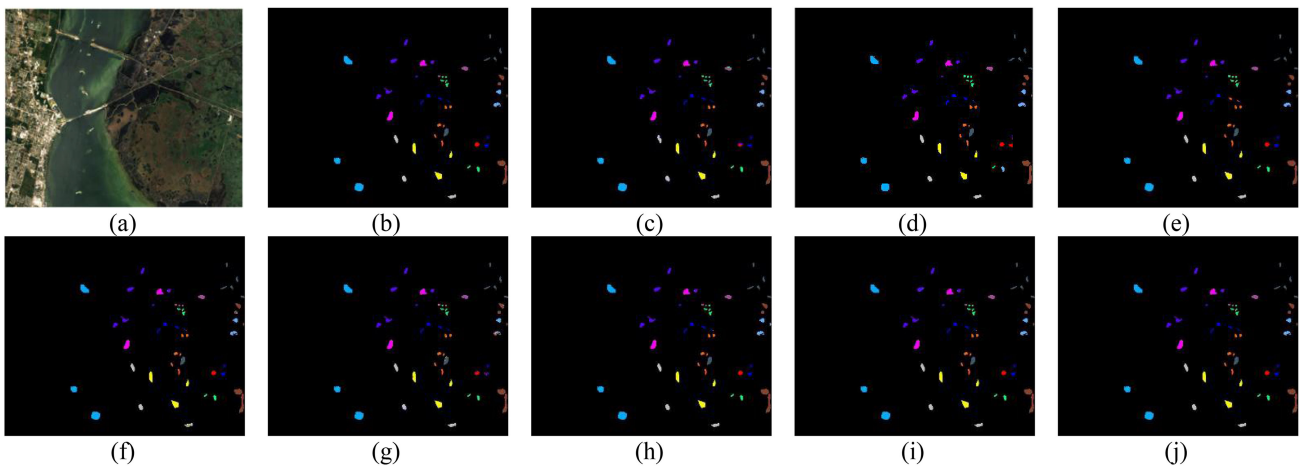


Fig. 12. KSC data classification. (a) False color maps. (b) Real feature marker. (c) SVM composite kernel. (d) Sparse multinomial logistic regression classifier. (e) Learning machine based on LBP. (f) Superpixel-based classification via multiple kernel. (g) Random patches network. (h) Adjacent superpixel-based multiscale spatial-spectral kernel method. (i) CNN and active learning with Markov random field. (j) LBPRP-MK.

TABLE V
CLASSIFICATION RESULTS FOR PAVIA UNIVERSITY DATA

Class name	SVMCK	SMLR-SpATV	LBPELM	SC_MK	RNet	ASMGSSK	CNN-AL-MRF	LBPRP-MK
Asphalt	0.9470	0.9770	0.9786	0.9491	0.9384	0.9920	0.9890	0.9902
Meadows	0.9783	0.9980	0.9957	0.9953	0.9905	0.9968	0.9991	0.9996
Gravel	0.7198	0.7323	0.9368	0.8820	0.9006	0.9978	0.8556	0.9958
Trees	0.9202	0.8669	0.6952	0.9643	0.9404	0.9309	0.9649	0.9600
Metal sheets	0.9913	0.9996	0.9494	0.9962	0.9285	0.9871	0.9990	0.9996
Bare soil	0.8620	0.8572	0.9974	0.9610	0.9299	0.9999	0.9725	1.0000
Bitumen	0.7942	0.7159	0.9969	0.9536	0.8828	0.9903	0.9604	0.9719
Bricks	0.8635	0.9089	0.9537	0.9348	0.9575	0.9835	0.9870	0.9897
Shadows	0.9689	0.0702	0.3866	0.9101	0.9012	0.8380	0.9487	0.9939
mean_OA	0.9275	0.9188	0.9503	0.9680	0.9572	0.9866	0.9817	0.9933
STD_OA	0.0053	0.0124	0.0080	0.0062	0.0061	0.0039	0.0019	0.0011
mean_kappa	0.9034	0.8908	0.9334	0.9575	0.9300	0.9822	0.9840	0.9911
STD_kappa	0.0072	0.0170	0.0107	0.0083	0.0141	0.0052	0.0028	0.0015

TABLE VI
CLASSIFICATION RESULTS FOR KSC DATA

Class name	SVMCK	SMLR-SpATV	LBPELM	SC_MK	RNet	ASMGSSK	CNN-AL-MRF	LBPRP-MK
Scrub	0.9871	1.0000	0.9934	0.9745	0.9422	0.9934	0.9900	0.9823
Willow swamp	0.9278	0.9287	0.9652	0.9113	0.8630	0.9930	0.9696	0.9913
CP hammock	0.9424	0.9757	0.9679	0.9815	0.9144	0.9963	0.9959	0.9971
CP/Oak	0.8063	0.8531	0.9849	0.9285	0.8192	0.9816	0.9669	0.9649
Slash pine	0.4520	0.9461	0.8388	0.9322	0.8697	0.9033	0.9809	0.9605
Oak/Broadleaf	0.8461	0.9853	0.9788	0.9014	0.6101	0.9995	0.9995	1.0000
Hardwood swamp	0.8535	0.9384	1.0000	1.0000	0.7687	1.0000	1.0000	1.0000
Graminoid marsh	0.9450	0.9914	1.0000	0.9848	0.9139	1.0000	0.9924	0.9892
Spartina marsh	1.0000	1.0000	0.9949	0.9943	0.9468	1.0000	0.9830	1.0000
Cattail marsh	0.9775	1.0000	1.0000	0.9590	0.9736	1.0000	1.0000	1.0000
Salt marsh	0.9641	0.9887	0.9706	0.9807	0.9892	0.9902	0.9902	0.9985
Mud flats	0.9413	0.9920	0.9656	0.9857	0.9824	0.9667	0.9686	0.9998
Water	0.9997	1.0000	1.0000	1.0000	0.9981	1.0000	1.0000	1.0000
mean_OA	0.9411	0.9825	0.9830	0.9734	0.9291	0.9906	0.9886	0.9929
STD_OA	0.0111	0.0103	0.0065	0.0071	0.0124	0.0049	0.0048	0.0042
mean_kappa	0.9344	0.9805	0.9811	0.9642	0.9210	0.9896	0.9873	0.9921
STD_kappa	0.0123	0.0114	0.0072	0.0095	0.0138	0.0065	0.0058	0.0053

highest accuracy in comparison algorithms, that of the LBPRP-MK method is increased by 0.4%. In addition, the remaining methods are less than 1.21%–9.71%. Similarity, the STD_kappa of the LBPRP-MK method is the most stable and the mean_kappa is the highest. It is observed from Table IV that the mean_kappa of the proposed method is more than 0.45%–11.09% compared with comparison methods. Thus, the proposed LBPRP-MK method is superior to the other classification methods in terms of OA and the kappa coefficient. Fig. 10 shows the classification results of the proposed method in Indian Pines dataset. The proposed method yields the best performance in these case. It can be seen from Fig. 10 that the classification results of the proposed LBPRP-MK model have the least number of misclassifications, the overall is smoother and there are only a few noise points, which is closer to the ground truth map.

2) *Pavia University Dataset*: We randomly select 1% of the data in each category as the training sample, so the number of training samples is 429, and the remaining 42347 samples are used for testing. Table V gives the experimental results of these methods. It can be seen that the OA of the LBPRP-MK method is the most stable and the mean_OA is the highest. Compared with SMLR-SpATV method, the OA of the LBPRP-MK method is increased by 7.45%. Even if it is the ASMGSSK method with the highest accuracy among these methods, the OA of

the proposed method is increased by 0.67%. In addition, the remaining methods are less than 1.16%–6.58%. Similarity, the STD_kappa of the LBPRP-MK method is the most stable and the mean_kappa is the highest. Therefore, the proposed LBPRP-MK method is superior to the other classification methods in terms of OA and the kappa coefficient. Fig. 11 shows a classification diagram of the Pavia University dataset. It can be seen that the proposed method shows excellent performance in terms of the edge and adjacent-area pixel classifications.

3) *KSC Dataset*: We randomly select 5% of the data in each category as training samples, and the remaining 95% are used for testing. Table VI gives the classification accuracies of the different methods for the KSC dataset. Compared with RNet method with the highest accuracy among these methods, the mean_OA of the LBPRP-MK method is increased by 6.38%. Compared with the RNet method, the mean_kappa of the LBPRP-MK method is increased by 7.11%. Additionally, the LBPRP-MK method is very stable. The STD_OA and STD_kappa are 0.42% and 0.53%, respectively. According to above analyses, the proposed LBPRP-MK method have good classification performance compared with other classification methods. It can be seen from Fig. 12 that the various classes of the KSC dataset are relatively scattered, and there is no difference directly.

TABLE VII
TIME COSTS FOR THE THREE DATASETS

Method	SVMCK	SMLR-SpATV	LBPELM	SC-MK	RPNNet	ASMGSSK	CNN-AL-MRF	LBPRP-MK
Average	1.1	13	5.7	2.1	3.4	10.5	8109.3	3.5
Time(s)								
Indian Pines	1.1	13	5.7	2.1	3.4	10.5	8109.3	3.5
Pavia University	5.4	61.6	55.5	7.3	11.8	43.5	2001.1	14.8
KSC	7.8	203.4	118.3	13.3	16	62.6	9142.6	14.2

E. Time cost

Table VII gives the average running-time consumption of 10 ten Monte-Carlo runs, which includes the training time and test time, and several existing HSI classification methods for the three datasets. Compared with the traditional kernel classification method SVMCK, superpixel-based kernel methods, such as SC-MK and ASMGSSK, take longer. The time cost of the deep learning-based CNN-AL-MRF method is many times greater than the time cost of RPNNet without a pre-training stage. The proposed method combines deep learning with the kernel method, and its time cost falls between the time costs for the two methods, nonetheless, its classification effect is the best among the methods investigated.

V. CONCLUSION AND FUTURE WORK

In this article, we use the kernel function to combine LBP and CNN and propose a multikernel mode using a LBP and random patches convolution for HSI classification. Specifically, in order to make up for the blindness of CNN in extracting features, we use textural features as its extension and use kernel functions to combine them to complete the HSI classification. In addition, we use a random strategy to select the convolutional kernel without any training, which makes up for the shortcomings of deep learning in terms of requiring a large number of training samples, and achieves a good classification effect.

All the parameters used in this article are adjusted manually. In the future, we will study a method for automatically selecting the best parameters and reducing the time cost.

REFERENCES

- [1] L. Gao, D. Hong, J. Yao, B. Zhang, P. Gamba, and J. Chanussot, "Spectral superresolution of multispectral imagery with joint sparse and low-rank learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2269–2280, Mar. 2021.
- [2] K. Zheng *et al.*, "Coupled convolutional neural network with adaptive response function learning for unsupervised hyperspectral super resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2487–2502, Mar. 2021.
- [3] W. J. Blackwell *et al.*, "Hyperspectral microwave atmospheric sounding," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 128–142, Jan. 2011.
- [4] J. Satapathy, "Extreme cyclonic storm monitoring using INSAT-3D/3DR -hyperspectral sounder observations," *Remote Sens. Appl., Soc. Environ.*, vol. 19, , 2020, Art. no. 100339.
- [5] S. Kendler, I. Ron, S. Cohen, R. Raich, Z. Mano, and B. Fishbain, "Detection and identification of sub-millimeter films of organic compounds on environmental surfaces using short-wave infrared hyperspectral imaging: Algorithm development using a synthetic set of targets," *IEEE Sensors J.*, vol. 19, no. 7, pp. 2657–2664, Apr. 2019.
- [6] D. Zeng, S. Zhang, F. Chen, and Y. Wang, "Multi-Scale CNN based garbage detection of airborne hyperspectral data," *IEEE Access*, vol. 7, pp. 104514–104527, Jul. 2019.
- [7] K. Tan and J. Qiao, "Development history and prospect of remote sensing technology in coal geology of china," *Int. J. Coal Sci. Technol.*, vol. 7, no. 2, pp. 311–319, 2020.
- [8] I. M. Aleshin *et al.*, "Online publication of tiltmeter data based on the seedlink protocol," *Seismic Instruments*, vol. 56, no. 5, pp. 509–515, 2020.
- [9] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geosci. Rem. Sen. Mag.*, vol. 7, no. 2, pp. 101–117, Jun. 2019.
- [10] L. Luft, N. Carsten, F. Matthias, B. Niels, and J. Florian, "Hyperspectral modeling of ecological indicators – A new approach for monitoring former military training areas," *Ecol. Indicators.*, vol. 46, pp. 264–285, Nov. 2014.
- [11] B. Tu, C. Zhou, D. He, S. Huang, and A. Plaza, "Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4116–4131, Jun. 2020.
- [12] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1424–1436, Feb. 2020.
- [13] 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.
- [14] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
- [15] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theor.*, vol. 14, no. 1, pp. 55–63, Jan. 1968.
- [16] J. C. Harsanyi, and C. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.
- [17] J. An, X. Zhang, H. Zhou, and L. Jiao, "Tensor-Based low-rank graph with multimanifold regularization for dimensionality reduction of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4731–4746, Aug. 2018.
- [18] Y. Duan, H. Huang, and Y. Tang, "Local constraint-based sparse manifold hypergraph learning for dimensionality reduction of hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 613–628, Jan. 2020.
- [19] A. Martínez-Usó, Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-Based hyperspectral band selection using information measures," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007.
- [20] P. Ribalta Lorenzo, L. Tulczyjew, M. Marcinkiewicz, and J. Nalepa, "Hyperspectral band selection using attention-based convolutional neural networks," *IEEE Access*, vol. 8, pp. 42384–42403, 2020.
- [21] X. Jia and J. A. Richards, "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 538–542, Jan. 1999.
- [22] V. Haertel and D. A. Langrebe, "On the classification of classes with nearly equal spectral response in remote sensing hyperspectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 5, pp. 2374–2386, Sep. 1999.
- [23] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.
- [24] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, Feb. 2001.
- [25] L. Shen and S. Jia, "Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5039–5046, Dec. 2011.
- [26] X. Huang, X. Liu, and L. Zhang, "A multichannel gray level co-occurrence matrix for multi/hyperspectral image texture representation," *Remote Sens.*, vol. 6, no. 9, pp. 8424–8445, 2014.
- [27] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

- [28] S. Jia, J. Hu, J. Zhu, X. Jia, and Q. Li, "Three-dimensional local binary patterns for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 2399–2413, Apr. 2017.
- [29] F. Gao, Q. Wang, J. Dong, and Q. Xu, "Spectral and spatial classification of hyperspectral images based on random multi-graphs," *Remote Sens.*, vol. 10, no. 8, pp. 1271, 2018.
- [30] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Learning and transferring deep joint spectral–spatial features for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4729–4742, Aug. 2017.
- [31] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks." in *Proc. Adv. Neural Inf. Process Syst.*, 2006, pp. 153–160.
- [32] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural. Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [33] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: [10.1109/TGRS.2020.3015157](https://doi.org/10.1109/TGRS.2020.3015157).
- [34] M. Han, R. Cong, X. Li, H. Fu, and J. Lei, "Joint spatial-spectral hyperspectral image classification based on convolutional neural network," *Pattern Recognit. Lett.*, vol. 130, pp. 38–45, 2020.
- [35] D. Hong *et al.*, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 1–15, Aug. 2020.
- [36] H. Gao, Y. Miao, X. Cao, and C. Li., "Densely connected Multi-scale attention network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2563–2576, 2021.
- [37] H. Gao, Y. Yang, C. Li, L. Gao, and B. Zhang. "Multiscale residual network with mixed depthwise convolution for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3396–3408, Apr. 2021.
- [38] T. 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.
- [39] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogramm.*, vol. 142, pp. 344–357, 2018.
- [40] Y. Sun, Z. Fu, and L. Fan, "A novel hyperspectral image classification pattern using random patches convolution and local covariance," *Remote Sens.*, vol. 11, no. 16, pp. 1954, 2019.
- [41] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla. "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.
- [42] L. Li, C. Wang, W. Li, and J. Chen, "Hyperspectral image classification by adaboost weighted composite kernel extreme learning machines." *Neurocomputing*, vol. 275, pp. 1725–1733, Sep. 2018.
- [43] Y. Wang and H. Duan, "Classification of hyperspectral images by SVM using a composite kernel by employing spectral, spatial and hierarchical structure information," *Remote Sens.*, vol. 10, no. 3, 2018.
- [44] W. Yang, J. Peng, and W. Sun, "Ideal regularized discriminative multiple kernel subspace alignment for domain adaptation in hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5833–5846, 2020.
- [45] Y. Wang, W. Yu, and Z. Fang, "Multiple kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information," *Remote Sens.*, vol. 12, no. 1, 2020.
- [46] A. Saboori, H. Ghassemian, and F. Razzazi, "Active multiple kernel fredholm learning for hyperspectral images classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 2, pp. 356–360, Feb. 2020, doi: [10.1109/LGRS.2020.2969970](https://doi.org/10.1109/LGRS.2020.2969970).
- [47] C. Carmeli, E. D. Vito, and A. Toigo, "Vector valued reproducing kernel Hilbert spaces of integrable functions and mercer theorem," *Anal. Appl.*, vol. 4, no. 4, pp. 377–408, 2006.
- [48] 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.
- [49] L. Fang, S. Li, and W. Duan, "Classification of hyperspectral images by exploiting spectral–spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.
- [50] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, "Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1905–1919, Jun. 2019.
- [51] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4604–4616, Jul. 2020.



Wei Huang (Member, IEEE) was born in Henan, China, in 1982. He received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, China, in 2015.

He is currently an Associate Professor with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. His current research interests include pan-sharpening, hyperspectral classification, image processing, and machine learning.



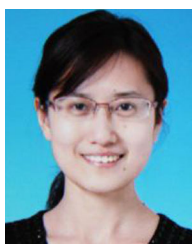
Yao Huang was born in Henan, China, in 1995. She received the B.Sc. degree in physics in 2018 from Zhengzhou University of Light Industry, Zhengzhou, China, where she is currently working toward the master's degree at the School of Computer and Communication Engineering.

Her research interests include hyperspectral image classification, kernel function, and machine learning.



Zebin Wu (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2007, respectively.

From 2014 to 2015, he was a Visiting Scholar with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, Cáceres, Spain. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include hyperspectral image processing, parallel computing, and remotely sensed big data processing.



Junru Yin was born in Henan, China, in 1984. She received the Ph.D. degree in forest management from Chinese Academy of Forestry, Beijing, China, in 2015.

Since 2015, she has been with Zhengzhou University of Light Industry, Zhengzhou, China, where she is currently a Lecture. Her current research interests include hyperspectral classification, machine learning and forest management.



Qiqiang Chen was born in Henan, China, in 1984. He received the Ph.D. degree in radio physics from Lanzhou University, Lanzhou, China, in 2015.

He is currently a Lecture with Zhengzhou University of Light Industry, Zhengzhou, China. His current research interests include hyperspectral classification, image processing, and machine learning.