A Pixel Cluster CNN and Spectral-Spatial Fusion Algorithm for Hyperspectral Image Classification With Small-Size Training Samples

Shuxian Dong, Yinghui Quan, Wei Feng, Gabriel Dauphin, Lianru Gao, and Mengdao Xing, *Fellow, IEEE*

***Abstract*—Convolutional neural networks (CNNs) can automatically learn features from the hyperspectral image (HSI) data, avoiding the difficulty of manually extracting features. However, the number of training samples for the classification of HSIs is always limited, making it difficult for CNN to obtain effective features and resulting in low classification accuracy. To solve this problem, a pixel cluster CNN and spectral-spatial fusion (SSF) algorithm for hyperspectral image classification with small-size training samples is proposed in this article. First, spatial information is extracted by the gray level co-occurrence matrix. Then, spatial information and spectral information are fused by means of bands superposition, forming spectral-spatial features. To expand the number of training samples, the pixels after SSF are combined into pixel clusters according to a certain rule. Finally, a CNN framework is utilized to extract effective features from the pixel clusters. Experiments based on three standard HSIs demonstrate that the proposed algorithm can get better performance than the conventional CNN and also outperforms other studied algorithms in the case of small training set.**

***Index Terms*—Convolutional neural network (CNN), hyperspectral image classification (HSIC), pixel cluster, small training set, spectral-spatial fusion (SSF).**

I. Introduction

HYPERSPECTRAL image (HSI) contains abundant amount of spectral information, and the 3-D data blocks can effectively reflect the information of the imaging target. So HSIs are widely used in the fields of precision agriculture, environmental monitoring, urban planning, and military reconnaissance [1]–[3]. Among these applications, hyperspectral image classification (HSIC) is one of the important links and has attracted more and more attention [4]–[9]. The ultimate goal of the classification is to give each pixel in the image a unique category identifier accurately.

Numerous algorithms have been proposed for HSIC [10]–[21] and they can be mainly classified into unsupervised classification, semisupervised classification, and supervised classification based on whether the prior information is required in the classification process. Iterative self-organizing data analysis techniques algorithm [10] and k-nearest neighbor [11] are typical unsupervised classification algorithms. Instead of using prior knowledge, both algorithms classify the HSI only based on the distribution of spectral characteristics of HSIs. Traditional semisupervised HSIC algorithms mainly include semisupervised SVM [12] and graph based algorithms [13]. Those algorithms train the labeled samples with the help of unlabeled samples, which can effectively make up for the shortage of training samples. Supervised classification is to train and learn the classifier using the labeled samples. Then, the trained classifier is used as decision rules to discriminate and classify the testing samples. Numerous supervised classification algorithms have been proposed and discussed, including support vector machine (SVM) [14], random forest [15], [16], decision tree [17], neural network [18]–[21], etc. All of those algorithms provide good ways for the classification of HSIs. But there is a disadvantage that the algorithms mentioned previously produce classification results only using the mechanism of shallow layer, cannot deal with the complex classification problem [3].

In recent years, deep learning-based algorithms have attracted increased attention and achieved remarkable progress in HSIC [22]. Deep learning allows the computer to automatically extract deep features and more abstract features to improve the accuracy of the classification. As the most popular and successful deep learning framework, convolutional neural network (CNN) utilizes a series of hidden layers to extract hierarchical features that have been proved to be effective in HSIC [23]–[28]. Wei *et al.* [23], first, employed the CNN with multiple layers for HSI classification. The deep CNN was used to classify HSIs directly in the spectral domain. After that, a set of improved algorithms based on CNN have been used to extract features for HSIC and yielded excellent performance. In [27], a novel supervised deep feature extraction algorithm combined siamese CNN with linear SVM was introduced. Chen

Manuscript received September 29, 2020; revised November 25, 2020, January 6, 2021, and February 19, 2021; accepted March 9, 2021. Date of publication March 29, 2021; date of current version April 23, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61772397 and Grant 12005169, in part by the National Key R&D Program of China under Grant 2016YFE0200400, in part by the Open Research Fund of Key Laboratory of Digital Earth Science under Grant 2019LDE005, and in part by the Science, and Technology Innovation Team of Shaanxi Province under Grant 2019TD-002. *(Corresponding authors: Wei Feng; Yinghui Quan.)*

Shuxian Dong, Yinghui Quan, and Wei Feng are with the Department of Remote Sensing Science, and Technology, School of Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: sxdong@stu.xidian.edu.cn; yhquan@mail.xidian.edu.cn; wfeng@xidian.edu.cn).

Gabriel Dauphin is with the Laboratory of Information Processing, and Transmission, Institut Galilée, University Paris XIII, 93430 Villetaneuse, France (e-mail: gabriel.dauphin@univ-paris13.fr).

Lianru Gao is with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: gaolr@aircas.ac.cn).

Mengdao Xing is with the Academy of Advanced Interdisciplinary Research, Xidian University, Xi'an 710071, China (e-mail: xmd@xidian.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3068864

et al. [28] combined deep learning and ensemble algorithms for HSIC first. Two deep learning ensemble-based classification methods [i.e., CNN ensemble and deep residual network ensemble] are proposed for HSI classification. Those algorithms achieved better classification performance than some traditional classifiers (SVM, extended morphological profiles, etc.).

With CNN-based algorithms have shown promising results in HSIC, spectral-spatial feature extraction and classification algorithms based on CNN have been explored recently [29]–[32]. It also has been verified that the spatial correlation across HSIs can provide complementary information to spectral features and should be taken into account [33]–[39]. Li *et al.* [37] designed a two-stream spectral and spatial feature extraction and fusion architecture based on 2-D CNN for HSIC. In [38], a novel HSIC framework based on a simplified 2-D–3-D CNN is implemented by the cooperation between a 2-D CNN and a 3-D convolution layer. Cao *et al.* [39] used a CNN framework in combination with Markov random field (MRF) to classify HSI pixel vectors in a way of fully taking spatial and spectral information into account. Those algorithms indicate that exploring the spectral-spatial information and jointly CNN structure can boost the performance of HSIC. Consequently, designing a deep learning model with spectral and spatial information is a promising direction to be explored.

However, the majority of CNN-based algorithms for HSIC can yield only promising results when the number of labeled samples for training is sufficiently large [27]. The CNN structure faces "overfitting" problem when the training samples are insufficient. Unfortunately, a small number of labeled samples are available for training in practical situations. To enhance the performance of the CNN-based algorithm when the training samples are limited, a pixel cluster CNN and spectral-spatial fusion (PC-CNN-SSF) algorithm for HSIC with small-size training samples is proposed in this article. The major contributions of this article can be summarized as follows.

1) Based on the fact that combined spectral-spatial classification improves the accuracy significantly compared with spectral classification alone [40], the gray level co-occurrence matrix (GLCM) is used to extract spatial information from each band of the original spectral information. Then, the spectral information and the extracted spatial information are fused by bands superposition. Finally, the SSF information is utilized together to generate an integrated classifier.
2) To guarantee the sufficient training of CNNs, Li *et al.* [41] proposed a CNN with pixel-pair feature (CNN-PPF) algorithm to increase the number of training samples, ensuring that the advantage of CNN can be actually offered. Inspired by the PPF, a pixel cluster algorithm is proposed in this article to expand the number of training samples.
3) Because CNN structures have shown the excellent performance of deep feature extraction and classification, pixel clusters based on SSF are trained by the CNN structure in this article. After that, the trained CNN model is used to classify the testing samples.

The remaining part of this article is organized as follows. The proposed pixel cluster CNN and SSF algorithm for HSIC with small-size training samples is described in Section II. Section III evaluates the performance of the proposed approach. Finally, Section IV concludes this article.

II. Proposed Approach

The flowchart of the proposed algorithm is shown in Fig. 1. And the algorithm mainly includes four steps: spatial information extraction and SSF, theory of pixel cluster, features extraction based on CNN, and classification by voting strategy. Note that, the first step and the second step are the key parts of the proposed algorithm.

A. Spatial Information Extraction and SSF

In the HSI, pixels have a strong spatial correlation because pixels with adjacent spatial locations generally belong to the same class of features. To improve the image classification performance, combining spectral information with spatial information is considered in this article.

Hyperspectral spatial information extraction process is shown in the Fig. 2. First, spatial information is extracted by the GLCM. GLCM not only reflects the distribution characteristics of intensity but also reflects the spatial distribution characteristics of pixels with the same intensity or close to the intensity.

Let us use a 3-D matrix Q to represent the original HSI and the size of matrix Q is $N_x \times N_y \times N_z$, where N_x and N_y represent the number of pixels in the horizontal and vertical directions of HSI, N_z means the number of bands. There exists the problem of a large amount of calculation due to the dimension of GLCM is equal to the square of the image gray level. Therefore, the gray level of the HSI should be compressed before extracting the spatial information. The value of Q is quantized to N_g levels to get a new matrix Q'

$$\begin{aligned} &Q'(x,y,d) \\ &= \left\lfloor (N_g - 1 + 0.9999) \cdot \frac{Q(x,y,d) - \min(Q(:,:,d))}{\max(Q(:,:,d)) - \min(Q(:,:,d))} \right\rfloor \end{aligned} \tag{1}$$

where $Q(x,y,d)$ and $Q'(x,y,d)$ represent the value of matrix Q and Q' in the column x, row y, band d, respectively. $\lfloor\rfloor$ is the down rounding operation. $\min(Q(:,:,d))$ and $\max(Q(:,:,d))$ are the minimum and maximum values of the dth band of matrix Q, respectively.

After the value of each band was quantized to $[0, N_g - 1]$, the GLCM is calculated by two moving windows—basic window (denoted by w_1) and transformed window (denoted by w_2). The size of two window are both $N_{wx} \times N_{wy}$. And the spatial information is extracted from each band independently. Let $P(i,j,d_x,d_y)$ represents the number of occurrences of a pixel pair between a gray value of i and a gray value of j, $0 \leqslant i,j \leqslant N_g - 1$, the number of rows and columns between the pairs of interest are d_y and d_x, respectively. For each pixel in the Q', the calculation formula for GLCM is

$$\begin{aligned} &P(i,j,d_x,d_y) \\ &= \sharp\{[(m,n),(m',n')], (m,n) \in w_1, (m',n') \in w_2 | \\ &\quad I(m,n) = i, I(m',n') = j, m' - m = d_y, n' - n = d_x\} \end{aligned} \tag{2}$$

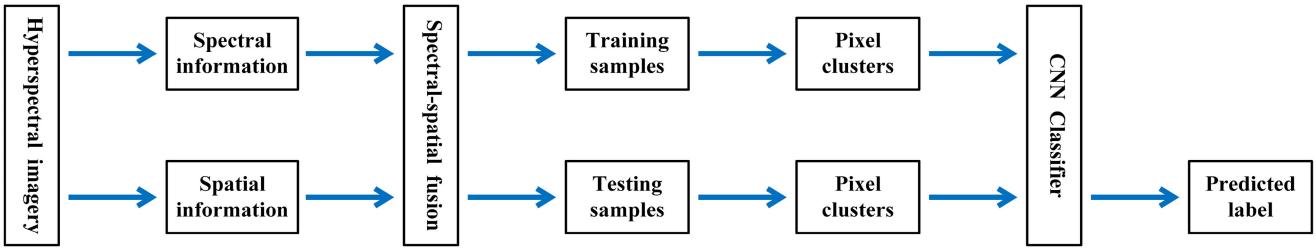


Fig. 1. Flowchart of the proposed algorithm.

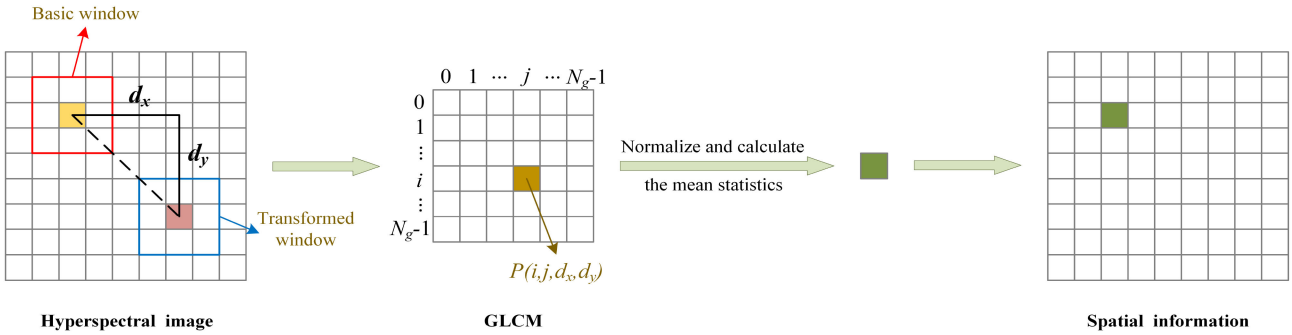


Fig. 2. Processing of spatial information extraction.

where ♯ is the number of pixels in the collection, (m, n) denotes the pixel of the mth row and the nth column in the basic window, and (m', n') means the m'th row and the n'th column in the transformed window, $1 \le n, n' \le N_{wx}, 1 \le m, m' \le N_{wy}$. $I(m, n)$ and $I(m', n')$ denote the gray level of the mth row and the nth column in basic window and the m'th row and the n'th column in the transformed window, respectively.

After calculating the GLCM, normalization is adopted to the matrix, represented as $P'(i, j, d_x, d_y)$

$$P'(i, j, d_x, d_y) = \frac{P(i, j, d_x, d_y)}{N_{wx} \times N_{wy}}. \tag{3}$$

In actual application, some statistics are defined as a feature value of texture analysis based on GLCM [42]. Haralick *et al.* [43] extracted 14 features from the GLCM. The mean statistic (denoted as I') is used to extract texture features in this article

$$I' = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} i \cdot P'(i, j, d_x, d_y). \tag{4}$$

When there are pixels in the basic window or the transformed window but not in the matrix Q', the gray value of the center pixel in spatial information is set to 0.

After the spatial information is successfully extracted from each band, all of them are superimposed on the original multispectral matrix Q to form the spectral-spatial matrix R. And the size of matrix R is $N_x \times N_y \times 2N_z$.

B. Theory of Pixel Cluster

With the purpose of increasing the number of training samples, a pixel cluster algorithm is designed to ensure that the advantage of CNN can be actually offered. Suppose there are N pixels in a training set, $e_1, e_2, \ldots, e_N$ be N training pixels with labels $f(e_1), f(e_2) \cdots f(e_N)$, respectively. $f(e_1), f(e_2) \cdots f(e_N) \in \{1, 2, \ldots, C\}$, where C is the number of classes in the hyperspectral dataset to be classified. The number of training samples per class is expressed as n_c and the total number of training samples is $N = \sum_{c=1}^{C} n_c$.

The pixel cluster proposed in this article is composed of P pixels, all of them are randomly selected from the training set. Therefore, a pixel cluster can be represented as $(e_1, e_2, \ldots, e_P)$, and the label of the pixel cluster is denoted as $f(e_1, e_2, \ldots, e_P)$. The label of pixel cluster is defined by using the following criteria: if the P samples are from the same class, the label of $f(e_1, e_2, \ldots, e_P)$ does not change; if the P samples are from different classes, the label of $f(e_1, e_2, \ldots, e_P)$ is set to be 0. That is

$$f(e_1, e_2, \ldots, e_P) = \{ {}^{c}_{0} \ {}^{\text{if } f(e_1)=f(e_2)=\cdots=f(e_P)=c)}_{\text{else}} \}. \tag{5}$$

Theoretically, if there are P pixels in a cluster, the number of pixel clusters is $A_N^P = N \times (N-1) \times \cdots \times (N-P+1)$. Denoting the number of training clusters per class is n_c', and $n_c' = n_c \times (n_c - 1) \times \cdots \times (n_c - P + 1)$, $c = 1, 2, \ldots, C$. For the $c = 0$, number of n_0' is larger than the number of $n_c'(c = 1, 2, \cdots C)$, since samples can be chosen from any different classes. To keep the data balanced, only an approximately equal number of clusters is determined. So the number of clusters used

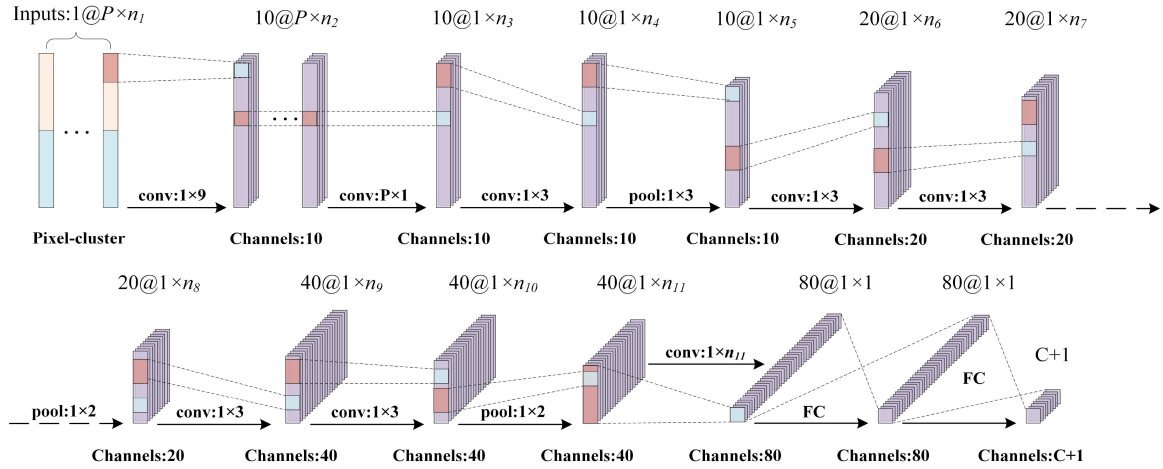


Fig. 3. Spectral-spatial features extraction from a deep CNN architecture.

to train the CNN model is about $N' = \sum_{c=0}^{C} n'_c$. It is obvious that N' is larger than N when $P \neq 1$, solving the shortage of training set obviously and enabling the training of a deep CNN model.

C. Spectral-Spatial Features Extraction Based on CNN

After obtaining the spectral-spatial pixel clusters, the training clusters, and training labels are feed into the designed CNN with the purpose of feature extraction. The CNN architecture is shown in Fig. 3. It transforms a pixel cluster with P pixels into C+1 channels, each of which predicts the reliability with which P pixels belong to the same class.

The framework includes eight convolutional layers, three max-pooling layers, and two fully connected layers. All of the convolutional layers are followed by a rectified linear unit (ReLU) layer to increase the nonlinear relationship between different layers. In the convolution layer, multiple shared convolution kernels are used to perform convolution operations on the input pixel-clusters to obtain multiple feature maps. Besides, the dimension of each feature map will be reduced according to the size of the convolution kernel if there is no padding used. Suppose the dimension of input map is d_1, and the size of convolution kernel is k_1, then the dimension of output layer is $d_1 - k_1 + 1$. Pooling layers are often used in CNN to reduce feature dimensions, model parameters and slow down the "overfitting" problem after convolutional layers. The pooling layer can ensure the invariance of the feature to the rotation and reduce the feature dimension. If the dimension of input map is d_2, and the size of kernel is k_2, then the dimension of output layer is d_2/k_2 after pooling downsampling. In this article, the max-pool function is used to calculate the maximum value of a rectangular area, which will be used to represent the area. After the input pixel-clusters of the network going through multiple convolutional layers and pooling layers, two fully connected layers are used to reassemble the local features extracted by the convolutional layer into a complete feature map, which is more conducive to classification. After two fully connected layers, the feature map is transformed into the feature vector of the corresponding label. Ultimately, the softmax function is used to calculate the probability distribution of input pixel-clusters being classified into different categories.

The 13 layers in the CNN structure can be divided into 7 different kinds of layers as follows:

1) **NP :** a neuron combining for each wavelength and channeling the intensity of each of the P pixels into one single intensity;
2) **NW :** a neuron combining for each channel the intensity of each wavelength into one single intensity;
3) **CT :** a convolution combining intensities of neighboring wavelengths in ten different convolutional kernels, yielding intensities for 10 different channels;
4) **CD :** a convolution combining for each channel intensities of neighboring wavelengths, yielding intensities for twice the number of channels;
5) **CS :** a convolution combining for each channel intensities of neighboring wavelengths, yielding intensities for the same number of channels;
6) **PL :** a pooling layer computing the maximal intensity, for each channel and each pair or triplet of neighboring wavelength;
7) **FC :** a fully connected layer combining the intensities of all channels in a certain number of ways and yielding a certain number of channels; and
8) **MV :** a majority voting, based on values of $C + 1$ outputs.

A simplified description of the deep CNN architecture is done by considering five sets of layers

$$c = \mathbf{CNN}\widehat{\mathbf{x}} \quad \text{where} \begin{cases} \mathbf{x}_2 &= \mathbf{NP}\,\mathbf{CT}\widehat{\mathbf{x}} \\ \mathbf{x}_4 &= \mathbf{PL}_1\,\mathbf{CS}_1\mathbf{x}_2 \\ \mathbf{x}_7 &= \mathbf{PL}_2\,\mathbf{CS}_2\,\mathbf{CD}_1\mathbf{x}_4 \\ \mathbf{x}_{10} &= \mathbf{PL}_3\,\mathbf{CS}_3\,\mathbf{CD}_2\mathbf{x}_7 \\ c &= \mathbf{MV}\,\mathbf{FC}_2\,\mathbf{FC}_1\,\mathbf{NW}\mathbf{x}_{10} \end{cases} \tag{6}$$

and $\widehat{\mathbf{x}}$ is the set of all intensity values of a pixel cluster with P pixels for all wavelengths and c is the predicted class.

Denoting $\mathbf{ReLU}(x) = \max(0, x)$ and using p to address the pixel in a given pixel-cluster, d to address a wavelength and l to address a channel, we get a layer-by-layer description of the studied architecture.

The first set of layers consists of $\mathbf{CT}$ and $\mathbf{NP}$

$$\begin{cases} x_{1,p,d,l} = \mathbf{ReLU}\left(\sum_{d'=0}^{8} a^{0,d,'l} x_{0,p,d+d'} + b^{0,l}\right) \\ \text{with } 1 \le p \le P,\ 1 \le d \le 2N_z - 8,\ 1 \le l \le 10 \\ x_{2,d,l} = \mathbf{ReLU}\left(\sum_{p=1}^{P} a^{1,p,l} x_{1,p,d} + b^{1,l}\right) \\ \text{with } 1 \le d \le 2N_z - 8,\ 1 \le l \le 10 \end{cases}. \quad (7)$$

The second set of layers consists of $\mathbf{CS}_1$ and $\mathbf{PL}_1$

$$\begin{cases} x_{3,d,l} = \mathbf{ReLU}\left(\sum_{d'=0}^{2} a^{2,d,'l} x_{2,d+d'} + b^{2,l}\right) \\ \text{with } 1 \le d \le 2N_z - 8,\ 1 \le l \le 10 \\ x_{4,d,l} = \underset{0 \le d' \le 2}{\mathbf{max}} \left(x_{3,3d-d',l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 8}{3} \rceil,\ 1 \le l \le 10 \end{cases}. \quad (8)$$

The third set of layers consists of $\mathbf{CD}_1$, $\mathbf{CS}_2$, and $\mathbf{PL}_2$

$$\begin{cases} x_{5,d,l} = \mathbf{ReLU}\left(\sum_{d'=0}^{2} a^{4,d,'l} x_{4,d+d'} + b^{4,l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 14}{3} \rceil,\ 1 \le l \le 20 \\ x_{6,d,l} = \mathbf{ReLU}\left(\sum_{d'=0}^{2} a^{5,d,'l} x_{5,d+d'} + b^{5,l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 20}{3} \rceil,\ 1 \le l \le 20 \\ x_{7,d,l} = \underset{0 \le d' < 2}{\mathbf{max}} \left(x_{6,2d-d',l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 20}{6} \rceil,\ 1 \le l \le 20 \end{cases}. \quad (9)$$

The fourth set of layers consists of $\mathbf{CD}_2$, $\mathbf{CS}_3$, and $\mathbf{PL}_3$

$$\begin{cases} x_{8,d,l} = \mathbf{ReLU}\left(\sum_{d'=0}^{2} a^{7,d,'l} x_{7,d+d'} + b^{7,l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 32}{6} \rceil,\ 1 \le l \le 40 \\ x_{9,d,l} = \mathbf{ReLU}\left(\sum_{d'=0}^{2} a^{8,d,'l} x_{8,d+d'} + b^{8,l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 44}{6} \rceil,\ 1 \le l \le 40 \\ x_{10,d,l} = \underset{0 \le d' \le 2}{\mathbf{max}} \left(x_{9,2d-d',l}\right) \\ \text{with } 1 \le d \le \lceil \frac{2N_z - 44}{12} \rceil,\ 1 \le l \le 40 \end{cases}. \quad (10)$$

The last set of layers consists of $\mathbf{NW}$, $\mathbf{FC}_1$, and $\mathbf{FC}_2$

$$\begin{cases} x_{11,l} = \mathbf{ReLU}\left(\sum_{d'=1}^{\lceil \frac{2N_z - 44}{12} \rceil} a^{10,d,'l} x_{10,d'} + b^{10,l}\right) \\ \text{with } 1 \le l \le 80 \\ x_{12,l} = \mathbf{ReLU}\left(\sum_{l'=1}^{80} a^{11,l'} x_{11,l'} + b^{11,l}\right) \\ \text{with } 1 \le l \le 80 \\ x_{13,l} = \mathbf{ReLU}\left(\sum_{l'=1}^{80} a_c^{12,l'} x_{12,l'} + b^{12,l}\right) \\ \text{with } 1 \le l \le C + 1 \\ c = \mathbf{arg\,max}\left(\mathbf{soft\,max}(x_{13})\right) \end{cases}. \quad (11)$$

D. Classification With Voting Strategy

The CNN described previously predicts category of a pixel using the pixel cluster algorithm. Such a classifier can be transformed into a pixel classifier by combining the given testing pixel $\mathbf{x}$ and the surrounding pixels. The testing process is shown in Fig. 4. As first step, a $w \times w$ size window W is set around the center testing pixel $\mathbf{x}$ (w being here an odd integer), forming pixel set $[\mathbf{x}]$

$$[\mathbf{x}] = S_W \mathbf{x} \quad (12)$$

where $S_W \mathbf{x}$ denotes pixels in the window W when the center pixel of the window W is $\mathbf{x}$. Note that to simplify notations, we assume that $\mathbf{x}$ stands for a pixel and the intensity values on the hyperspectral image at that pixel.

The second step is to set the Q pixel clusters denoted as $[\widehat{\mathbf{x}}]$ by joining each $\mathbf{x}$ with its neighbors

$$[\widehat{\mathbf{x}}] = \left[\begin{bmatrix} \mathbf{x}_{1,1} \\ \mathbf{x}_{2,1} \\ \vdots \\ \mathbf{x}_{P-1,1} \\ \mathbf{x} \end{bmatrix}, \begin{bmatrix} \mathbf{x}_{1,2} \\ \mathbf{x}_{2,2} \\ \vdots \\ \mathbf{x}_{P-1,2} \\ \mathbf{x} \end{bmatrix}, \ldots, \begin{bmatrix} \mathbf{x}_{1,Q} \\ \mathbf{x}_{2,Q} \\ \vdots \\ \mathbf{x}_{P-1,Q} \\ \mathbf{x} \end{bmatrix} \right] \quad (13)$$

where $\mathbf{x}_{p,q} (p = 1, 2, \ldots, P-1, q = 1, 2, \ldots, Q)$ are pixels randomly selected form the set $[\mathbf{x}]$ (except for the center pixel $\mathbf{x}$).

In the last step, the Q pixel clusters are fed to the $\mathbf{CNN}$-classifier and yield Q possible predicted class values. The predicted label is the class chosen most often by the Q clusters of pixels. Denoting $[\widehat{\mathbf{x}}_q]$ as the qth column of $[\widehat{\mathbf{x}}]$, the category of testing pixel $\mathbf{x}$ is determined by the following formula:

$$c = \mathbf{PC\text{-}CNN}\mathbf{x} = \underset{1 \le c' \le C}{\mathbf{max}} \sum_{q=1}^{Q} \mathbf{1}\left(\mathbf{CNN}\widehat{\mathbf{x}}_q = c'\right) \quad (14)$$

where $1(\text{Statement})$ is equal to 1 when Statement is true and is equal to 0 when Statement is wrong.

III. Experimental Results

A. Experiment Settings

There are nine algorithms discussed in the experiment: SVM, SVM-RFS [44], SVM-MRF [45], conventional CNN [23], CNN-MRF [39], CNN-PPF [41], R-PCA-CNN [46], GCN-CNN [47], and the proposed PC-CNN-SSF. The configurations of the above methods are as follows.

1) SVM: A classical machine learning algorithm, which is widely used in small-size training datasets. The SVM algorithm applied in this section is implemented using the "e1071" package.[1] The kernel function used for SVM is the radial basis function. Both the gamma and the cost value used in the kernel are jointly selected so as to optimize the average performance obtained with cross validation. The optimized values of the gamma and the cost are searched respectively within $[2^{-8} : 2^{8}]$ and $[2^{1} : 2^{8}]$.
2) SVM-RFS [44]: Multiple classifier systems (MCSs) based on SVM and random feature selection (RFS). According to the discussion of the optical parameters in [44], the number of selected features and the size of the MCS are set to 30% and 25 in the experiment.

[1] Online. [Available]: https://cran.r-project.org/web/packages/e1071/index.html

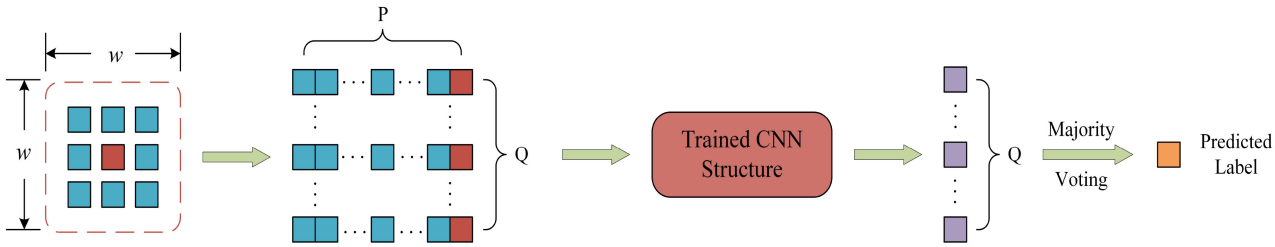


Fig. 4. Testing Process with voting strategy.

3) SVM-MRF [45]: A traditional spectral-spatial algorithm, which consists in performing a probabilistic SVM pixel-wise classification, followed by MRF-based regularization for incorporating spatial and edge information into classification.
4) CNN [23]: A conventional deep CNN structure, which is used to classify HSIs directly in spectral domain. And experimental results demonstrate that the algorithm could achieve higher accuracy even with a small number of training samples.
5) CNN-MRF [39]: A typical CNN combined with MRF method, both spectral and spatial information are fully considered in the algorithm.
6) CNN-PPF [41]: A CNN framework based on CNN-PPF. The PPFs are used to expand the number of training samples, which can make up for the shortage of training samples.
7) R-PCA-CNN [46]: A typical SSF algorithm based on CNN, which extracted spatial information by randomized principal component analysis (R-PCA) first, and then a CNN was used to encode pixel spectral and spatial information and a multilayer perceptron was used to conduct the classification task.
8) GCN-CNN [47]: A new concatenated fusion framework by integrating features extracted from CNNs and mini-batch graph convolutional networks (GCNs). The joint use of CNNs and GCNs extracts more diverse and discriminative feature representations for the HSIC task.
9) PC-CNN-SSF: Parameters designed in the proposed algorithm are demonstrated as follows. In spatial information extraction and SSF stage, window size $N_{wx} = N_{wy} = 5$, each band of the original HSI is quantized to $N_g = 64$, $d_x = 1$, $d_y = 0$. During the testing stage, window size around the center testing pixel and the number of pixel clusters for a testing pixel are set to $w = 5$ and $Q = 24$.

B. Evaluation Algorithms

The effectiveness of the proposed classification algorithm is measured in the experiment with three indicators: Overall accuracy (OA), average accuracy (AA), and Kappa coefficient (KC). These indicators are computed using the number of testing pixels T and the confusion matrix $[T_{cc'}]$ specific to a given classifier. $T_{cc'}$ is the number of testing pixels whose actual class is y^* and whose predicted class is $\widehat{y}$

$$T_{cc'} = \sum_{t=1}^{T} \mathbf{1}\left(y_t^* = c\right) \mathbf{1}\left(\widehat{y}_t = c'\right) \tag{15}$$

where y_t^* and $\widehat{y}_t$ are, respectively, the true class and predicted class of testing sample number t.

1) **OA** refers to the ratio of the number of correctly classified pixels to the total number of pixels

$$\mathbf{OA} = \frac{1}{T} \sum_{c=1}^{C} T_{cc}. \tag{16}$$

2) **AA** measures the average per-class classification accuracy, where the per-class accuracy is defined as the number of correctly classified pixels of a given class to the number of pixels of that given class

$$\mathbf{AA} = \frac{1}{C} \sum_{c=1}^{C} \frac{T_{cc}}{\sum_{c'=1}^{C} T_{cc'}}. \tag{17}$$

3) **KC** is a statistic that attempts to correct **OA** by reducing its value when agreement could be obtained by chance

$$\mathbf{KC} = \frac{\frac{1}{T}\sum_c T_{cc} - \frac{1}{T^2}\left(\sum_{c'} T_{cc'}\right)\left(\sum_{c'} T_{c'c}\right)}{1 - \frac{1}{T^2}\left(\sum_{c'} T_{cc'}\right)\left(\sum_{c'} T_{c'c}\right)}. \tag{18}$$

C. Experiment Data

Three well-know hyperspectral datasets, including Pavia University, Salinas, and Indian Pines, are employed to validate the effectiveness of the proposed algorithm in this article.

The Pavia University scene is acquired by the reflective optics system imaging spectrometer sensor during a flight campaign over Pavia, northern Italy. There are 610 × 340 pixels, with a spatial resolution of 1.3 m. The number of spectral bands is 103, covering 0.43 to 0.86 μm. As is shown in Table I, the Pavia University data scene mainly includes 9 classes, denoted by labels from 1 to 9. The background pixels are represented by label 0 and will not be taken into account for classification.

The second one is the Salinas scene collected by the airborne visible-infrared imaging spectrometer (AVIRIS) sensor over Salinas Valley, California. It contains 512 × 217 pixels with a ground resolution of 3.7 m. 204 (out of 224) bands after 20 water absorption bands removed are utilized in the experiment. The spectral coverage is also from 0.43 to 0.86 μm, which is similar to the wavelength of the Pavia University dataset. There are 16 classes in the scene, details are shown in Table I.

TABLE I
HYPERSPECTRAL DATA INFORMATION

Pavia University			Salinas			Indian Pines		
No.	**Categories**	**Samples**	**No.**	**Categories**	**Samples**	**No.**	**Categories**	**Samples**
C1	Asphalt	6631	C1	Brocoli_green_weeds_1	2009	C1	Alfalfa	46
C2	Meadows	18649	C2	Brocoli_green_weeds_2	3726	C2	Corn_notill	1428
C3	Gravel	2099	C3	Fallow	1976	C3	Corn_mintill	830
C4	Trees	3064	C4	Fallow_rough_plow	1394	C4	Corn	237
C5	Sheets	1345	C5	Fallow_smooth	2678	C5	Grass_pasture	483
C6	Bare Soil	5029	C6	Stubble	3959	C6	Grass_trees	730
C7	Bitumen	1330	C7	Celery	3579	C7	Grass_pasture_mowed	28
C8	Bricks	3682	C8	Grapes_untrained	11271	C8	Hay_windrowed	478
C9	Shadows	947	C9	Soil_vinyard_develop	6203	C9	Oats	20
			C10	Corn_senesced_green_weeds	3278	C10	Soybean_notill	972
			C11	Lettuce_romaine_4wk	1068	C11	Soybean_mintill	2455
			C12	Lettuce_romaine_5wk	1927	C12	Soybean_clean	593
			C13	Lettuce_romaine_6wk	916	C13	Wheat	205
			C14	Lettuce_romaine_7wk	1070	C14	Woods	1265
			C15	Vinyard_untrained	7268	C15	Buildings_Grass_Trees_Drives	386
			C16	Vinyard_vertical_trellis	1807	C16	Stone_Steel_Towers	93
Total		**42776**	**Total**		**54129**	**Total**		**10249**

Bold highlights the highest classification accuracy.

TABLE II
CLASSIFICATION PERFORMANCES OF DIFFERENT ALGORITHMS FOR THE PAVIA UNIVERSITY DATASET

No.	SVM	SVM-RFS	SVM-MRF	CNN	CNN-MRF	CNN-PPF	R-PCA-CNN	GCN-CNN	PC-CNN-SSF
1	76.67	79.21	82.72	80.07	89.45	95.19	87.77	81.32	94.58
2	85.31	85.28	86.75	83.69	89.58	85.98	84.10	94.90	90.97
3	72.14	78.67	81.02	80.83	84.65	89.46	85.64	85.53	88.56
4	90.07	92.30	89.62	90.87	96.23	90.98	93.65	94.75	94.59
5	99.69	99.69	99.85	99.10	99.83	99.91	100	99.88	99.89
6	88.56	90.07	93.45	88.97	89.48	92.87	85.66	89.62	94.34
7	92.89	93.00	95.08	91.47	91.06	94.58	95.64	90.42	96.58
8	85.71	85.80	91.60	81.34	81.41	87.36	80.89	78.95	97.02
9	99.96	99.89	99.89	99.46	99.98	99.13	99.15	100	97.84
OA(%)	85.07	86.11	88.19	84.96	89.66	89.87	86.52	90.46	**93.18**
AA(%)	87.89	89.32	91.11	88.42	91.30	92.83	90.28	90.60	**94.93**
KC($\times$100)	80.61	81.98	84.63	80.54	86.56	86.81	82.68	87.41	**91.05**

Bold highlights the highest classification accuracy.

The last data is Indian Pines gathered by AVIRIS sensor over the Indian Pines test site in North-western Indiana. It consists of 145×145 pixels and 200 (out of 220) spectral reflectance bands by removing bands covering the region of water absorption. The wavelength range is $0.4 - 2.5$ μ m and the ground truth available is designated into 16 classes.

D. Experiment Results and Analysis

1) Comparison With Other Algorithms: In this section, the proposed algorithm is compared with some of the state-of-the-art algorithms for HSIC. Specifically, 50 training samples are randomly selected from each class to construct the training set, and the remaining samples are used to validate the effectiveness of the proposed algorithm. The number of pixels in a pixel is set to $P = 2$. We note that for the smallest classes Alfalfa, Grass-pasture-mowed and Oats of the Indian Pines dataset, only half of their samples are selected randomly to construct the training set.

Per class accuracy, OA, AA, and KC obtained by nine classification algorithms on three hyperspectral datasets are shown in Tables II–IV. For the Indian Pines scene, the proposed algorithm (PC-CNN-SSF) obtains the biggest improvement than the comparison algorithms on the term of OA, AA, and KC. Compared with the SVM-based algorithms, the proposed algorithm achieves OA 94.02%, with the gains of 23.36%, 15.72%, and 12.77% over SVM, SVM-RFS, and SVM-MRF, respectively. When compared to the conventional CNN and spectral-spatial algorithm based on CNN, the proposed algorithm yields over 18.44%, 18.12%, 13.57%, and 15.77% higher accuracy than the CNN, R-PCA-CNN, CNN-MRF, and GCN-CNN, respectively. The reason might be explained that the pixel clusters generated by the proposed algorithm are effective and meaningful, ensuring the full training of CNN and the improvement of classification accuracy. The proposed algorithm obtains about 9.02% improvement compared with the pixel expanding algorithm CNN-PPF due to exploiting more spatial context information. When applying nine algorithms to realize the classification of the Pavia University dataset and the Salinas dataset, the proposed algorithm also better classification performance than the other eight competitive algorithms.

Apart from quantitative analysis, Figs. 5–7 provide the ground truth and classification maps of the nine algorithms with the same labeled samples for the Pavia University dataset, the Salinas dataset, and the Indian Pines dataset. Visually, they are consistent with the results reported in Tables II–IV. From these figures, it

TABLE III
CLASSIFICATION PERFORMANCES OF DIFFERENT ALGORITHMS FOR THE SALINAS DATASET

No.	SVM	SVM-RFS	SVN-MRF	CNN	CNN-MRF	CNN-PPF	R-PCA-CNN	GCN-CNN	PC-CNN-SSF
1	98.81	98.77	98.52	96.87	95.50	98.66	98.83	99.20	98.97
2	99.30	99.35	97.66	98.15	99.07	99.09	98.89	99.95	99.81
3	96.96	98.13	98.18	98.17	88.45	99.17	96.98	97.38	99.92
4	99.70	99.70	100	99.78	97.68	99.87	98.57	99.51	99.85
5	96.18	96.80	95.13	95.94	99.48	95.30	97.26	98.55	96.90
6	99.60	99.59	99.72	99.39	99.62	99.57	98.09	99.98	99.76
7	99.63	99.63	99.69	99.11	99.68	99.61	98.82	99.81	98.51
8	74.50	79.42	88.58	76.42	85.26	84.84	80.32	66.84	86.84
9	99.06	99.11	97.94	98.88	95.43	98.50	97.98	99.85	99.31
10	87.74	88.85	92.32	87.32	97.24	90.90	93.85	93.71	91.98
11	97.23	98.04	97.54	93.48	98.55	99.27	97.15	98.41	97.45
12	98.96	99.68	100	99.77	99.99	99.99	100	99.94	99.99
13	98.50	98.73	98.61	98.52	99.93	99.19	100	99.88	99.01
14	93.43	92.84	95.59	93.20	99.39	94.71	96.82	98.37	93.25
15	64.96	71.04	62.33	70.67	71.49	75.23	79.92	73.82	83.85
16	98.84	98.69	100	98.81	98.70	98.93	96.56	97.94	98.22
OA(%)	88.31	90.35	91.08	89.22	91.48	92.14	92.62	88.72	**93.90**
AA(%)	93.96	94.90	95.11	94.03	95.34	95.80	95.63	95.20	**96.48**
KC(×100)	87.11	89.25	90.05	88.02	90.51	91.24	91.79	87.47	**93.21**

Bold highlights the highest classification accuracy.

TABLE IV
CLASSIFICATION PERFORMANCES OF DIFFERENT ALGORITHMS FOR THE INDIAN PINES DATASET

No.	SVM	SVM-RFS	SVM-MRF	CNN	CNN-MRF	CNN-PPF	R-PCA-CNN	GCN-CNN	PC-CNN-SSF
1	91.30	95.65	91.30	95.65	95.65	95.65	91.30	95.65	97.39
2	66.39	77.00	69.67	69.03	78.40	85.24	63.00	60.26	96.39
3	64.36	66.92	61.67	71.46	86.92	84.00	76.18	72.18	95.38
4	84.28	89.84	98.40	87.06	99.15	94.22	87.59	92.09	98.72
5	89.88	90.99	94.23	91.96	96.44	91.59	90.33	91.96	94.27
6	91.97	96.18	98.53	92.74	93.79	98.15	97.14	98.59	99.62
7	85.71	85.71	21.43	85.71	100	82.86	100	97.14	97.14
8	91.12	96.96	100	96.59	97.15	99.16	99.77	98.04	99.86
9	68.00	90.00	100	100	100	96.00	100	100	100
10	68.98	74.95	72.89	78.72	75.27	84.08	72.54	70.95	95.51
11	56.31	67.53	73.26	61.32	59.89	72.08	60.95	73.74	84.78
12	67.7	86.19	83.24	80.96	76.50	92.85	62.58	77.72	97.94
13	99.23	99.35	100	99.61	98.96	100	100	99.87	100
14	78.85	84.03	97.53	82.80	94.47	86.68	93.65	86.29	97.56
15	65.65	66.67	98.81	65.65	98.27	94.35	86.67	83.10	98.45
16	93.02	93.02	88.37	95.35	100	100	100	100	99.53
OA(%)	70.66	78.30	81.25	75.58	80.45	85.00	75.90	78.25	**94.02**
AA(%)	78.92	85.06	84.33	84.66	90.68	91.06	86.36	87.35	**97.03**
KC(×100)	66.8	75.31	78.54	72.38	77.92	82.95	72.67	75.19	**93.17**

Bold highlights the highest classification accuracy.

is evident that the classification maps of the proposed algorithm clearly shows the proper assignment of class label to each of the reference labels, tending to result in clearer and smoother classification maps.

It is well known that the number of training samples affects the training of CNNs significantly. In this section, the impact of the number of training samples on the accuracies of different algorithms for the three datasets is tested. For all the data, 10 to 50 pixels per class are randomly selected as training pixels and the remaining pixels are used as the test set. For the smallest classes Alfalfa, Grass-pasture-mowed, and Oats of the data Indian Pines AVRIS, there are only half of their samples are selected randomly to construct training sets when the number of training samples is more than half of the total number of samples. Figs. 8– 10 show the OAs, AAs, and KCs of the nine algorithms under different training sample numbers. Note that the CNN-based algorithms are sensitive to the number of training samples. And all the classifiers performed better with the increase of the number of training samples in most cases. The proposed algorithm consistently provides superior OAs, AAs, and KCs compared with the reference algorithms for three HSIs. The reason can be summarized as follows.

1) Spectral and spatial information (extracted by the GLCM) are considered simultaneously in the proposed algorithm.
2) Sufficient training samples provided by the proposed pixel cluster algorithm guarantee the network parameters to be well-tuned.
3) The voting operation makes the test set get more accurate classification results.

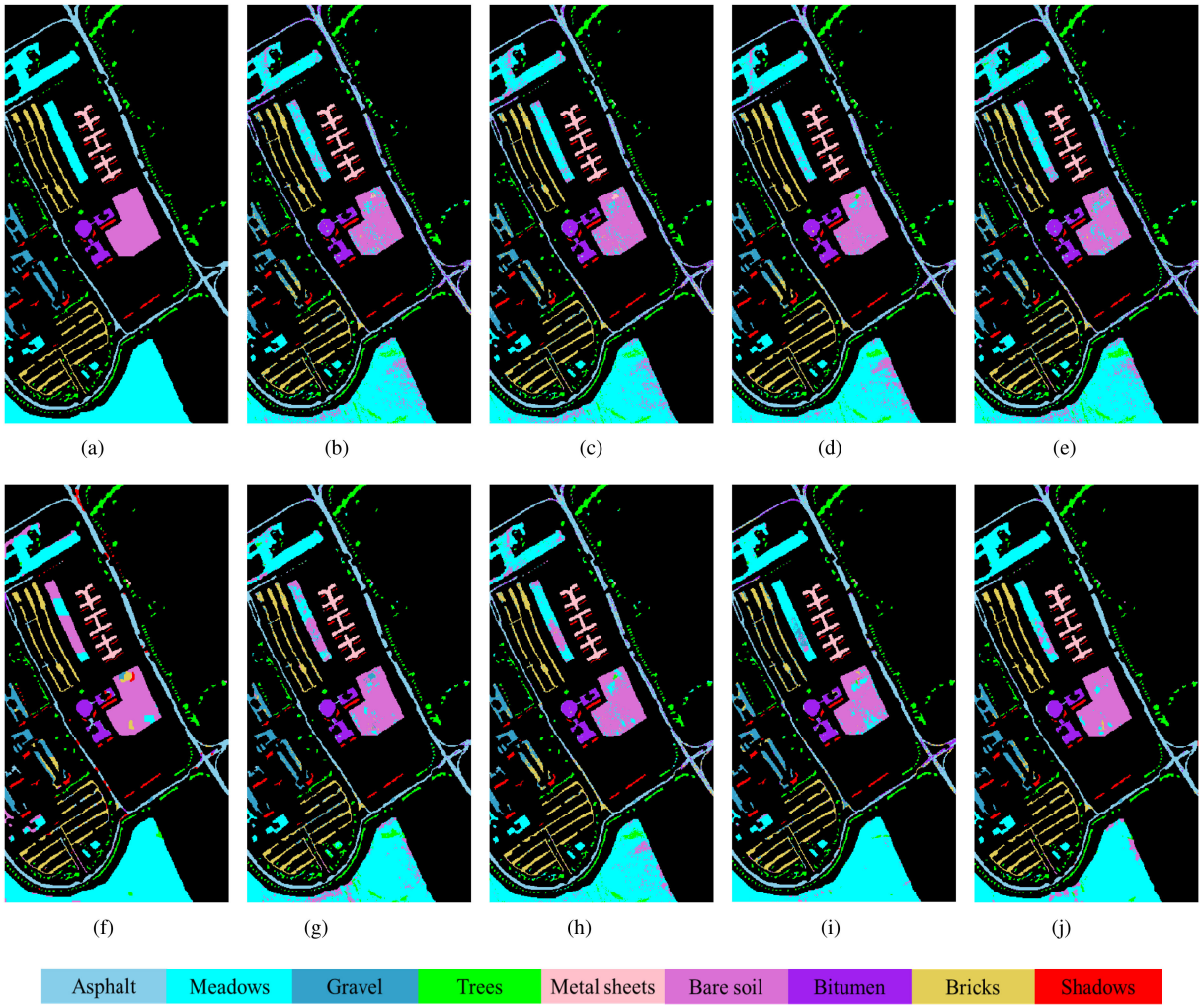


Fig. 5. Classification maps generated by different algorithms for the Pavia University dataset. (a) SVM (OA = 85.07%). (b) SVM-RFS (OA = 86.11%). (c) SVM-MRF (OA = 88.19%). (d) CNN (OA = 84.96%). (e) CNN-MRF (OA = 89.66%). (f) CNN-PPF (OA = 89.87%). (g) R-PCA-CNN (OA = 86.52%). (h) GCN-CNN (OA = 90.46%). (i) PC-CNN-SSF (OA = **93.18%**).

TABLE V
CLASSIFICATION PERFORMANCES (OA(%)) OF DIFFERENT WINDOW SIZE AND PIXEL CLUSTER NUMBER FOR THE THREE HSI DATASETS

	Pavia University				Salinas				Indian Pines			
	Q=6	Q=12	Q=18	Q=24	Q=6	Q=12	Q=18	Q=24	Q=6	Q=12	Q=18	Q=24
w=3	84.59	/	/	/	87.63	/	/	/	81.52	/	/	/
w=5	85.46	85.56	85.74	85.77	88.17	88.28	88.41	88.48	81.81	82.88	83.05	83.18
w=7	85.73	86.35	86.66	**86.76**	88.41	88.80	88.88	**88.97**	82.22	83.28	83.78	**83.95**

Bold highlights the highest classification accuracy.

2) Parameter Analysis: Parameter analysis of the proposed algorithm is presented in the following section. With the purpose of reducing the amount of computation and improving efficiency, 20 training samples are randomly selected from per class, and the remaining samples are used for testing.

In order to discuss the influence of the window size w and the number of voting pixel-clusters Q on the final classification results, the two parameters are set to $w = 3, 5, 7$ and $Q = 6, 12, 18, 24$, respectively. The number of pixels in a pixel is set to $P = 2$. Because there are only eight pixels around the center testing pixel when the window size is $w = 3$, we just use $Q = 8$ pixel clusters to voting the label of the center testing pixel. Classification performance based on three hyperspectral data is presented in Table V. It is shown that the classification performance is improving with the increase of window size w or the number of voting pixel-clusters Q. Therefore, randomly selecting pixels from a larger window or selecting more pixels from the same size window is helpful to improve the classification accuracy. This might can be explained that taking into account more pixels around the testing pixel ensures the accuracy and

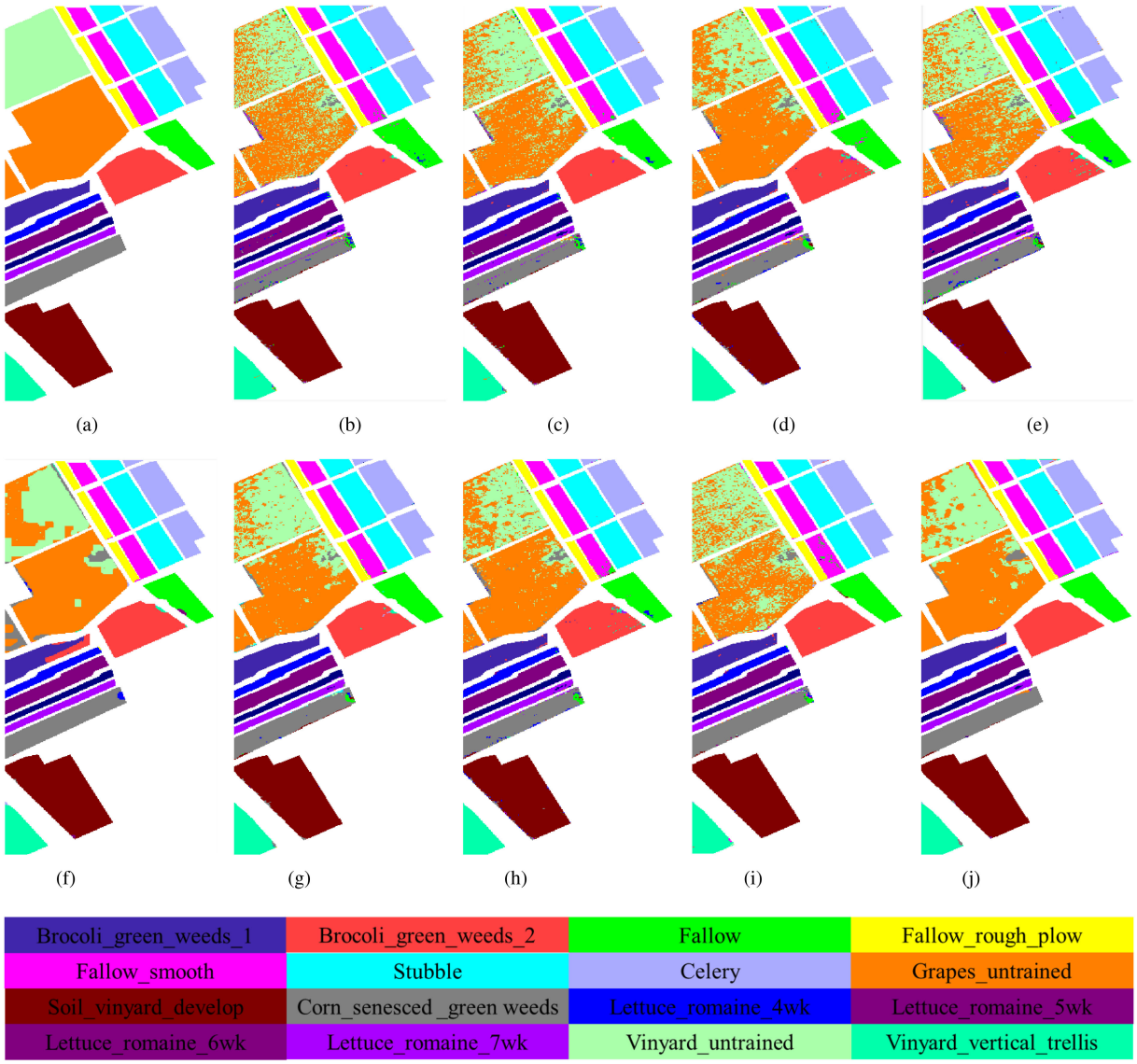


Fig. 6. Classification maps generated by different algorithms for the Salinas dataset. (a) SVM (OA=88.31%). (b) SVM-RFS (OA=90.35%). (c) SVM-MRF (OA=91.08%). (d) CNN (OA=89.22%). (e) CNN-MRF (OA=91.48%). (f) CNN-PPF (OA=92.14%). (g) R-PCA-CNN (OA=92.62%). (h) GCN-CNN (OA=88.72%). (i) PC-CNN-SSF (OA=**93.90%**).

TABLE VI
CLASSIFICATION PERFORMANCES OF THE PROPOSED ALGORITHM WHEN THE 0TH CLASS IS CONSIDERED OR NOT FOR THE THREE HSI DATASETS

HSI Data Set	the 0th is not considered			the 0th is considered		
	OA (%)	AA (%)	KC (×100)	OA (%)	AA (%)	KC (×100)
Pavia University	81.74	85.63	76.48	**85.77**	89.32	81.59
Salinas	84.36	91.80	82.65	**88.48**	94.01	87.24
Indian Pines	79.92	86.74	77.36	**83.18**	90.37	81.01

Bold highlights the highest classification accuracy.

reliability. However, it also takes more computation time and computer memory to vote the label of the center testing pixel. Thus, we choose $w = 5$ and $Q = 24$ as the default setting as a tradeoff between performance and the running time.

As described in Section II-B, the label of a pixel cluster is defined as $c = 0$ when the P pixels in the cluster are from different classes. Correspondingly, the impact of the 0th class on the classification performance of the proposed algorithm is reported in Table VI. It can be seen that there are 4.03%, 4.12% and 3.26% overall accuracy improvements on the three hyperspectral datasets when the 0th class is added to the training pixel-clusters. For AAs and KCs, the proposed algorithm

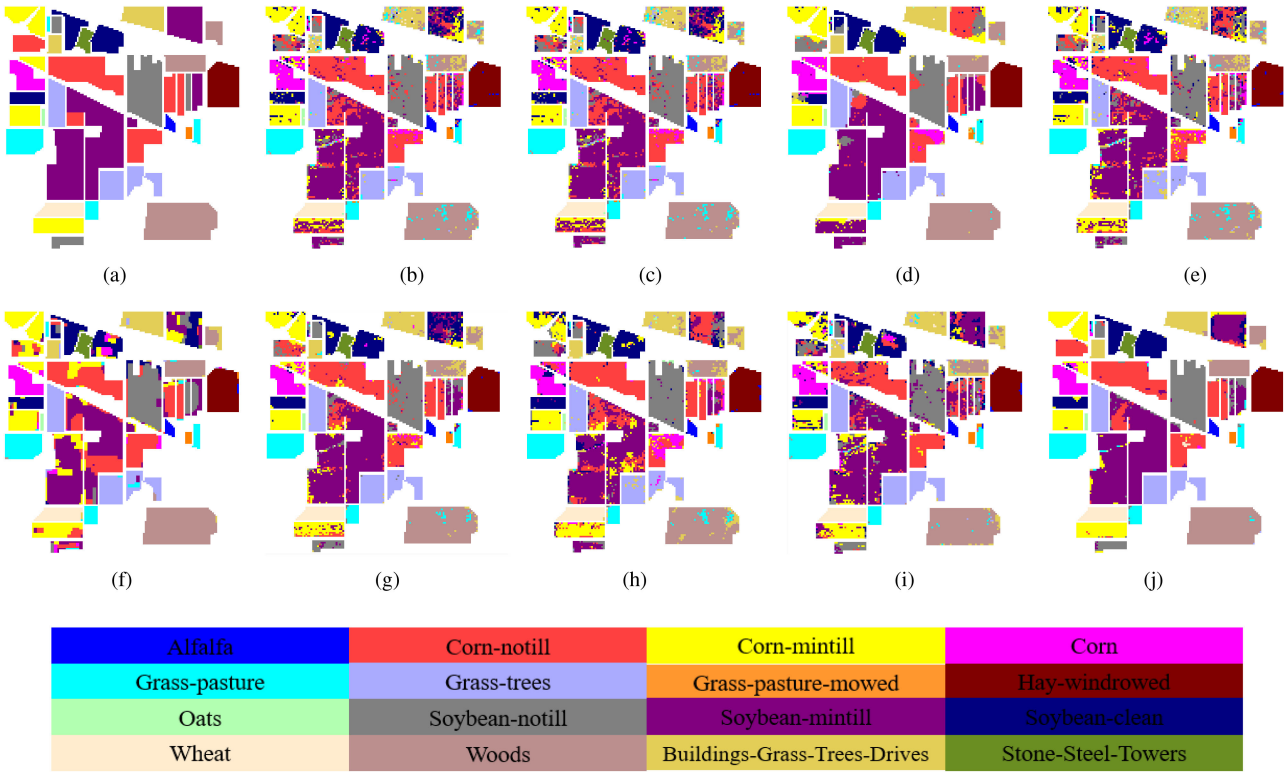


Fig. 7. Classification maps generated by different algorithms for the Indian pines dataset. (a) SVM (OA=70.66%). (b) SVM-RFS (OA=78.30%). (c) SVM-MRF (OA=81.25%). (d) CNN (OA=75.58%). (e) CNN-MRF (OA=80.45%). (f) CNN-PPF (OA=85.00%). (g) R-PCA-CNN (OA=75.90%). (h) GCN-CNN (OA=78.25%). (i) PC-CNN-SSF (OA=**94.02%**).

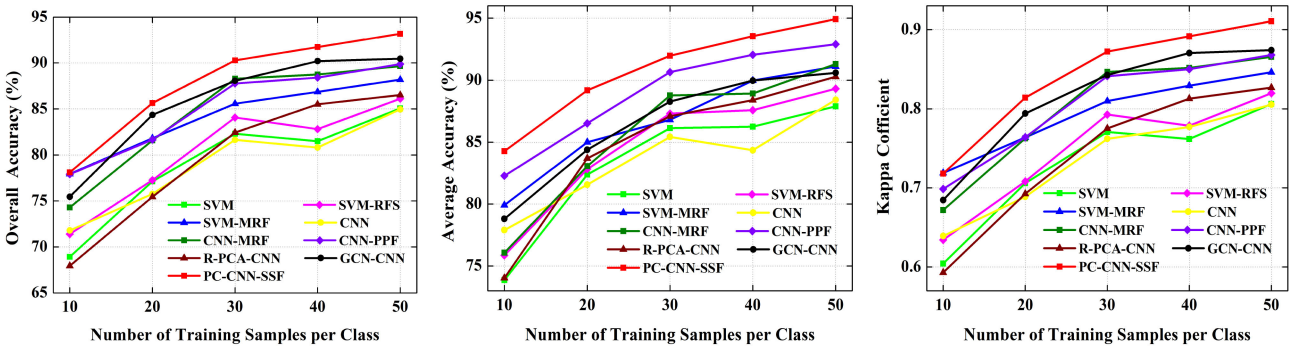


Fig. 8. Classification performances of nine algorithms with different training sample sizes on the Pavia University dataset.

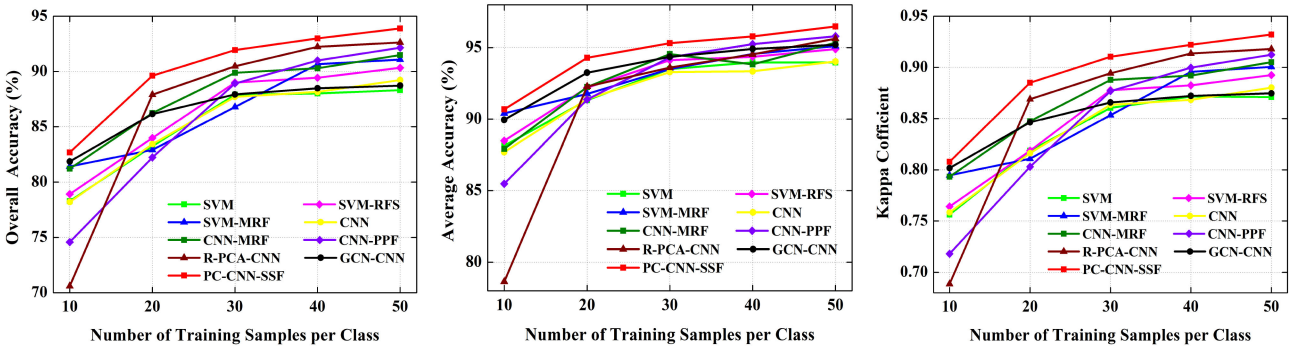


Fig. 9. Classification performances of nine algorithms with different training sample sizes on the Salinas.

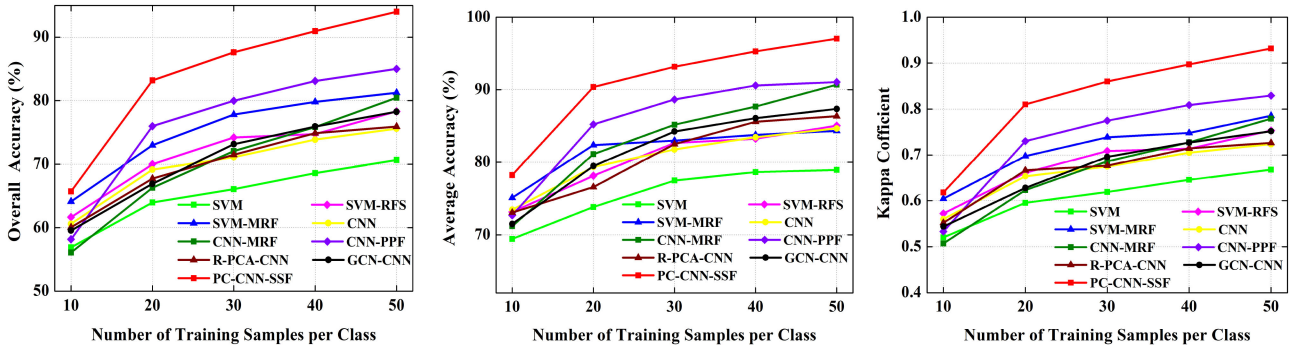


Fig. 10. Classification performances of nine algorithms with different training sample sizes on the Indian Pines dataset.

TABLE VII
CLASSIFICATION PERFORMANCES OF SPECTRAL AND SPATIAL-SPECTRAL FUSION WITH PAVIA UNIVERSITY DATASET

index	Spectral				Spectral-spatial			
	P=1	P=2	P=3	P=4	P=1	P=2	P=3	P=4
OA (%)	76.66	81.66	80.66	**82.87**	82.47	**85.77**	83.53	83.63
AA (%)	82.33	86.52	85.87	85.63	86.83	89.32	87.15	87.04
KC(×100)	70.23	76.41	75.32	77.99	77.45	81.59	78.90	78.78

Bold highlights the highest classification accuracy.

TABLE VIII
CLASSIFICATION PERFORMANCES OF SPECTRAL AND SPATIAL-SPECTRAL FUSION WITH SALINAS DATASET

index	Spectral				Spectral-spatial			
	P=1	P=2	P=3	P=4	P=1	P=2	P=3	P=4
OA (%)	84.10	83.93	85.92	**87.32**	86.58	88.48	**88.77**	88.13
AA (%)	91.60	92.09	92.68	92.81	93.07	94.01	93.67	93.12
KC(×100)	82.35	82.19	84.41	85.92	85.12	87.24	87.57	86.94

Bold highlights the highest classification accuracy.

TABLE IX
CLASSIFICATION PERFORMANCES OF SPECTRAL AND SPATIAL-SPECTRAL FUSION WITH INDIAN PINES DATASET

index	Spectral				Spectral-spatial			
	P=1	P=2	P=3	P=4	P=1	P=2	P=3	P=4
OA (%)	67.75	75.98	**79.60**	77.90	78.06	**83.18**	81.84	79.13
AA (%)	77.63	85.22	87.20	86.99	86.08	90.37	90.20	88.17
KC(×100)	63.76	73.02	77.10	75.27	75.22	81.01	79.66	76.65

Bold highlights the highest classification accuracy.

considering the 0th class also achieves higher accuracy than that without considering the 0th class. Because the label of pixels in a testing cluster may be different, the training pixel-clusters with the label $c = 0$ should be contained within the input of CNN. If the 0th class is not added when training the CNN model, testing pixel-clusters with label $c = 0$ will be classified into error classes, causing lower classification accuracy. Therefore, it is necessary to consider the 0th class when training the CNN model.

To verify the effectiveness of SSF behavior, the original spectral feature and SSF feature are further analyzed in Tables VII–IX. Note that the number of samples in a pixel cluster is set to $P = 1, 2, 3, 4$, respectively, where $P = 1$ demonstrates that no pixel-cluster algorithm is used. As can be seen, instead of just using spectral information, incorporating additional spatial information from the HSI achieves higher classification accuracy. Besides, the classification performance of the proposed algorithm is influenced by the number of pixels in the pixel cluster. When only spectral information is considered, the Pavia University and the Salinas datasets obtain the highest overall accuracy at $P = 4$, while the Indian Pines dataset obtains it at $P = 3$. When the SSF features are used for classification, the Pavia University and Indian Pines datasets achieve the highest overall accuracy at $P = 2$, and the Salinas dataset obtains the highest overall accuracy at $P = 3$. With the expansion of the training set, the classification performance based on pixel cluster is significantly higher than that of a single pixel. Therefore, the proposed pixel cluster algorithm can make up for the shortage of training samples.

IV. Conclusion

In this article, a pixel cluster CNN and SSF algorithm for HSIC with small-size samples is proposed. To improve the HSIC performance of CNN when the number of labeled pixels is limited, the spatial information is extracted from each band of the original spectral information by the GLCM. Then, the spectral information and the extracted spatial information are fused by bands superposition. Besides, a pixel cluster algorithm is proposed in this article to expand the number of training samples. Ultimately, a CNN structure is used to extract effective features, which will be used as classification rules. The results based on three standard HSIs demonstrate that the proposed algorithm achieves better classification accuracy than the SVM, SVM-RFS, SVM-MRF, CNN, CNN-MRF, CNN-PPF, R-PCA-CNN, as well as GCN-CNN.

References

[1] W. Feng, W. Huang, and W. Bao, "Imbalanced hyperspectral image classification with an adaptive ensemble method based on smote and rotation forest with differentiated sampling rates," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1879–1883, Dec. 2019.

[2] W. Feng *et al.*, "Dynamic synthetic minority over-sampling technique-based rotation forest for the classification of imbalanced hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2159–2169, Jul. 2019.

[3] C. Yu *et al.*, "Hyperspectral image classification method based on CNN architecture embedding with hashing semantic feature," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1866–1881, Jun. 2019.

[4] B. Kumar, "Hyperspectral image classification using three-dimensional geometric moments," *IET Image Process.*, vol. 14, no. 10, pp. 2175–2186, 2020.

[5] M. Chen, L. Ma, W. Wang, and Q. Du, "Augmented associative learning-based domain adaptation for classification of hyperspectral remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6236–6248, Oct. 2020.

[6] X. Wang, Y. Li, and Y. Cheng, "Hyperspectral image classification based on unsupervised heterogeneous domain adaptation cyclegan," *Chin. J. Electron.*, vol. 29, no. 4, pp. 608–614, 2020.

[7] C. Bai *et al.*, "Kernel low-rank entropic component analysis for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5682–5693, 2020.

[8] C. Chang, K. Y. Ma, C. Liang, Y. Kuo, S. Chen, and S. Zhong, "Iterative random training sampling spectral spatial classification for hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3986–4007, 2020.

[9] H. Wang, X. Wang, C. L. P. Chen, and Y. Cheng, "Hyperspectral image classification based on domain adaptation broad learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3006–3018, 2020.

[10] Q. Wang, Q. Li, H. Liu, Y. Wang, and J. Zhu, "An improved ISODATA algorithm for hyperspectral image classification," in *Proc. 7th Int. Congr. Image Signal Process.*, Oct 2014, pp. 660–664.

[11] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[12] M. Chi and L. Bruzzone, "Semisupervised classification of hyperspectral images by SVMS optimized in the primal," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1870–1880, Jun. 2007.

[13] D. J. Im and G. W. Taylor, "Semisupervised hyperspectral image classification via neighborhood graph learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1913–1917, Sep. 2015.

[14] J. Peng, Y. Zhou, and C. L. P. Chen, "Region-kernel-based support vector machines for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4810–4824, Sep. 2015.

[15] W. Feng, G. Dauphin, W. Huang, Y. Quan, and W. Liao, "New margin-based subsampling iterative technique in modified random forests for classification," *Knowl.-Based Syst.*, vol. 182, 2019, Art. no. 104845.

[16] W. Feng, W. Huang, G. Dauphin, J. Xia, Y. Quan, and H. Ye, "Ensemble margin based semi-supervised random forest for the classification of hyperspectral image with limited training data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 971–974.

[17] V. E. Garcia Millan, G. A. Sanchez-Azofeifa, and G. C. Malvarez, "Mapping tropical dry forest succession with chris/proba hyperspectral images using nonparametric decision trees," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3081–3094, Jun. 2015.

[18] M. E. Paoletti, J. M. Haut, S. K. Roy, and E. M. T. Hendrix, "Rotation equivariant convolutional neural networks for hyperspectral image classification," *IEEE Access*, vol. 8, pp. 179575–179591, 2020.

[19] C. Shi and C. Pun, "Multiscale superpixel-based hyperspectral image classification using recurrent neural networks with stacked autoencoders," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 487–501, Feb. 2020.

[20] C. Chen, Y. Ma, and G. Ren, "Hyperspectral classification using deep belief networks based on conjugate gradient update and pixel-centric spectral block features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4060–4069, 2020.

[21] H. Huang, C. Pu, Y. Li, and Y. Duan, "Adaptive residual convolutional neural network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2520–2531, 2020.

[22] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, "Automatic design of convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7048–7066, Sep. 2019.

[23] H. Wei, Y. Huang, W. Li, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, 2015.

[24] L. Windrim, A. Melkumyan, R. J. Murphy, A. Chlingaryan, and R. Ramakrishnan, "Pretraining for hyperspectral convolutional neural network classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2798–2810, May 2018.

[25] 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.

[26] R. Li, Z. Pan, Y. Wang, and P. Wang, "A convolutional neural network with mapping layers for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3136–3147, May 2020.

[27] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.

[28] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep learning ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1882–1897, Jun. 2019.

[29] Z. Ge, G. Cao, X. Li, and P. Fu, "Hyperspectral image classification method based on 2d-3 d CNN and multibranch feature fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5776–5788, 2020.

[30] Q. Li, B. Zheng, B. Tu, J. Wang, and C. Zhou, "Ensemble EMD-based spectral-spatial feature extraction for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5134–5148, 2020.

[31] L. Zou, X. Zhu, C. Wu, Y. Liu, and L. Qu, "Spectral-spatial exploration for hyperspectral image classification via the fusion of fully convolutional networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 659–674, 2020.

[32] Z. Li, F. Guo, Q. Li, G. Ren, and L. Wang, "An encoder-decoder convolution network with fine-grained spatial information for hyperspectral images classification," *IEEE Access*, vol. 8, pp. 33600–33608, 2020.

[33] X. Jiang *et al.*, "Hyperspectral image classification with capsnet and markov random fields," *IEEE Access*, vol. 8, pp. 191956–191968, 2020.

[34] S. M. Kennedy, W. Williamson, J. D. Roth, and J. W. Scrofani, "Cluster-based spectral-spatial segmentation of hyperspectral imagery," *IEEE Access*, vol. 8, pp. 140361–140391, 2020.

[35] Z. Lu, B. Xu, L. Sun, T. Zhan, and S. Tang, "3-D channel and spatial attention based multiscale spatial-spectral residual network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4311–4324, 2020.

[36] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral-spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3232–3245, May 2020.

[37] X. Li, M. Ding, and A. Pizurica, "Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2615–2629, Apr. 2020.

[38] C. Yu, R. Han, M. Song, C. Liu, and C. Chang, "A simplified 2D-3 D CNN architecture for hyperspectral image classification based on spatial-spectral fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2485–2501, 2020.
[39] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image classification with markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.
[40] 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.
[41] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
[42] J. Yu, "Texture image segmentation based on gaussian mixture models and gray level co-occurrence matrix," in *Proc. 3rd Int. Symp. Inf. Sci. Eng.*, 2010, pp. 149–152.
[43] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC- 3, no. 6, pp. 610–621, Nov. 1973.
[44] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.
[45] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.
[46] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 4959–4962.
[47] 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.

Shuxian Dong is currently working toward the Ph.D. degree in information and communication engineering with the Department of Remote Sensing Science and Technology, School of Electronic Engineering, Xidian University, Xi'an, China.

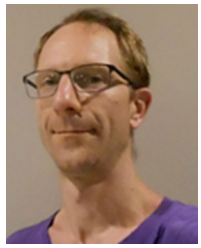
Her research interests include hyperspectral image classification and deep learning.

Yinghui Quan received the B.S. and Ph.D. degrees in electrical engineering from Xidian University, Xi'an, China, in 2004 and 2012, respectively.

He is currently a Full Professor with the Department of Remote Sensing Science and Technology, School of electronic engineering, Xidian University. His research interests include radar imaging, radar signal processing, and radar microsystem.

Wei Feng received the B.S. degree in computer science and technology from Northeast Agricultural University, Harbin, China, in 2009, the M.Sc. degree in computer applications technology from North Minzu University, Yinchuan, China, in 2013, and the Ph.D. degree in information science and technology from the Université Michel de Montaigne-Bordeaux 3, Bordeaux, France, in 2017.

She was a Postdoctoral Researcher with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China, from 2017 to 2019. She is currently a Lecturer with the Department of Remote Sensing Science and Technology, School of Electronic Engineering, Xidian University. Her current research interests include remote sensing, machine learning, and image processing.



Gabriel Dauphin received the Engineering degree in signal processing from Mines ParisTech, Paris, France, in 1996, and the Ph.D. degree in signal and image processing from Télécom Paris Tech University, Paris, France, in 2001.

Since 2002, he has been an Associate Professor with the Laboratory of Information Processing and Transmission, University Paris 13, Villetaneuse, France. His research interests include gate recognition, stereoscopic image compression, machine learning, and remote sensing.

Lianru Gao received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in cartography and geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, China, in 2007.

He is currently a Professor with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, CAS. He has authored/coauthored more than 160 peer-reviewed papers, and there are more than 80 journal papers included by Science Citation Index. His research interests include hyperspectral image processing and information extraction.

Mengdao Xing (Fellow, IEEE) was born in Zhejiang, China, in November 1975. He received the B.S. and Ph.D. degrees in electrical engineering from Xidian University, Xi'an, China, in 1997 and 2002, respectively.

He is currently a Full Professor with the Academy of Advanced Interdisciplinary Research, Xidian University. He is also with the National Key Laboratory of Microwave Imaging Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing, China. He has authored or coauthored two books and more than 200 papers. His research interests include synthetic aperture radar, inverse SAR, and sparse signal processing.