A Lightweight Convolutional Neural Network for Hyperspectral Image Classification

Sen Jia, *Senior Member, IEEE*, Zhijie Lin, Meng Xu, *Member, IEEE*, Qiang Huang, Jun Zhou, *Senior Member, IEEE*, Xiuping Jia, *Senior Member, IEEE*, and Qingquan Li

***Abstract*— In the hyperspectral image, each pixel corresponds to a small area on the Earth's surface and represents the intrinsic characteristic of objects, which can be applied for recognition of land covers. Nevertheless, hyperspectral image processing should face some critical issues, and a small sample set problem may be the most challenging one in the research. Deep learning (DL), which has successfully been applied in many fields, has also been introduced for hyperspectral image classification. However, the large gap between the massive parameters to be tuned and limited labeled samples can lead to overfitting scenario, inevitably deteriorating the generalization ability of the DL model. In this article, a lightweight convolutional neural network (LWCNN) is proposed for hyperspectral image classification to mainly tackle the small sample set problem. Especially, spatial–spectral Schroedinger eigenmaps (SSSE) feature extraction is first adopted to obtain the joint spatial–spectral information, and the compressed dimensionality could significantly reduce the number of parameters in the following DL model. Second, a dual-scale convolution (DSC) module is carefully designed to address the SSSE features from a 1-D vector viewpoint (the number of parameters is further decreased), and the DSC procedure is successively employed to obtain the hierarchical structure description that could represent data distribution from different aspects. Subsequently, the feature vectors from all DSC layers are separately filtered by a new bichannel fusion (BCF) module, which could well encode both the intrinsic and contextual information inside DSC features. Finally, the filtered features are concatenated together and imported into a global average pooling classifier to achieve the predicted probability of each category. Experimental results on three famous hyperspectral image data sets illustrate that the developed LWCNN approach is advantageous in both the efficiency and robustness sides for hyperspectral image classification tasks and outperforms other state-of-the-art methods (both traditional-based and DL-based) with very limited labeled samples.**

***Index Terms*— Deep learning (DL), hyperspectral imagery.**

I. Introduction

HYPERSPECTRAL sensor can capture spectral and spatial information of the observing object simultaneously and provide a hyperspectral data cube, called a hyperspectral image. Generally, the hyperspectral image contains hundreds of narrow spectral bands ranging from visible to near-infrared and can be naturally used to identify the various materials, which is one of the most important techniques in many areas [1]–[3]. However, the high dimensionality with a small number of labeled samples may lead to the Hughes phenomenon [4]. Correspondingly, band selection/feature extraction methods have been extensively studied, in which the former tries to pick out the most representative or discriminative bands directly from the raw hyperspectral data [5], while the latter aims to find an appropriate transformation to map the high-dimensional data into low-dimensional space, including principal component analysis (PCA) [6]–[8], independent component analysis (ICA) [9]–[11], and local linear embedding (LLE) [12], [13]. Alternatively, since spatial consistency can often be observed in the hyperspectral image (which means that pixels in the adjacent spatial region have a high possibility within the same class), the works that exploit the spatial correlation have also been researched, including gray-level co-occurrence matrix (GLCM) [14], [15], extended morphological profiles (EMPs) [16], sparse representation [17], [18], and superpixel-based methods [19], [20].

Nowadays, the most attracted strategy for hyperspectral image classification is to take advantage of both spectral and spatial information together. Concretely, multiple kernel learning with nonlinear description [21] and superpixel guidance [22] are applied to extract spatial–spectral features. In [23], edge-preserving filtering is utilized as a postprocessing technique to improve the probability output of the support vector machine (SVM) classifier. Besides, a number of 2-D operators have been extended to 3-D domain, such as 3-D GLCM [24], 3-D local binary pattern (LBP) [25], 3-D wavelet transform [26], [27], and 3-D Gabor wavelet [28]–[30], and the internal spatial–spectral structure can be well characterized.

Due to the versatility and huge representation capacity of the deep learning (DL) model, while hyperspectral image classification is similar to traditional computer vision tasks,

Manuscript received January 20, 2020; revised May 5, 2020 and July 4, 2020; accepted July 29, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41971300, Grant 61671307, and Grant 61901278; in part by the Program for Young Changjiang Scholars; and in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20180305124802421 and Grant JCYJ20180305125902403. *(Corresponding author: Qiang Huang.)*

Sen Jia, Zhijie Lin, Meng Xu, and Qiang Huang are with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, Shenzhen 518060, China, also with the SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, China, also with the Guangdong Key Laboratory of Urban Informatics, Shenzhen University, Shenzhen 518060, China, and also with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: senjia@szu.edu.cn; zigitlam@yeah.net; m.xu@szu.edu.cn; jameshq@szu.edu.cn).

Jun Zhou is with the School of Information and Communication Technology, Griffith University, Nathan, QLD 4111, Australia (e-mail: jun.zhou@griffith.edu.au).

Xiuping Jia is with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2612, Australia (e-mail: x.jia@adfa.edu.au).

Qingquan Li is with the Guangdong Key Laboratory of Urban Informatics, Shenzhen University, Shenzhen 518060, China (e-mail: liqq@szu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.3014313

DL-based methods have been introduced into this field [31], [32]. Generally, PCA is first applied in the preprocessing stage, and the compressed spectral and spatial information is exploited by DL model [33]. In [34], an effective classification framework that combined deep belief network (DBN) with active learning is developed. The weighted incremental dictionary learning (WI-DL) algorithm is designed to actively select additional representing samples from the unlabeled data set, and DBN provides the final prediction. Meanwhile, Zhong *et al.* [35] designed a diversified DBN to improve the classification performance by regularizing the pretrained procedure of DBN. In [36], the balanced local discriminant embedding (BLDE) and a simple CNN model are integrated to extract spectral and spatial features, respectively. In particular, compared with the 2-D operator that only exploits the spatial domain, the 3-D operator can better characterize the spatial–spectral correlation of the hyperspectral image. Zhang *et al.* [37] utilized a 3-D generative adversarial network (GAN) to construct a spectral–spatial classifier. The GAN framework uses a CNN to discriminate the inputs (discriminative model), and another CNN is designed to generate the fake input (generative model). Li *et al.* [38] introduced 3DCNN that is mainly designed for video-based applications to learn the local signal change in the spectrum and spatial dimensions without any data transformation.

Since each spatial pixel in the hyperspectral image corresponds to a spectral vector with hundreds of bands, the DL methods that deal with sequential data can naturally be employed. Hu *et al.* [39] proposed a deep convolutional network that tries to use a 1-D convolution operator to exploit the deep feature of the spectrum. Besides, Zhu *et al.* [40] designed a 1-D GAN for hyperspectral image classification. Xue *et al.* [41] combined the CapsNet with Triple-GANs to form a classification scheme where Triple-GANs expand the training set by generating additional virtual sample. Furthermore, 1-D and 2-D CapsNet are, respectively, adopted to extract the spectral and spatial features at the shallow layer, and the classification is carried out by a dense capsule layer [42]. Especially, the recurrent neural network (RNN), which is a powerful tool for sequential data processing, such as speech recognition [43], machine translation [44], and video behavior recognition [45], has recently been incorporated for hyperspectral image classification. Wu and Prasad [46] proposed a deep convolutional RNN for hyperspectral image classification, in which a convolutional operator is used to extract middle-level invariant local features from the original spectral sequence and then recurrent layer extracts contextual feature from the output of convolution layer. Besides, Xu *et al.* [47] integrated RNN with 2DCNN to build a uniform framework where RNN is only applied to the spectral domain. Concretely, the spectrum is divided into several segments with equal length and entered into RNN, while 2DCNN is responsible to acquire the spatial information. Although most DL-based methods described earlier have obtained good performance, a large number of manually labeled samples is usually required to well train the model (for example, 2DCNN model contains more than 60 000 parameters for the classic Indian Pines hyperspectral image data), which is unrealistic in practical applications and significantly weakens the practicability of the DL-based model.

With respect to the small training sample issue of hyperspectral image, many DL-based methods have been proposed in recent years. In MugNet [48], rolling guidance filtering was adopted as the preprocessing step to avoid the infection of noise and small meaningless detail. In the following, PCANet [49] was used to integrate the multigrain and semisupervised information. In [46], the superpixel segmentation and Dirichlet process mixture model were used to produce the pseudolabels, and a semisupervised CNN was trained with these pseudolabels. Then, the classifier was fine-tuned with the truth labels. Liu *et al.* [50] adopted the deep model to map the sample vector into a metric embedding and used the Euclidean distance to measure the distance between them. The nearest neighbor classifier is utilized to reduce the parameter size of the model. Alternatively, the lightweight DL model can be a feasible way to tackle the small training sample issue. Su *et al.* [51] simply extended the 2-D depthwise separable convolution to 3-D convolution to construct a deep lightweight model, while Zhang *et al.* [52] designed a special 3-D depthwise separable convolution for hyperspectral image classification, and its lightweight model is trained with transfer learning. In [53], squeeze and excitation operations were used to discard the meaningless features so that the kernel number of the following convolution can be correspondingly reduced. Liu *et al.* [54] proposed a lightweight model named SG-CNN that applied group convolution and channel shuffle to reuse the extracted features.

In this article, we aim to construct a lightweight convolutional neural network (LWCNN) for hyperspectral image classification to deal with the small sample set problem and increase the generalization ability of the DL model. First, spatial–spectral Schroedinger eigenmaps (SSSE) operator [55], which is derived from Laplacian eigenmaps (LE), is employed to extract the fused spectral–spatial features from the raw hyperspectral imagery, and the joint spectral–spatial information can be well concentrated on the pixel level. Second, a 1-D convolution operator is utilized rather than the traditional 2-D one, and thus, the number of parameters can be greatly reduced. Meanwhile, a dual-scale convolution (DSC) module with two receptive fields is introduced to extract the discriminative features from the SSSE encoding vectors. Furthermore, in order to obtain the hierarchical features of data representation, a series of the designed DSC modules is successively applied. Third, inspired by the DL model in natural language processing [56], the feature vectors obtained from all DSC layers are separately filtered by a new bichannel fusion (BCF) module instead of simply fusing the features in elementwise form, which could achieve the weighted fusion and encode the contextual information inside DSC features simultaneously. Finally, the filtered features are concatenated together and put into a global average classifier to obtain the classification score. To make the proposed LWCNN framework easier to be understood, Fig. 1 shows the sketch map, in which the Indian Pines hyperspectral image is taken into account. In particular, the major contributions of our work are summarized as follows.

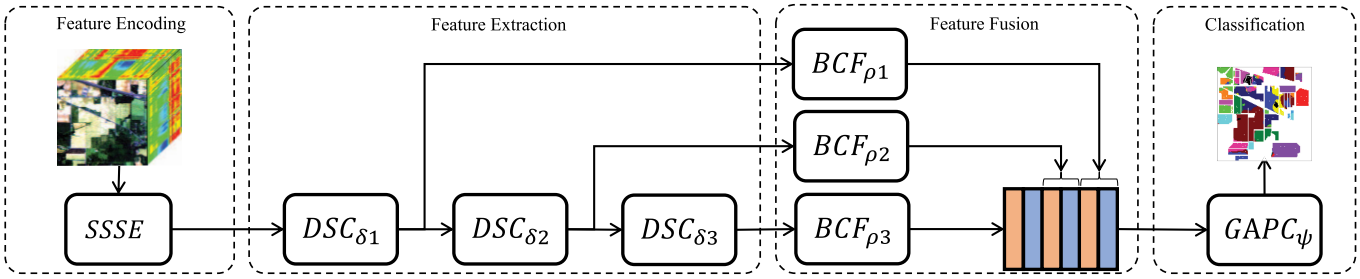


Fig. 1. Flowchart of the proposed LWCNN for hyperspectral image classification.

1) First, the 1-D vector input of the DSC module can not only provide a lightweight way to address the small sample set problem but also ensure the efficiency of the proposed method. Meanwhile, layer normalization (instead of batch normalization) is employed to comply with the small sample set scenario, and thus, the statistical significance of feature can be well reserved.
2) Second, concerning the BCF module, a gate recurrent unit (GRU) is incorporated to exploit the potential contextual information of DSC feature vectors, while the convolution operator is utilized to establish the linear combination of features, making the subsequent classification procedure more elegant. Through concatenating all the bichannel features together, the feature representativeness can be largely enhanced.
3) Third, the SSSE operator is introduced to reduce the dimension of the hyperspectral image, which can effectively represent the joint spectral–spatial structure of objects, and thus, the following 1-D oriented feature extraction procedure can be more distinct. Besides, a global average pooling scheme is introduced in the classification stage to increase the robustness of the proposed framework.

The rest of this article is organized as follows. Section II briefly introduces Schroedinger eigenmaps (SE), convolution neural network, and RNN. The proposed MLCNN framework is presented with four parts in Section III. The hyperspectral image data sets used in this study and the experimental results are provided in Section IV. Conclusions are summarized in Section V.

II. Related Works

A. SSSE

LE [57], [58] is a nonlinear dimensionality reduction method and could construct the intrinsic geometric manifold structure with high efficiency. It considers the manifold structure in high-dimensional space and preserves local neighborhood information in low-dimensional space. However, it only reflects one aspect of hyperspectral image (either spectral or spatial information); thus, the SSSE is introduced [59].

Let $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$ denote the original hyperspectral data cube, where H and W are the height and width in the scanned scene, while B is the number of bands. The SSSE operator is a generalization of LE and incorporates a potential matrix with LE. Especially, the SSSE procedure can be depicted as follows.

1) Construct an adjacency graph $\mathbf{G}$ from $\mathbf{X}$. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are in spatial neighborhood, then there is an edge $\mathbf{G}_{i,j}$ between the two points, i.e., $\mathbf{G}_{i,j}$ is set as 1 (otherwise, it equals to zero). The neighborhood relation is decided by ϵ-neighborhoods, i.e., the location distance between two points is less than a predefined threshold ϵ.
2) Calculate the weight matrix $\mathbf{W}$ and the Laplacian matrix $\mathbf{L}$. Heat kernel is a commonly used method to compute the weight

$$\mathbf{W}_{i,j} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma), & \mathbf{G}_{i,j} > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

It is obvious that the spectral information is fully utilized to characterize the similarity of pixels. Furthermore, a diagonal matrix is defined as $\mathbf{E}_{i,i} = \sum_j \mathbf{W}_{i,j}$, and the Laplacian matrix $\mathbf{L}$ is simply computed as $\mathbf{L} = \mathbf{E} - \mathbf{W}$.
3) Compute the cluster potential matrix $\mathbf{V}$. Here, the spatial locations of pixels are taken into account, and $\mathbf{V}_{i,j}$ is calculated as

$$\mathbf{V}_{i,j} = \sum_{\mathbf{x}_j \in N_\epsilon(\mathbf{x}_i)} \mathbf{S}_{(i,j)} \cdot \gamma_{i,j} \cdot \exp\left(-\frac{\left\|\mathbf{x}_i^p - \mathbf{x}_j^p\right\|^2}{\sigma}\right) \tag{2}$$

where $\mathbf{x}_i^p$ and $\mathbf{x}_j^p$ are the spatial positions of $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. $\mathbf{S}$ is a sparse matrix that describes the relation of spatial position of pixels, and parameter γ represents the correlation of pixels in neighborhood [59].
4) Implement the feature extraction of SSSE. The eigenvalues and eigenvectors of the following formula are investigated:

$$(\mathbf{L} + \alpha\mathbf{V})\mathbf{f} = \lambda\mathbf{E}\mathbf{f} \tag{3}$$

where α is a balance coefficient. In contrast to the PCA method, here, the eigenvectors corresponding to the smallest K eigenvalues are reserved, and $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K] \in \mathbb{R}^{HW\times K}$ is the obtained feature data. It is clear that $\mathbf{F}$ can be reshaped into 3-D form, denoted as $\mathbf{H} \in \mathbb{R}^{H\times W\times K}$.

Fig. 2 shows the basic procedure of SSSE.

B. CNN

The conventional CNN (as shown in Fig. 3) mainly relies on two components, convolution layer and pooling layer, to extract hierarchical deep feature automatically, and the dropout technique is utilized to alleviate the overfitting phenomenon. The pooling layers have two main functions, which can not only reduce the map size quickly and improve computation efficiency but also enlarge the receptive field of the

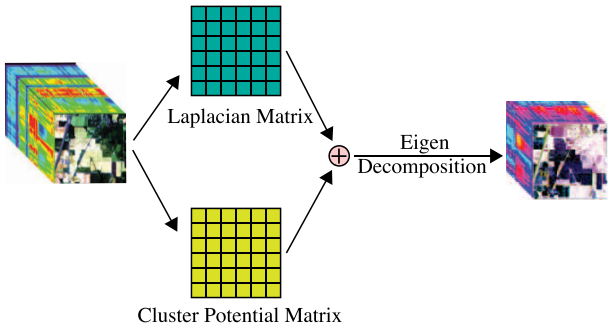


Fig. 2. SSSE.

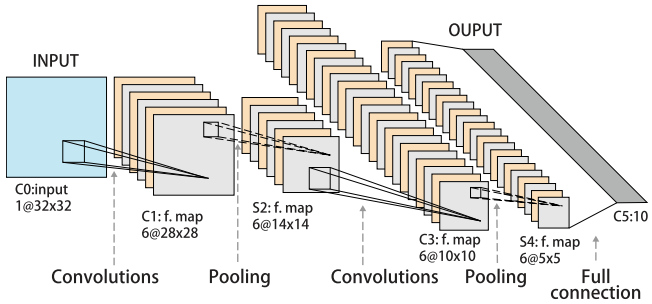


Fig. 3. Structure of CNN.

kernels in the input or feature maps (that come from the previous convolution layer).

The number of layers in the model is an important factor that controls the learning capacity of CNN. The kernels in the shallow layer are mainly responsible for detecting texture features, while the convolution operators in the deeper layer use this shallow information to build up abstract feature maps. As the layers get deeper, the features would become more abstract and have stronger representation ability.

C. RNN

RNN has a self-loop connection and maintains an internal hidden state. At each time step $t \in \{1, 2, \dots\}$, for an input vector $\mathbf{e}_t$, the RNN network will output a vector $\mathbf{o}_t$ that characterizes the contextual information. Meanwhile, a hidden state vector $\mathbf{h}_t$ is also taken into account and enters into the network again with the new input in the next step. In this way, RNN can learn the related features between the items of the input sequence. The procedure of RNN can be formulated as follows:

$$\mathbf{h}_t = \mathcal{F}(\mathbf{U}_e \cdot \mathbf{e}_t + \mathbf{U}_h \cdot \mathbf{h}_{t-1}), \quad t = 2, \dots \tag{4}$$

$$\mathbf{o}_t = \mathcal{F}(\mathbf{U}_o \cdot \mathbf{h}_t) \tag{5}$$

where $\mathbf{U}_e$, $\mathbf{U}_h$, and $\mathbf{U}_o$ are the transformation matrices of the corresponding variables, respectively, and can be learned in the training process of the RNN model. $\mathbf{h}_1$ is randomly initialized by a Gaussian distribution. Besides, $\mathcal{F}$ is the nonlinear activation function, such as Tanh, Sigmoid, and ReLu.

III. Proposed Method

In this section, the proposed LWCNN framework is depicted in four modules: SSSE-based preprocessing, DSC, BCF, and global average pooling.

A. SSSE-Based Preprocessing

The data samples of the hyperspectral image observed from the real world normally have high spectral dimensionality, whereas large redundancy and noisy bands can impact the efficiency of data processing. Meanwhile, CNN-based models generally need more labeled samples to well train the enormous parameters with the raw hyperspectral image, so dimensionality reduction is usually applied in advance. Although there are numerous dimensionality reduction methods in the literature, the SSSE is incorporated in our work, which is mainly due to the following two reasons.

1) On the one hand, the SE is calculated on the graph that is constructed from the spectral information of the hyperspectral image. On the other hand, the spatial proximity is encoded with a cluster potential matrix that is derived from the LE operator (a detailed description can be found in Section II-A and Fig. 2). Therefore, the joint spatial–spectral features can be adequately exploited by the SSSE method, and the performance of the proposed LWCNN framework can be guaranteed.
2) Different from the other methods (such as the 3-D filtering-based ones) that should extract a huge amount of features to characterize the spatial–spectral structure of the hyperspectral image, the SSSE operator maps raw hyperspectral image into low-dimensional space and preserves the local Euclidean characteristic of data. Since the dimensionality of the achieved SSSE features is significantly decreased, the parameter volume needed to be tuned in the LWCNN model is correspondingly compressed, and thus, the robustness of the proposed method can be enhanced. Meanwhile, the computational load and storage requirements are also reduced.

As presented in Section II-A, the SSSE operator is directly applied to the raw hyperspectral image $\mathbf{X} \in \mathbb{R}^{H \times W \times B}$, and the achieved SSSE feature is denoted as $\mathbf{H} \in \mathbb{R}^{H \times W \times K}$, where K is the number of reserved features. Since K is much smaller than B, the data volume of SSSE feature $\mathbf{H}$ is greatly decreased, and the network structure of the proposed LWCNN model is simplified.

B. DSC

After the spatial–spectral related information has been well concentrated in the pixel level of SSSE features, only 1-D-oriented signal convolution is considered rather than 2-D operation, which can greatly decrease the model complexity, and the performance can also be improved. Before presenting the carefully designed DSC module, some mathematical notations are explained. Based on the SSSE feature cube $\mathbf{H} \in \mathbb{R}^{H \times W \times K}$ achieved earlier and suppose there are C materials existed in the scene, $\mathbf{A} \in \mathbb{R}^{K \times n}$ denotes the training set, where n is the number of training samples. Generally, in order to accelerate the learning efficiency of CNN-based methods, the training set is randomly divided into J subgroups given a predefined batch size S, i.e., $J = \lceil n/S \rceil$, where $\lceil \cdot \rceil$ is the rounding up operator. Hereafter, the description of the proposed LWCNN framework is from group $\mathbf{A}_j \in \mathbb{R}^{K \times S}$, $j = 1, \dots, J$ viewpoint rather than a single vector. Besides, small Greek letters, such as δ and θ, are used to stand

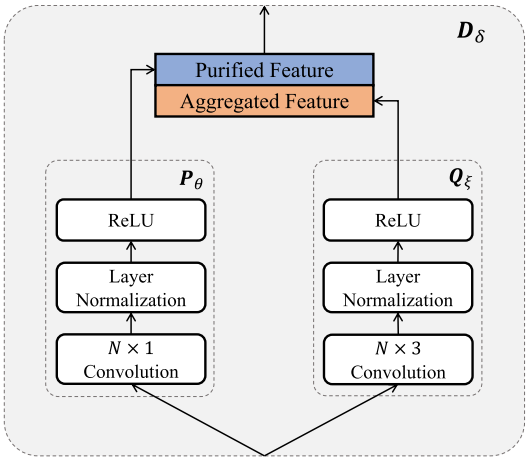


Fig. 4. DSC module. Note that the input of the two branches $\mathbf{P}_\theta$ and $\mathbf{Q}_\xi$ is the extracted SSSE spectral feature of each spatial pixel, and the kernel size of the second branch $\mathbf{Q}_\xi$ is $N \times 3$, which aggregates the spectral neighboring information of feature vector.

for the parameters of each module, which can be learned in the training process.

As shown in Fig. 4, the DSC module $\mathbf{D}_\delta$ mainly contains two separate branches, including the purified feature $\mathbf{P}_\theta$ and aggregated feature $\mathbf{Q}_\xi$, which, respectively, aims to use the 1-D kernels of the different receptive field to enhance the difference of various samples. Especially, in the first branch $\mathbf{P}_\theta$, the size of the kernel in filter $\mathcal{P}$ is set to be one ($\mathcal{P} \in \mathbb{R}^{N\times 1}$, where N is the filter size) so that the filtering operator could focus on the single location of all channels and not be interfered by the spectral neighborhood. Alternatively, the size of the kernel in filter $\mathcal{Q}$ of the second branch $\mathbf{Q}_\xi$ is set as three ($\mathcal{Q} \in \mathbb{R}^{N\times 3}$) so that our model has the ability to aggregate the spectral neighborhood information of feature vector. Mathematically

$$\mathbf{B}_j = \mathcal{P} \otimes \mathbf{A}_j^T, \quad j = 1, \ldots, J \tag{6}$$

$$\mathbf{C}_j = \mathcal{Q} \otimes \mathbf{A}_j^T, \quad j = 1, \ldots, J \tag{7}$$

where $\otimes$ is the convolutional operator and $(\cdot)^T$ is the matrix transformation. As a result, the dimensions of convolutional results $\mathbf{B}_j$ and $\mathbf{C}_j$ are the same and become $S \times N \times K$.

After that, the sample normalization procedure is conventionally applied before the ReLu activation function to avoid gradient vanishing phenomenon. Generally, the normalization is carried out on the first dimension, which is just the so-called batch normalization and can be formalized as follows:

$$\mathbf{B}_j^{BN} = \beta \frac{\mathbf{B}_j - \text{mean}(\mathbf{B}_j, 1)}{\text{std}(\mathbf{B}_j, 1)} + \eta, \quad j = 1, \ldots, J \tag{8}$$

where mean($\mathbf{B}_j$, 1) and std($\mathbf{B}_j$, 1), respectively, denotes the mean value and standard variation of $\mathbf{B}_j$ along the first dimension, and β and η are two weighting parameters. It is obvious that the values of β and η have an important influence for the learning procedure. As far as the small sample set problem is concerned, since the batch size S could not be large enough, the stability and preciseness of the both parameters cannot be ensured under batch normalization.

Fortunately, inspired by nature language processing, the layer normalization method is incorporated in our work,

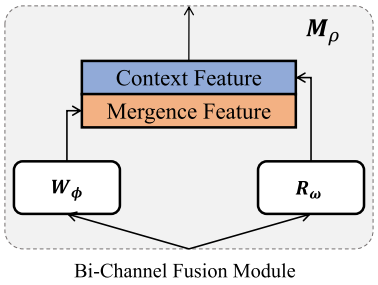


Fig. 5. BCF module.

which can be described as

$$\mathbf{B}_j^{LN} = \beta \frac{\mathbf{B}_j - \text{mean}(\mathbf{B}_j, 3)}{\text{std}(\mathbf{B}_j, 3)} + \eta, \quad j = 1, \ldots, J \tag{9}$$

where mean($\mathbf{B}_j$, 3) and std($\mathbf{B}_j$, 3), respectively, are the operations of calculating mean value and standard variation of $\mathbf{B}_j$ along the channel dimension K. It can be seen from (9) that the operations of mean($\mathbf{B}_j$, 3) and std($\mathbf{B}_j$, 3) in layer normalization only focus on one sample at a time, and thus, each sample has an independent description and is not affected by the batch size. Since the samples belonging to the same category generally have stable mean values and standard deviation, the parameters β and η in the layer normalization can be estimated more precisely. A numerical comparison of the two normalization methods is provided in Section IV. The layer normalization of $\mathbf{C}_j$ can be computed in the same way, which is expressed as $\mathbf{C}_j^{LN}$.

After applying the ReLu activation function on the two normalized feature cubes ($\mathbf{B}_j^{LN}$ and $\mathbf{C}_j^{LN}$), they are simply concatenated together to extract the dual-scale features, and the DSC feature $\mathbf{D}_j \in \mathbb{R}^{S\times 2N\times K}$ is obtained

$$\mathbf{D}_j = \text{cat}\big(\text{ReLu}\big(\mathbf{B}_j^{LN}\big), \text{ReLu}\big(\mathbf{C}_j^{LN}\big), 2\big), \quad j = 1, \ldots, J. \tag{10}$$

Meanwhile, in order to acquire sufficient hierarchical features of data structure, the DSC module is successively employed for three times (the number of DSC module is analyzed in Section IV), as shown in Fig. 1, and the DSC features $\mathbf{D}_j^{(i)}, i = 1, 2, 3$ are achieved with different parameters $\delta_i, i = 1, 2, 3$.

C. BCF

It is clear that the volume of features extracted by the abovementioned DSC module $\mathbf{D}_j^{(i)}, i = 1, 2, 3$ is huge. When it comes to the small sample set problem, the large number of parameters in the classifier will make the classification model very hard to train and lead to serious overfitting. On the other hand, in most conventional CNN model, such highly abstract features obtained from previous convolutional modules are flattened and then concatenated together to construct a long vector before entering into the classifier, which could lose the contextual information between features.

To largely reduce the number of parameters in the classification phase and increase the learning efficiency as well, a BCF $\mathbf{M}_\rho$ module is proposed to exploit the contextual information between features and accomplish the weighted mergence of representative information. The structure of BCF is shown in Fig. 5.

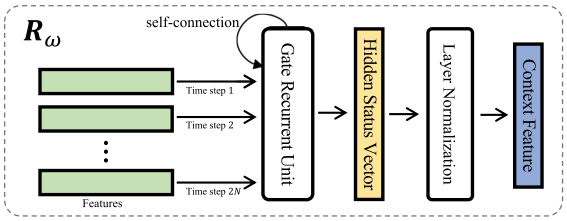


Fig. 6. Detail structure of the context branch in the BCF module.

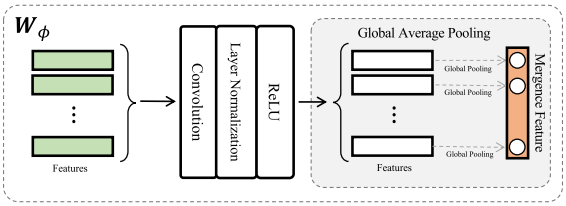


Fig. 7. Detail structure of the mergence branch in the BCF module.

1) Context Branch: From the natural language processing viewpoint, the features extracted by the DSC module can be regarded as a set of word embedding, and the potential contextual information between the features should be investigated to enhance the discriminative ability. As shown in Fig. 6, the DSC features $\mathbf{D}_j \in \mathbb{R}^{S\times 2N\times K}$, $j = 1, \ldots, J$ successively enter into a GRU at each time step. Consequently, the hidden status vector is used to encode all the expected contextual information, which usually has the size of 80% of the input size K, i.e., $L = \lceil 0.8K \rceil$. It can compress the most useful information and discard the redundant ones as well. Similarly, the hidden status vector of GRU in the last step will be managed by the abovementioned layer normalization, and the output $\mathbf{R}_j \in \mathbb{R}^{S\times 1\times L}$, $j = 1, \ldots, J$ is the extracted context feature.

2) Mergence Branch: Since the importance of features extracted by the DSC module is different from each other, it is desirable to introduce a weighted representation rather than taking all the features into account equally. More precisely, instead of simply flattening or elementwise adding the DSC features, the DSC features $\mathbf{D}_j$, $j = 1, \ldots, J$ are convolved with a filter (the kernel size is 1) to change the number of the feature vector, which equals to the length of hidden status vector of the context branch (i.e., L). Global average pooling is subsequently applied to each individual feature and transforms all of them into a 1-D vector, which is shown in Fig. 7, and the obtained mergence feature is $\mathbf{W}_j \in \mathbb{R}^{S\times 1\times L}$, $j = 1, \ldots, J$.

The features obtained from the contextual module $\mathbf{R}_j \in \mathbb{R}^{S\times 1\times L}$, $j = 1, \ldots, J$ and the mergence module $\mathbf{W}_j \in \mathbb{R}^{S\times 1\times L}$, $j = 1, \ldots, J$ are stacked together to formulate the output of the BCF module

$$\mathbf{M}_j = \text{cat}(\mathbf{R}_j, \mathbf{W}_j, 2), \quad j = 1, \ldots, J. \tag{11}$$

In order to make full use of the abstract information obtained from the three DSC modules $\mathbf{D}_j^{(i)}, i = 1, 2, 3$, an individual BCF module is configured for each DSC module, and all the outputs of these three BCF modules $\mathbf{M}_j^{(i)}, i = 1, 2, 3$, are concatenated together (as shown in Fig. 1)

$$\mathbf{M}_j^{\text{all}} = \text{cat}\big(\mathbf{M}_j^{(1)}, \mathbf{M}_j^{(2)}, \mathbf{M}_j^{(3)}, 2\big) \tag{12}$$

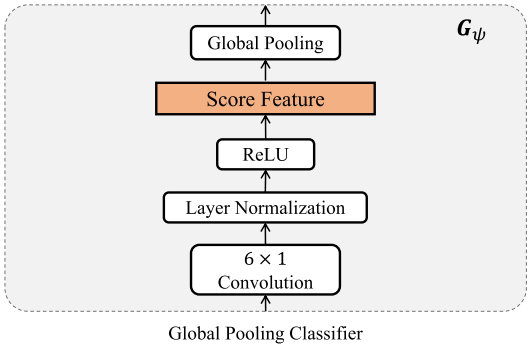


Fig. 8. Global average pooling classifier.

where $\mathbf{M}_j^{\text{all}} \in \mathbb{R}^{S\times 6\times L}$ is the input feature for the following classification modules.

D. Global Average Pooling Classifier

Conventional CNN uses a convolution operator to learn massive abstract feature maps from the input data, and then, the fully connected layers, also called dense layers, are used to map the feature map into the sample label space whose dimensions are equal to the number of class. Due to the dense connection, the fully connected layers contain a huge number of parameters and are prone to overfitting, especially in the case of a small sample set. In addition, the strong mapping capacity of the fully connected layers is excess for the classification task with limited labeling samples, that is to say, there are a lot of redundant parameters in dense layers, which could increase the cost of learning process and size of the model. Fortunately, the global average pooling classifier [60] that integrates the convolution operation and global average pooling can be a reasonable solution because the characteristic of shared weights and biases could greatly decrease the volume of parameters in the model. Fig. 8 shows the detailed structure of the global average pooling classifier.

To easily demonstrate the classification procedure, one training sample $\mathbf{m}_i \in \mathbb{R}^{6\times L}, i = 1, 2, \ldots, S$, is picked out from the batch $\mathbf{M}_j^{all}$. First, in our work, the kernel size of convolution is fixed to 1, and the convolution operation is equivalent to the linear combination of the features. Recall that C is the number of categories, and the kernel number is equal to C. The set of convolution operators is formalized as $\mathbf{\Psi} = \{\Psi_1, \ldots, \Psi_C\}$. Consequently, for the cth kernel $\Psi_i \in \mathbb{R}^6$, the convolutional value of $\mathbf{m}_i$ is calculated as follows:

$$\mathbf{z}_c = \mathbf{m}_i^T \times \Psi_c, \quad c = 1, \ldots, C. \tag{13}$$

All the convolution results $\mathbf{z}_c \in \mathbb{R}^L$ of these kernels are stacked, normalized, and activated to form the score features $\mathbf{Z} = [\mathbf{z}_1^T, \ldots, \mathbf{z}_C^T] \in \mathbb{R}^{C\times L}$, which represents the classification confidence information, and each feature $\mathbf{z}_c, c = 1, \ldots, C$ in $\mathbf{Z}$ corresponds to one class.

Second, the global average pooling is developed to calculate the average of each individual feature. It brings a strong prior into the network without extra parameters that the average of each feature represents the classification probability confidence, which enforces the previous convolution operation to map high-level abstract feature into category space. With respect to the training stage, the softmax function [61] accepts

the output of the layer and produces the category probability, while the loss value of the model is calculated as

$$[\hat{y}_1, \ldots, \hat{y}_C] = \text{softmax}(\text{mean}(\mathbf{Z})) \tag{14}$$

$$y_c = \begin{cases} 1, & \text{Class}(\mathbf{m}_i) = k \\ 0, & \text{other}, \end{cases} \quad c = 1, \ldots, C \tag{15}$$

$$\text{Loss}(\mathbf{m}_i) = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{16}$$

where mean($\mathbf{Z}$) outputs the mean value of each column of $\mathbf{Z}$, and the label of $\mathbf{m}_i$ is supposed to be k. $y_c (c = 1, \ldots, C)$ denotes a one-hot vector of $\mathbf{m}_i$, while $\hat{y}_c (c = 1, \ldots, C)$ is the classification probability confidence. The Loss($\mathbf{m}_i$) is used to compute the gradient and update the parameters of the whole network.

Once the training process of our LWCNN model is accomplished, the label of a test sample $\mathbf{y} \in \mathbb{R}^K$ is predicted as

$$\text{Class}(\mathbf{y}) = \underset{c \in \{1,\ldots,C\}}{\arg\max} \ (\text{mean}(\mathbf{Z})). \tag{17}$$

Finally, the time complexity of the proposed LWCNN method is analyzed, which can be roughly divided into three parts. The first part is related to the feature decomposition of SSSE, which is $O((XY)^3)$ (X and Y denote the two spatial dimensions of the hyperspectral image). The second part is related to the DSC module, which is $O(K)$ (K is the number of reserved components in the SSSE method). The third part is related to the BCF module, which is $O(K^2)$. Especially, it can be easily found that the SSSE procedure is unrelated to the training set (which is computed only once), while the parameter K is far less than the spatial coverage of hyperspectral image $X \times Y$; therefore, the proposed LWCNN approach is applicable to hyperspectral image with large spatial size.

IV. Experiment

A. Experiment Data Sets

The first data set is the Indian Pines data set that was acquired by the AVIRIS sensor and contains 10 366 labeling pixels with 16 ground-truth classes. The spatial resolution is as low as 20 m per pixel, and the spatial dimension of the data is 145 × 145. It contains 2/3 agriculture and 1/3 forest or other natural perennial vegetation. Since the data are collected in June 1992, some of the crops present in the scene, such as corn and soybean, are in the early stages of growth with less than 5% coverage. Within the original 224 bands, four zero bands and 35 lower SNR bands affected by atmospheric absorption have been discarded in the experiments; thus, the rest 185 bands are preserved. Its ground-truth and detailed information per class are shown in Fig. 9 and Table I.

The second hyperspectral data set was acquired in the area of Pavia University (PaviaU), Northern Italy, by using the ROSIS sensor during a flight campaign. This data set consists of 610 × 340 pixels, and 42 776 samples are labeled from nine different classes. The geometric resolution of the data set is as high as 1.3 m. The raw hyperspectral image data contain 115 spectral bands. After removing the 12 noisy bands, the remaining 103 channels are processed. Fig. 10 and Table II list the details of the data set.

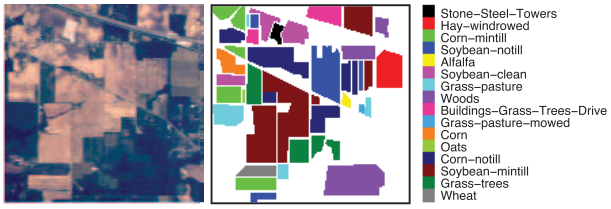


Fig. 9. (Left) False-color image and (Right) ground-truth map of the Indian pines data set.

TABLE I
LAND COVER CLASSES WITH NUMBER OF SAMPLES FOR THE INDIAN PINES DATA SET

Class	Land Cover Type	No. of Samples
$C1$	Stone-steel-towers	95
$C2$	Hay-windrowed	489
$C3$	Corn-min Till	834
$C4$	Soybean-no Till	968
$C5$	Alfalfa	54
$C6$	Soybean-clean Till	614
$C7$	Grass/Pasture	497
$C8$	Woods	1294
$C9$	Bldg-Grass-Tree-Drives	380
$C10$	Grass/Pasture-mowed	26
$C11$	Corn	234
$C12$	Oats	20
$C13$	Corn-no Till	1434
$C14$	Soybean-min Till	2468
$C15$	Grass/Trees	747
$C16$	Wheat	212
	Total	10366

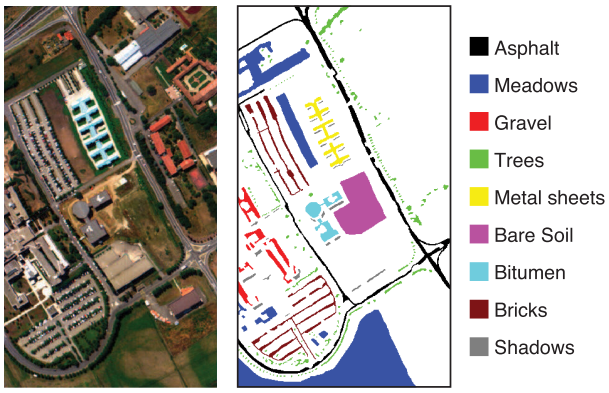


Fig. 10. (Left) False-color image and (Right) ground-truth map of the PaviaU data set.

The third hyperspectral data set was gathered by AVIRIS sensor over Salinas Valley, CA, USA, and consists of 512 × 217 pixels with a high spatial resolution of 3.7 m per pixel, as shown in Fig. 11. After 20 noisy bands are discarded from the original 224 bands, including bands [108–112], bands [154–167], and band 224, 204 bands are reserved. The ground-truth map contains 54 129 labeling samples belonging to 16 classes [see Table III].

B. Parameter and Module Analysis

In this section, we will first analyze the parameter setting for the proposed LWCNN method. Although most parameters

TABLE II
LAND COVER CLASSES WITH NUMBER OF SAMPLES FOR THE PAVIAU DATA SET

Class	Land Cover Type	No. of Samples
$C1$	Asphalt	6631
$C2$	Meadow	18649
$C3$	Gravel	2099
$C4$	Trees	3064
$C5$	Metal sheets	1345
$C6$	Bare Soil	5029
$C7$	Bitumen	1330
$C8$	Bricks	3682
$C9$	Shadows	947
	Total	42776

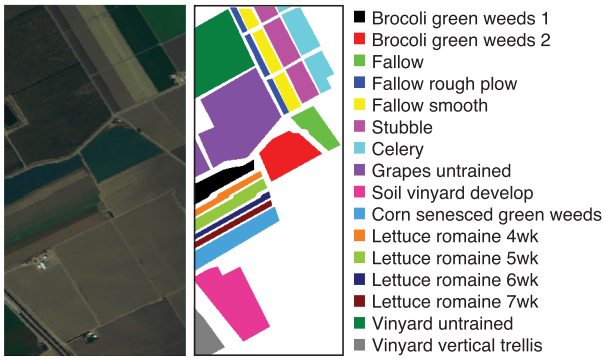


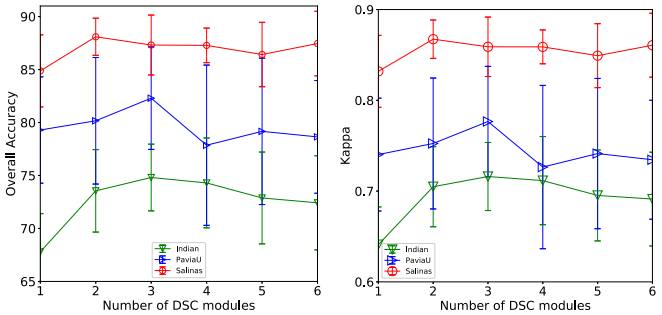
Fig. 11. (Left) False-color image and (Right) ground-truth map of the Salinas data set.

TABLE III
LAND COVER CLASSES WITH NUMBER OF SAMPLES FOR THE SALINAS DATA SET

Class	Land Cover Type	No. of Samples
$C1$	Brocoli-green-weeds-1	2009
$C2$	Brocoli-green-weeds-2	3726
$C3$	Fallow	1976
$C4$	Fallow-rough-plow	1394
$C5$	Fallow-smooth	2678
$C6$	Stubble	3959
$C7$	Celery	3579
$C8$	Grapes-untrained	11,271
$C9$	Soil-vineyard-develop	6203
$C10$	Corn-senesced-green-weeds	3278
$C11$	Lettuce-romaine-4wk	1068
$C12$	Lettuce-romaine-5wk	1927
$C13$	Lettuce-romaine-6wk	916
$C14$	Lettuce-romaine-7wk	1070
$C15$	Vineyard-untrained	7268
$C16$	Vineyard-vertical-trellis	1807
	Total	54,129

contained in the model, including δ in the DSC module, ρ in the BCF module, and Ψ in the global average pooling classifier, can be learned in the training procedure (all the parameters are optimized by RMSprop algorithm [62]; learning rate is set to 0.001, while the coefficient of weight decay is 0.003.), there are some hyperparameter (such as the number of DSC modules, the batch size S, and the filter size N in the DSC module) that should be carefully concerned in advance. It is worth pointing out that the number of reserved features K in the SSSE operation is automatically decided in [59]. Here, the batch size S is set as 64 for all experiments by experience.

It can be easily observed from Fig. 1 that the feature extractor in the LWCNN framework is constructed by stacking



(a) (b)

Fig. 12. Proposed LWCNN framework with different number of DSC modules on the three hyperspectral data sets. (a) OA. (b) Kappa coefficient.

several DSC modules; hence, the amount of DSC modules will have a crucial impact on the representative capacity of the proposed method. Furthermore, increasing the number of DSC modules can generally improve the performance of the proposed model, but more DSC modules in the model will be prone to overfitting in the case of a small sample set, and the computational load is also increased. Fig. 12 evaluates the proposed LWCNN framework with a different number of DSC modules on the three hyperspectral data sets. Here, five labeled samples per class are randomly picked out from the labeled set to build up a small training set, and the remaining ones are used for testing. All the experiments are repeated for ten times, while both the mean and standard variation are reported. Besides, overall accuracy (OA) and kappa coefficient [63] are adopted to quantify the performance.

It can be found from Fig. 12 that the LWCNN model with three DSC modules has achieved the best performance with the Indian Pines and Pavia University data sets. For the Salinas data set, the best choice for the number of DSC modules is 2, which is slightly better than that of 3. For the sake of consistency and the generalization of the proposed LWCNN model, the number of DSC modules is set as 3, as shown in Fig. 1. Moreover, the filter size N in each DSC module is kept the same for simplicity and set as 10.

Second, the power of three important steps in our work, including the DSC module, BCF module, and layer normalization, is extensively studied. Accordingly, three methods are carefully formulated and presented in comparison with the proposed LWCNN approach. More precisely, in order to analyze the role of the DSC module, the SSSE-GAPC is constructed by removing both the DSC and BCF modules from the LWCNN model, and the pixel vector of SSSE feature directly enters the global average pooling classifier to produce the predicted probabilities. Meanwhile, with respect to the BCF module, the LWCNN-noBCF model is considered by removing the BCF module from the proposed framework, and the features obtained from DSC modules directly enter the global average pooling classifier without any dimension reduction processing. Besides, since the normalization procedure is crucial for the CNN-based system to prevent degeneration of the model, LWCNN-BN is taken into account, which replaces all the layer normalization of our proposed LWCNN method with batch normalization.

TABLE IV
CLASSIFICATION PERFORMANCE OF DIFFERENT METHODS ON THE THREE HYPERSPECTRAL DATA SETS

Data set	Algorithm	OA (%)	AA (%)	Kappa
Indian Pines	SSSE-GAPC	57.12	68.55	0.52
	LWCNN-noBCF	67.63	79.48	0.64
	LWCNN-BN	73.39	84.28	0.70
	LWCNN	**74.78**	**84.85**	**0.72**
PaviaU	SSSE-GAPC	62.72	72.23	0.54
	LWCNN-noBCF	73.46	83.82	0.68
	LWCNN-BN	80.76	87.90	0.76
	LWCNN	**82.30**	**87.27**	**0.78**
Salinas	SSSE-GAPC	74.10	81.97	0.71
	LWCNN-noBCF	82.12	87.91	0.80
	LWCNN-BN	87.82	93.35	0.86
	LWCNN	**88.61**	**93.77**	**0.87**

Table IV lists the detailed classification performance of four compared methods on the three hyperspectral data sets. Except for the two metrics, both OA and kappa coefficient, the average accuracy (AA) is also taken into consideration. Here, the experimental setting is the same as earlier, i.e., five labeled samples per class are taken as the training set, and the experiment is repeated ten times. It can be found from Table IV that the accuracy of the LWCNN-noBCF model is considerably higher than SSSE-GAPC for all the three hyperspectral data sets, validating the significance of the deep representative features obtained by the powerful DSC modules. Furthermore, the performance of our LWCNN framework is substantially improved compared with the LWCNN-noBCF, indicating the importance of the potential contextual information extracted by the novel BCF module. Besides, layer normalization in the proposed LWCNN model can bring a continuous improvement over the batch normalization in LWCNN-BN, showing the rationality of the incorporated layer normalization scheme.

C. Classification Performance

To illustrate the advantage of the proposed LWCNN framework for hyperspectral image classification, two state-of-the-art classifiers and three DL-learning model proposed recently are taken into comparison, which is described as follows.

1) *Generalized Composite Kernel (GCK):* Li *et al.* [64] constructed a framework of nonparameter GCKs to extract the spatial–spectral information from hyperspectral image. Logistic regression is used as the classifier, and the spatial feature is obtained from extended multiattribute profiles.
2) *Morphological-Based K-Nearest Neighborhood (MOR-KNN):* Morphological features are constructed for hyperspectral image based on the operation of openings and closings with a structuring element of increasing size, and the KNN classifier is engaged for classification.
3) *2DCNN:* The typical convolution neural network, i.e., LeNet [65], was extended for hyperspectral image classification [39].
4) *3DCNN:* To the best of our acknowledge, [38] is the first work that introduced the 3DCNN for hyperspectral image classification. There are two 3-D convolution layers in this network without a pooling layer to exploit the joint spatial–spectral information. The classification was performed by a fully connected layer and softmax function.
5) *Spectral–Spatial Long Short-Term Memory (SaSeLSTM):* SaSeLSTM [66] adopted the LSTM to develop a feature extractor for hyperspectral image. For the spectrum, the values of the spectral pixel vector enter into the spectral LSTM one by one. Alternatively, PCA was applied. The first component was separated into several vectors and entered into spatial LSTM. Each LSTM branch will produce a score and decision procedure fuses both score vectors and provide a classification probability.
6) *LWCNN-RAW:* The raw spectral data are directly used without applying the SSSE procedure.
7) *LWCNN-PCA:* PCA is used to replace the SSSE procedure to reduce the spectral dimension.

It is worth pointing out that the compared methods used in the experiments follow their original paper. Especially, 2DCNN and LWCNN-PCA use PCA to reduce the dimension, while both the raw data and PCA-reduced features are adopted for SaSeLSTM. The MOR-KNN utilizes the morphological feature. For the rest compared methods, including GCK, 3DCNN, and LWCNN-RAW, the original raw data are taken as the input.

TABLE V
PARAMETER SIZE OF FOUR DL-BASED METHODS ON THREE HYPERSPECTRAL DATA SETS

Network	Indian Pines	PaviaU	Salinas
2DCNN	62216 243.03KB	49489 193.31KB	62216 243.03KB
3DCNN	128248 500.97KB	127753 499.04KB	128328 501.28KB
SaSeLSTM	399904 1.53MB	397202 1.52MB	399904 1.53MB
LWCNN	13849 54.10KB	13800 53.90KB	13849 54.10KB

Before presenting the detailed experimental results, the parameter size of four DL-based methods (including 2DCNN, 3DCNN, SaSeLSTM, and our LWCNN) on three hyperspectral data sets is summarized in Table V. The first row of each method is the number of parameters, while the second row is the memory space consumed by the parameters. It can be clearly seen from Table V that the parameter size of the proposed LWCNN framework is significantly smaller than the other three methods, ensuring the efficiency and feasibility of our method (this is the main reason that the proposed method is named LWCNN).

With respect to the training set, a fixed number of samples are randomly selected from each category, and the remaining labeled samples make up the testing set. For the Indian Pines hyperspectral image data set, the number ranges from 3 to 15 per class (since the 12th class only contains 20 labeled samples). For the other two data sets, Pavia University and Salinas, the number ranges from 3 to 20 per category. Similarly, due to the weight initialization of the CNN network and randomness influence of data sampling, each circumstance of the experiment is repeated ten times, and both the mean value

TABLE VI

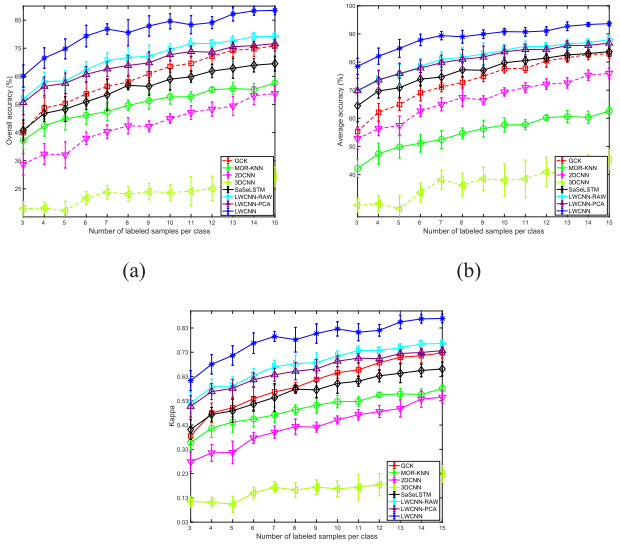
CLASSIFICATION ACCURACY (%) AND KAPPA MEASURE OBTAINED FROM THE INDIAN PINES DATA SET ON THE TEST SET WITH FIVE LABELING SAMPLES PER CLASS AS TRAINING SET

Class	GCK	MOR-KNN	2DCNN	3DCNN	SaSeLSTM	LWCNN-RAW	LWCNN-PCA	LWCNN
$C1$	82.44	94.44	**100.00**	52.44	97.56	95.29	95.52	99.51
$C2$	72.60	**77.79**	29.44	12.66	44.07	51.47	48.68	62.74
$C3$	52.34	41.87	44.12	11.99	43.71	51.36	50.57	**66.64**
$C4$	48.07	40.63	78.10	51.29	75.26	76.15	74.56	**91.77**
$C5$	77.55	81.84	50.90	15.69	75.21	72.95	74.00	**86.57**
$C6$	36.19	36.91	31.30	10.97	52.23	85.55	86.94	**98.14**
$C7$	67.09	40.14	**100.00**	66.52	**100.00**	97.87	99.13	99.13
$C8$	66.60	72.72	42.20	18.39	**92.88**	80.59	82.04	88.29
$C9$	56.29	67.12	88.67	51.33	89.33	95.31	**100.00**	99.33
$C10$	**90.95**	90.95	32.89	21.14	58.99	60.07	57.96	72.96
$C11$	42.31	64.37	13.88	13.98	29.18	49.01	46.84	**64.87**
$C12$	81.33	**100.00**	41.87	14.29	53.81	55.58	54.50	60.65
$C13$	40.32	38.15	71.70	28.30	86.05	90.93	95.83	**99.55**
$C14$	49.33	42.39	58.17	19.61	68.29	82.57	83.43	**85.27**
$C15$	81.12	42.22	53.18	28.16	77.51	72.16	71.03	**82.57**
$C16$	93.67	93.72	82.95	29.89	89.77	91.56	94.10	**99.66**
OA	55.37	49.87	37.05	17.23	53.35	63.38	62.51	**74.78**
AA	64.89	64.08	57.46	27.92	70.87	75.53	75.95	**84.85**
k	0.50	0.44	0.32	0.10	0.49	0.59	0.58	**0.72**

TABLE VII

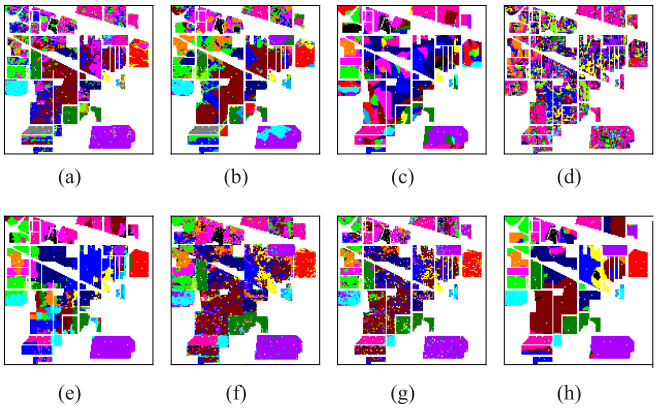
CLASSIFICATION ACCURACY (%) AND KAPPA MEASURE OBTAINED FROM THE PAVIA UNIVERSITY DATA SET ON THE TEST SET WITH FIVE LABELING SAMPLES PER CLASS AS TRAINING SET

Class	GCK	MOR-KNN	2DCNN	3DCNN	SaSeLSTM	LWCNN-RAW	LWCNN-PCA	LWCNN
$C1$	74.64	51.85	31.99	59.47	35.27	76.68	78.06	**90.22**
$C2$	64.73	50.97	33.61	61.93	48.90	50.92	56.63	**74.20**
$C3$	71.82	55.37	36.68	57.44	28.88	70.66	67.93	**75.64**
$C4$	75.56	92.50	40.02	76.04	42.39	**93.97**	87.78	87.64
$C5$	75.20	96.26	49.24	86.82	55.82	**99.75**	99.39	94.79
$C6$	53.54	66.11	23.82	37.09	40.51	75.28	67.78	**89.37**
$C7$	91.48	78.82	53.71	79.73	40.14	86.72	80.39	**99.88**
$C8$	75.94	80.07	55.21	55.71	21.43	77.69	73.20	**86.67**
$C9$	81.46	80.28	66.77	88.16	44.52	**99.89**	99.54	87.04
OA	68.57	61.51	36.52	60.79	41.84	67.86	68.50	**82.30**
AA	73.82	72.47	43.45	66.93	39.76	81.29	78.97	**87.27**
k	0.60	0.53	0.25	0.51	0.29	0.61	0.61	**0.78**



(a) (b) (c)

Fig. 13. Indian Pines hyperspectral image: classification performance versus number of labeled samples per class. (a) OA. (b) AA. (c) Kappa.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 14. Indian Pines data set. Classification maps obtained by (a) GCK (53.84%), (b) MOR-KNN (50.90%), (c) 2DCNN (41.37%), (d) 3DCNN (14.52%), (e) SaSeLSTM (53.58%), (f) LWCNN-RAW (63.00%), (g) LWCNN-PCA (62.78%), and (h) LWCNN (78.07%).

and standard deviation of OA, AA, and kappa coefficient are reported.

Fig. 13 shows the classification accuracy of various compared methods with a different number of labeled samples per class (3, . . . , 15) on the Indian Pines hyperspectral image. Here, three metrics, including OA, AA, and Kappa coefficient, are utilized. It can be seen from Fig. 13 that the trend of the change in curves is basically the same, while the increase in the number of labeled samples has a positive impact on the classification performance. Especially, since the spatial resolution of the Indian Pines hyperspectral image is particularly low (i.e., 20 m per pixel), the 3DCNN method

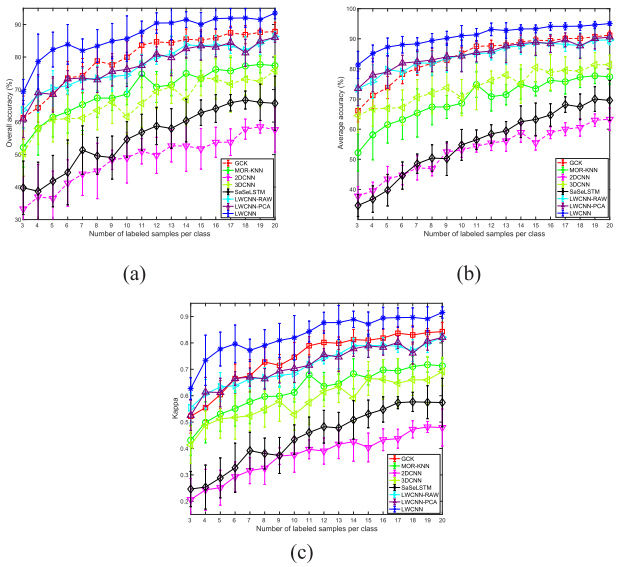


Fig. 15. Pavia University hyperspectral image: classification performance versus number of labeled samples per class. (a) OA. (b) AA. (c) Kappa.

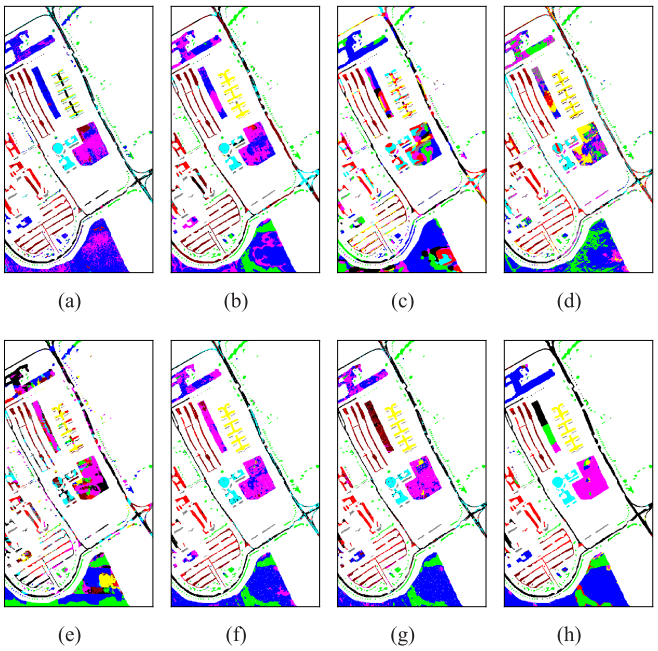


Fig. 16. Pavia University data set. Classification maps obtained by (a) GCK (71.80%), (b) MOR-KNN (59.77%), (c) 2DCNN (47.11%), (d) 3DCNN (47.03%), (e) SaSeLSTM (40.30%), (f) LWCNN-RAW (72.08%), (g) LWCNN-PCA (73.36%), and (h) LWCNN (81.88%).

cannot extract informative features, resulting in a very poor result. Alternatively, our LWCNN framework always provides the best accuracy, validating the effectiveness of the proposed method. Furthermore, the performance of LWCNN is also better than LWCNN-RAW and LWCNN-PCA. This is reasonable since the SSSE procedure not only reduces the spectral dimension but also incorporates the spatial structural information, demonstrating the necessity of the incorporated SSSE procedure. When only five labeled samples per class are used to build the training set, detailed results (including the accuracy of each class and the three metrics) of the eight compared methods are listed in Table VI. Similarly, most of the accuracies achieved by LWCNN are higher than the others. Besides, the classification maps of the compared methods in a single experiment are also illustrated in Fig. 14. It can be visually seen that the map obtained by our LWCNN [see Fig. 14(h)] is more consistent with the ground-truth map (see Fig. 9) than the others.

In the following, the Pavia University hyperspectral image data set is considered. Fig. 15 provides the classification performance of the compared methods with a different number of labeled samples $(3, \ldots, 20)$. Different from the abovementioned Indian Pines data set, the performance of 3DCNN is greatly improved since the spatial resolution of the Pavia University data set is much higher (1.3 m per pixel). Meanwhile, because the small sample set problem is mainly concerned in our work, the three CNN-based methods, including 2DCNN, 3DCNN, and SaSeLSTM, cannot be well trained, and the performance of them is even lower than the traditional GCK method. Alternatively, our LWCNN approach gives the best results all the time. Similarly, Table VII presents a thorough description of the classification performance of eight compared methods with five labeled samples per class as a training set, and our LWCNN approach provides the best accuracies in most cases. Moreover, Fig. 16 exhibits the classification maps of eight methods, and the proposed LWCNN method is more advantageous than the compared ones.

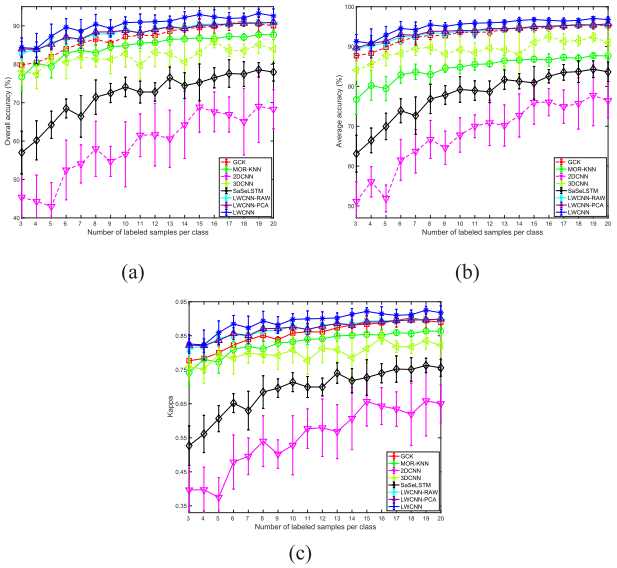


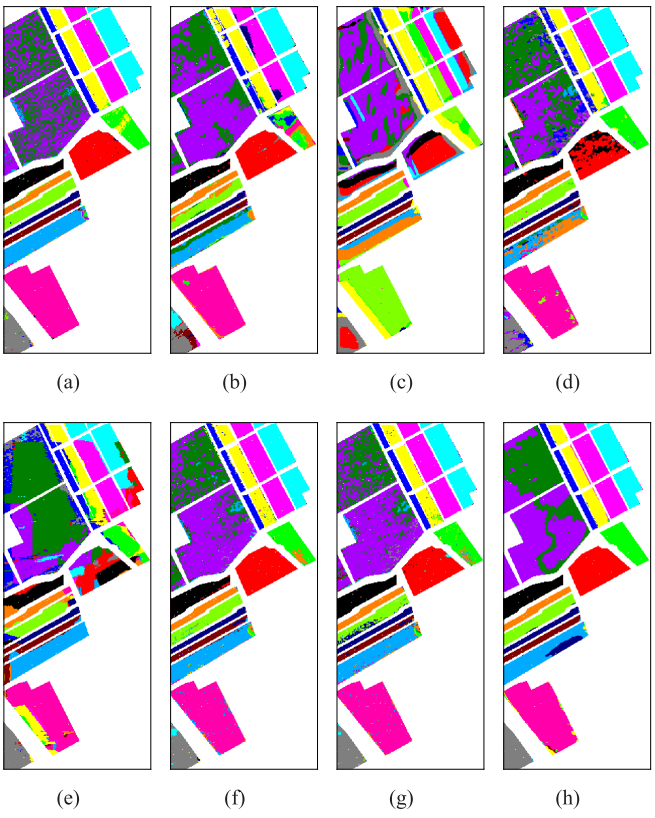
Fig. 17. Salinas hyperspectral image: classification performance versus number of labeled samples per class. (a) OA. (b) AA. (c) Kappa.

Finally, the Salinas hyperspectral image is investigated. Fig. 17 shows the OA, AA, and kappa values of the compared eight methods with a different number of training samples $(3, \ldots, 20)$. Since the spatial distribution of the data is very regular (as shown in Fig. 11), the performance of all methods is better than the abovementioned two data sets. Likewise, our LWCNN always achieves the highest accuracy. Table VIII and Fig. 18, respectively, summarizes the classification accuracy of eight methods when only five labeled samples per class are used for training, and our LWCNN model exhibits the best results in most cases, indicating the superiority of the proposed LWCNN approach.

TABLE VIII

CLASSIFICATION ACCURACY (%) AND KAPPA MEASURE OBTAINED FROM THE SALINAS DATA SET ON THE TEST SET WITH FIVE LABELING SAMPLES PER CLASS AS TRAINING SET

Class	GCK	MOR-KNN	2DCNN	3DCNN	SaSeLSTM	LWCNN-RAW	LWCNN-PCA	LWCNN
$C1$	90.31	99.21	37.54	91.69	70.28	96.62	98.00	**99.61**
$C2$	95.79	98.00	17.89	95.61	34.88	**99.59**	99.42	97.40
$C3$	**91.55**	44.63	15.01	85.88	64.69	85.90	85.41	90.50
$C4$	96.30	75.00	83.42	92.20	78.91	98.78	**99.01**	98.40
$C5$	**96.34**	84.74	71.15	94.57	75.86	96.04	96.31	94.09
$C6$	95.27	93.19	49.76	98.40	79.72	98.36	**99.19**	98.41
$C7$	99.17	97.96	29.09	98.92	58.58	98.96	98.88	**99.86**
$C8$	49.03	62.45	29.46	59.62	40.15	68.80	62.12	**78.03**
$C9$	**99.53**	92.21	56.34	95.35	86.28	98.09	98.07	94.47
$C10$	87.00	69.38	39.47	67.15	80.70	80.87	84.78	**91.29**
$C11$	93.59	93.57	76.53	91.15	73.03	96.44	97.39	**99.98**
$C12$	98.39	83.98	63.06	**99.24**	73.74	96.27	94.94	95.46
$C13$	**98.85**	97.35	92.40	96.44	78.87	98.68	98.60	98.43
$C14$	86.57	95.92	84.43	93.39	79.81	89.06	88.91	**98.13**
$C15$	68.97	**72.48**	37.57	64.01	64.08	66.78	70.58	67.32
$C16$	90.27	63.06	46.84	80.64	79.73	85.88	90.04	**98.91**
OA	81.94	79.46	42.97	81.40	64.23	85.66	85.21	**88.61**
AA	89.81	82.70	51.87	87.77	69.96	90.95	91.35	**93.77**
k	0.80	0.77	0.38	0.79	0.61	0.84	0.84	**0.87**



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 18. Salinas data set. Classification maps obtained by (a) GCK (82.84%), (b) MOR-KNN (79.72%), (c) 2DCNN (46.09%), (d) 3DCNN (77.34%), (e) SaSeLSTM (64.21%), (f) LWCNN-RAW (87.74%), (g) LWCNN-PCA (86.98%), and (h) LWCNN (89.91%).

V. CONCLUSION

In this article, we aim to construct an LWCNN to tackle the small sample set problem of hyperspectral image classification. By incorporating the SSSE operation, the parameter size has been substantially reduced. Meanwhile, multiple DSC and BCF modules have been carefully designed and connected, and the discriminative ability of the extracted features can be ensured. Besides, the batch normalization is replaced by the layer normalization scheme, and a global average pooling classifier is imported; hence, the classification performance can be further improved.

In the experiments, the power of three crucial steps in the proposed LWCNN framework, including the DSC module, BCF module, and layer normalization, is validated. Moreover, a number of state-of-the-art methods, including GCK, MOR-KNN, 2DCNN, 3DCNN, SaSeLSTM, LWCNN-RAW, and LWCNN-PCA, are taken into consideration, and the results consistently demonstrate the advantage and robustness of the proposed LWCNN approach, especially when the training set is small.

REFERENCES

[1] D. Haboudane, "Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture," *Remote Sens. Environ.*, vol. 90, no. 3, pp. 337–352, Apr. 2004.

[2] S. Delalieux, P. J. Zarco-Tejada, L. Tits, M. A. J. Bello, D. S. Intrigliolo, and B. Somers, "Unmixing-based fusion of hyperspatial and hyperspectral airborne imagery for early detection of vegetation stress," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2571–2582, Jun. 2014.

[3] L. Liang *et al.*, "Estimation of crop LAI using hyperspectral vegetation indices and a hybrid inversion method," *Remote Sens. Environ.*, vol. 165, pp. 123–134, Aug. 2015.

[4] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lect.*, vol. 1, p. 32, Aug. 2000.

[5] S. Jia, G. Tang, J. Zhu, and Q. Li, "A novel ranking-based clustering approach for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 88–102, Jan. 2016.

[6] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.

[7] S. Prasad and L. M. Bruce, "Limitations of principal components analysis for hyperspectral target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 625–629, Oct. 2008.

[8] X. Liu, X. Feng, F. Liu, and Y. He, "Identification of hybrid rice strain based on near-infrared hyperspectral imaging technology," *Trans. Chin. Soc. Agricult. Eng.*, vol. 33, no. 22, pp. 189–194, 2017.

[9] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[10] K. Murinto and N. R. D. P. Astuti, "Discriminant independent component analysis for hyperspectral image classification," *IOP Conf. Mater. Sci. Eng.*, vol. 403, Oct. 2018, Art. no. 012059.

[11] J. Li, Y. Zhang, M. Liu, J. Chen, and L. Xue, "Rapid detection and visualization of mechanical bruises on nanfeng mandarin using the hyperspectral imaging combined with ICA_LSQ method," *Food Anal Methods*, vol. 12, no. 26, pp. 1–10, 2019.

[12] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[13] M. Huang, Q. Zhu, B. Wang, and R. Lu, "Analysis of hyperspectral scattering images using locally linear embedding algorithm for apple mealiness classification," *Comput. Electron. Agricult.*, vol. 89, pp. 175–181, Nov. 2012.

[14] M. Pesaresi, A. Gerhardinger, and F. Kayitakire, "A robust built-up area presence index by anisotropic rotation-invariant textural measure," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 1, no. 3, pp. 180–192, Sep. 2008.

[15] F. Ghedass and I. R. Farah, "An improved classification of hyperspectral imaging based on spectral signature and gray level co-occurrence matrix," in *Proc. SAGEO*, 2015, pp. 269–282.

[16] P. Ghamisi, R. Souza, J. A. Benediktsson, X. X. Zhu, L. Rittner, and R. A. Lotufo, "Extinction profiles for the classification of remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5631–5645, Oct. 2016.

[17] S. Jia, Y. Xie, G. Tang, and J. Zhu, "Spatial-spectral-combined sparse representation-based classification for hyperspectral imagery," *Soft Comput.*, vol. 20, no. 12, pp. 4659–4668, Dec. 2016.

[18] L. Fang, C. Wang, S. Li, and J. A. Benediktsson, "Hyperspectral image classification via multiple-feature-based adaptive sparse representation," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1646–1657, Jul. 2017.

[19] S. Jia, B. Deng, J. Zhu, X. Jia, and Q. Li, "Local binary pattern-based hyperspectral image classification with superpixel guidance," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 749–759, Feb. 2018.

[20] S. Jia, X. Deng, J. Zhu, M. Xu, J. Zhou, and X. Jia, "Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7770–7784, Oct. 2019.

[21] Y. Gu, T. Liu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3235–3247, Jun. 2016.

[22] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "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.

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

[24] S. Wang, "Texture feature extraction of hyper-spectral image with three-dimensional gray-level co-occurrence," *J. Inf. Comput. Sci.*, vol. 12, no. 4, pp. 1439–1448, Mar. 2015.

[25] 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.

[26] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.

[27] X. Zhou, S. Prasad, and M. M. Crawford, "Wavelet-domain multiview active learning for spatial-spectral hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4047–4059, Sep. 2016.

[28] S. Jia, L. Shen, J. Zhu, and Q. Li, "A 3-D Gabor phase-based coding and matching framework for hyperspectral imagery classification," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1176–1188, Apr. 2018.

[29] J. Zhu, J. Hu, S. Jia, X. Jia, and Q. Li, "Multiple 3-D feature fusion framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1873–1886, Apr. 2018.

[30] S. Jia, Z. Lin, B. Deng, J. Zhu, and Q. Li, "Cascade superpixel regularized Gabor feature fusion for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1638–1652, May 2019.

[31] T. Pham *et al.*, "Airborne object detection using hyperspectral imaging: Deep learning review," in *Proc. ICCSA*. Springer, 2019, pp. 306–321.

[32] H. Petersson, D. Gustafsson, and D. Bergstrom, "Hyperspectral image analysis using deep learning—A review," in *Proc. 6th IPTA*, 2017, pp. 1–6.

[33] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[34] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, Feb. 2017.

[35] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[36] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[37] M. Zhang, M. Gong, Y. Mao, J. Li, and Y. Wu, "Unsupervised feature extraction in hyperspectral images based on wasserstein generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2669–2688, May 2019.

[38] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017.

[39] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jul. 2015, Art. no. 258619.

[40] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

[41] W. Xue, T. Kun, and C. Yu, "Capsnet and triple-gans towards hyperspectral classification," in *Proc. 5th Int. Earth Observ. Remote Sens. Appl.*, 2018, pp. 1–4.

[42] H. Zhang *et al.*, "1D-convolutional capsule network for hyperspectral image classification," 2019, *arXiv:1903.09834*. [Online]. Available: http://arxiv.org/abs/1903.09834

[43] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014, *arXiv:1402.1128*. [Online]. Available: https://arxiv.org/abs/1402.1128

[44] Z. Tan, J. Su, B. Wang, Y. Chen, and X. Shi, "Lattice-to-sequence attentional neural machine translation models," *Neurocomputing*, vol. 284, pp. 138–147, Apr. 2018.

[45] N. Dickson, K. Sahari, Y. Hu, and L. Kiong, "Multiple sequence behavior recognition on humanoid robot using long short-term memory (LSTM)," in *Proc. ROMA*, 2014, pp. 109–114.

[46] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[47] Y. Xu, L. Zhang, B. Du, and F. Zhang, "Spectral–spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.

[48] B. Pan, Z. Shi, and X. Xu, "MugNet: Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 108–119, Nov. 2018.

[49] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[50] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.

[51] M. Su, Y. Liu, L. Liu, Y. Peng, and T. Jiang, "Interleaved group convolution network for hyperspectral image classification," *Proc. SPIE*, vol. 11427, Jan. 2020, Art. no. 114270B.

[52] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[53] S. K. Roy, S. Chatterjee, S. Bhattacharyya, B. B. Chaudhuri, and J. Platos, "Lightweight spectral–spatial squeeze-and-excitation residual bag-of-features learning for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5277–5290, Aug. 2020.

[54] Y. Liu, L. Gao, C. Xiao, Y. Qu, K. Zheng, and A. Marinoni, "Hyperspectral image classification based on a shuffled group convolutional neural network with transfer learning," *Remote Sens.*, vol. 12, no. 11, p. 1780, Jun. 2020.

[55] W. Czaja and M. Ehler, "Schroedinger eigenmaps for the analysis of biomedical data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1274–1280, May 2013.
[56] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. NIPS*, 2014, pp. 3104–3112.
[57] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.
[58] W. Luo, "Face recognition based on Laplacian eigenmaps," in *Proc. CSSS*, 2011, pp. 416–419.
[59] N. D. Cahill and D. W. Messinger, "Schroedinger eigenmaps with nondiagonal potentials for spatial-spectral clustering of hyperspectral imagery," *Proc. SPIE*, vol. 9088, 2014.
[60] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: https://arxiv.org/abs/1312.4400
[61] G. Bouchard, "Efficient bounds for the softmax function, applications to inference in hybrid models," Citeseer, Meylan, France, Tech. Rep. 31, 2007.
[62] M. Emtiyaz Khan, Z. Liu, V. Tangkaratt, and Y. Gal, "Vprop: Variational inference using RMSprop," 2017, *arXiv:1712.01038*. [Online]. Available: http://arxiv.org/abs/1712.01038
[63] J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction*. Springer, 2013.
[64] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.
[65] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[66] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMs," *Neurocomputing*, vol. 328, pp. 39–47, Feb. 2019.

Sen Jia (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively.

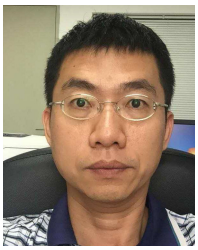
Since 2008, he has been with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, where he is a Full Professor. His research interests include hyperspectral image processing, signal and image processing, and machine learning.

Zhijie Lin received the B.E. degree from the Hefei University of Technology, Hefei, China, in 2017. He is pursuing the master's degree in computer science and technology with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

His research interests include hyperspectral and LiDAR classification, machine learning, and pattern recognition.

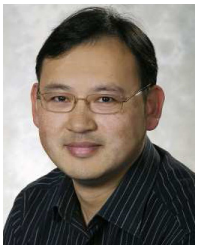
Meng Xu (Member, IEEE) received the B.S. and M.E. degrees in electrical engineering from the Ocean University of China, Qingdao, China, in 2011 and 2013, respectively, and the Ph.D. degree from the University of New South Wales, Canberra, ACT, Australia, in 2017.

She is an Associate Research Fellow with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her research interests include cloud removal and remote sensing image processing.



Qiang Huang received the B.S. degree in electrical engineering from Shenzhen University, Shenzhen, China, in 1999, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K., in 2004.

He was a Research Associate with the University of Leicester, Leicester, U.K., from 2003 to 2004. He was teaching as a Lecturer, an Associate Processor, and a Processor with Shenzhen University, Shenzhen, China. He has published more than 36 research articles. His research interests include distributed embedded real-time systems, digital image processing, supercomputer systems, and cloud computing.



Jun Zhou (Senior Member, IEEE) received the B.S. degree in computer science and the B.E. degree in international business from the Nanjing University of Science and Technology, Nanjing, China, in 1996 and 1998, respectively, the M.S. degree in computer science from Concordia University, Montreal, QC, Canada, in 2002, and the Ph.D. degree in computing science from the University of Alberta, Edmonton, AB, Canada, in 2006.

He was a Research Fellow with the Research School of Computer Science, The Australian National University, Canberra, ACT, Australia, and a Researcher with Canberra Research Laboratory, National Information and Communications Technology Australia, Canberra. In 2012, he joined the School of Information and Communication Technology, Griffith University, Nathan, QLD, Australia, where he is an Associate Professor. His research interests include pattern recognition, computer vision, and spectral imaging and their applications in remote sensing and environmental informatics.

Xiuping Jia (Senior Member, IEEE) received the B.Eng. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 1982, and the Ph.D. degree in electrical engineering from The University of New South Wales, Sydney, NSW, Australia, in 1996.

Since 1988, she has been with the School of Information Technology and Electrical Engineering, The University of New South Wales, Canberra Campus, Canberra, ACT, Australia, where she is a Senior Lecturer. She is also a Guest Professor with Harbin Engineering University, Harbin, China, and an Adjunct Researcher with the China National Engineering Research Center for Information Technology in Agriculture. She has coauthored a textbook on remote sensing, titled *Remote Sensing Digital Image Analysis* [Springer-Verlag, third (1999) and fourth editions (2006)]. Her research interests include remote sensing, image processing, and spatial data analysis.

Dr. Jia is a Subject Editor of the *Journal of Soils and Sediments* and an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Qingquan Li received the M.S. degree in engineering and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1988 and 1998, respectively.

From 1988 to 1996, he was an Assistant Professor with Wuhan University, where he was an Associate Professor from 1996 to 1998, and has been a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing since 1998. He is the President of Shenzhen University, Shenzhen, China, where he is also the Director of the Shenzhen Key Laboratory of Spatial Information Smart Sensing and Services. His research interests include photogrammetry, remote sensing, and intelligent transportation systems.

Dr. Li is an expert in modern traffic with the National 863 Plan and an Editorial Board Member of the *Surveying and Mapping Journal* and the *Wuhan University Journal—Information Science Edition*.