Multiscale Dynamic Graph Convolutional Network for Hyperspectral Image Classification

Sheng Wan, Chen Gong, *Member, IEEE*, Ping Zhong, *Senior Member, IEEE*, Bo Du, *Senior Member, IEEE*, Lefei Zhang, *Member, IEEE*, and Jian Yang, *Member, IEEE*

***Abstract*— Convolutional neural network (CNN) has demonstrated impressive ability to represent hyperspectral images and to achieve promising results in hyperspectral image classification. However, traditional CNN models can only operate convolution on regular square image regions with fixed size and weights, and thus, they cannot universally adapt to the distinct local regions with various object distributions and geometric appearances. Therefore, their classification performances are still to be improved, especially in class boundaries. To alleviate this shortcoming, we consider employing the recently proposed graph convolutional network (GCN) for hyperspectral image classification, as it can conduct the convolution on arbitrarily structured non-Euclidean data and is applicable to the irregular image regions represented by graph topological information. Different from the commonly used GCN models that work on a fixed graph, we enable the graph to be dynamically updated along with the graph convolution process so that these two steps can be benefited from each other to gradually produce the discriminative embedded features as well as a refined graph. Moreover, to comprehensively deploy the multiscale information inherited by hyperspectral images, we establish multiple input graphs with different neighborhood scales to extensively exploit the diversified spectral–spatial correlations at multiple scales. Therefore, our method is termed multiscale dynamic GCN (MDGCN). The experimental results on three typical benchmark data sets firmly demonstrate the superiority of the proposed MDGCN to other state-of-the-art methods in both qualitative and quantitative aspects.**

***Index Terms*— Dynamic graph, graph convolutional network (GCN), hyperspectral image classification, multiscale information.**

Manuscript received May 14, 2019; revised July 26, 2019, September 19, 2019, and September 24, 2019; accepted October 8, 2019. Date of publication November 20, 2019; date of current version April 22, 2020. This work was supported in part by the National Science Foundation (NSF) of China under Grant 61602246, Grant 61973162, Grant U1713208, Grant 61971428, and Grant 61671456, in part by the NSF of Jiangsu Province under Grant BK20171430, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319, in part by the Open Project of State Key Laboratory of Integrated Services Networks (Xidian University, ID: ISN19-03), in part by the Summit of the Six Top Talents Program under Grant DZXX-027, in part by the Young Elite Scientists Sponsorship Program by Jiangsu Province, in part by the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology (CAST) under Grant 2018QNRC001, in part by the Guangdong Key Area Research Project under Grant 2018B010108003, and in part by the Program for Changjiang Scholars. *(Corresponding author: Chen Gong.)*

S. Wan and C. Gong are with the PCA Lab, Nanjing University of Science and Technology, Nanjing 210094, China, with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: wansheng315@hotmail.com; chen.gong@njust.edu.cn).

P. Zhong is with the National Key Laboratory of Science and Technology on ATR, National University of Defense Technology, Changsha 410073, China (e-mail: zhongping@nudt.edu.cn).

B. Du and L. Zhang are with the School of Computer Science, Wuhan University, Wuhan 430079, China (e-mail: gunspace@163.com; zhanglefei@whu.edu.cn).

J. Yang is with the PCA Lab, Nanjing University of Science and Technology, Nanjing 210094, China, with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China, with the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: csjyang@njust.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.2019.2949180

I. Introduction

THE rapid development of optics and photonics has significantly advanced hyperspectral techniques. As a result, hyperspectral images, which consist of hundreds of contiguous bands and contain large amounts of useful information, can be easily acquired [1], [2]. Over the past few decades, hyperspectral image classification has played an important role in various fields, such as military target detection, vegetation monitoring, and disaster prevention and control.

Up to now, diverse kinds of approaches have been proposed for classifying the pixels of a hyperspectral image into certain land-cover categories. The early-staged methods are mainly based on conventional pattern recognition methods, such as nearest neighbor classifier and linear classifier. Among these conventional methods, K-nearest neighbor [3] has been widely used due to its simplicity in both theory and practice. Support vector machine (SVM) [4] also performs robustly and satisfactorily with high-dimensional hyperspectral data. In addition to these, graph-based methods [5], extreme learning machine [6], sparse representation-based classifier [7], and many other methods have been further employed to promote the performance of hyperspectral image classification. Nevertheless, it is difficult to distinguish different land-cover categories accurately by only using the spectral information [8]. With the observation that spatially neighboring pixels usually carry correlated information within a smooth spatial domain, many researchers have resorted to spectral–spatial classification methods and several models have been proposed to exploit such local continuity [9], [10]. For example, Markov random field (MRF)-based models [11] have been widely used for deploying spatial information and have achieved great popularity. In MRF-based models, spatial information

is usually regarded as *a priori* before optimizing an energy function via posteriori maximization. Meanwhile, morphological profile-based methods [12], [13] have also been proposed to effectively combine spatial and spectral information.

However, the aforementioned methods are all based on the handcrafted spectral–spatial features [14], which heavily depend on professional expertise and are quite empirical. To address this defect, deep learning [15]–[19] has been extensively employed for hyperspectral image classification and has attracted increasing attention for its strong representation ability. The main reason is that deep learning methods can automatically obtain abstract high-level representations by gradually aggregating the low-level features, by which the complicated feature engineering can be avoided [20]–[22]. The first attempt to use deep learning methods for hyperspectral image classification was made by Chen *et al.* [23], where the stacked autoencoder was built for high-level feature extraction. Subsequently, Mou *et al.* [18] first employed recurrent neural network (RNN) for hyperspectral image classification. Besides, Ma *et al.* [24] attempted to learn the spectral–spatial features via a deep learning architecture by fine-tuning the network via a supervised strategy. Recently, the convolutional neural network (CNN) has emerged as a powerful tool for hyperspectral image classification [25]–[27]. For instance, Jia *et al.* [28] employed CNN to extract spectral features and achieved superior performance to SVM. In addition, Hu *et al.* [29] proposed a five-layer 1-D CNN to classify hyperspectral images directly in the spectral domain. In these methods, the convolution operation is mainly applied to spectral domain, while the spatial details are largely neglected. Another set of deep learning approaches performs hyperspectral image classification by incorporating spectral–spatial information. For example, Makantasis *et al.* [30] encoded spectral–spatial information with a CNN and conducted classification with a multilayer perceptron. Besides, Zhang *et al.* [31] proposed a multidimensional CNN to automatically extract hierarchical spectral features and spatial features. Furthermore, Lee and Kwon [32] designed a novel contextual deep CNN, which is able to optimally explore contextual interactions by exploiting local spectral–spatial relationship among spatially neighboring pixels. Specifically, the joint exploitation of spectral–spatial information is obtained by a multiscale convolutional filter bank. Although the existing CNN-based methods have achieved good performance to some extent, they still suffer from some drawbacks. To be specific, conventional CNN models only conduct the convolution on the regular square regions, so they cannot adaptively capture the geometric variations of different object regions in a hyperspectral image. Besides, the weights of each convolution kernel are identical when convolving all image patches. As a result, the information of class boundaries may be lost during the feature abstraction process and misclassifications will probably happen due to the inflexible convolution kernel. In other words, the convolution kernels with fixed shape, size, and weights are not adaptive to all the regions in a hyperspectral image. Apart from that, CNN-based methods often take a long training time because of the large number of parameters.

Consequently, in this article, we propose to utilize the recently proposed graph convolutional network (GCN) [33], [34] for hyperspectral image classification. GCN operates on a graph and is able to aggregate and transform feature information from the neighbors of every graph node. Consequently, the convolution operation of GCN is adaptively governed by the neighborhood structure of a graph, and thus, GCN can be applicable to the non-Euclidean irregular data based on the predefined graph. Besides, both node features and local graph structure can be encoded by the learned hidden layers, so GCN is able to exhaustively exploit the image features and flexibly preserve the class boundaries.

Nevertheless, the direct use of traditional GCN for hyperspectral image classification is still inadequate. Since hyperspectral data are often contaminated by noise, the initial input graph may not be accurate. Specifically, the edge weight of pair-wise pixels may not represent their intrinsic similarities, which makes the input graph less than optimal. Furthermore, traditional GCN can only use the spectral features of image pixels without incorporating the spatial context, which is actually of great significance in hyperspectral images. Additionally, the computational complexity of traditional GCN will be unacceptable when the number of pixels gets too large. To tackle these difficulties in applying GCN to hyperspectral image classification, we propose a new type of GCN called multiscale dynamic GCN (MDGCN). Instead of utilizing a predefined fixed graph for convolution, we design a dynamic graph convolution operation, by which the similarity measures among pixels can be updated by fusing current feature embeddings. Consequently, the graph can be gradually refined during the convolution process of GCN, which will, in turn, make the feature embeddings more accurate. The processes of graph updating and feature embedding alternate, which work collaboratively to yield faithful graph structure and promising classification results. To take the multiscale cues into consideration, we construct multiple graphs with different neighborhood scales so that the spatial information at different scales can be fully exploited [35]. Different from commonly used GCN models which utilize only one fixed graph, the multiscale design enables MDGCN to extract spectral–spatial features with varied receptive fields, by which the comprehensive contextual information from different levels can be incorporated. Moreover, due to the large number of pixels brought by the high spatial resolution of hyperspectral images, the computational complexity of network training can be extremely high. To mitigate this problem, we group the raw pixels into a certain amount of homogeneous superpixels and treat each superpixel as a graph node. As a result, the number of nodes in each graph will be significantly reduced, which also helps to accelerate the subsequent convolution process.

Note that the graphs used in the proposed MDGCN are constructed based on spatial neighborhoods, and thus, the scarce initially labeled seed pixels, together with the massive unlabeled pixels for classification, have been involved in graph convolution and the learning process. In this sense, our proposed MDGCN falls into the scope of transductive semisupervised learning [36], where unlabeled data are accessible during

the training stage and the goal is to accurately classify these unlabeled data rather than building a generalizable classifier.

To sum up, the main contributions of the proposed MDGCN are as follows. First, we propose a novel dynamic graph convolution operation, which can reduce the impact of a bad predefined graph. Second, multiscale graph convolution is utilized to extensively exploit the spatial information and acquire better feature representation. Third, the superpixel technique is involved in our proposed MDGCN framework, which significantly reduces the complexity of model training. Finally, the experimental results on three typical hyperspectral image data sets show that MDGCN achieves the state-of-the-art performance when compared with the existing methods.

II. Related Works

In this section, we review some representative works on hyperspectral image classification and GCN, as they are related to this article.

A. Hyperspectral Image Classification

As a traditional yet important remote sensing technique, hyperspectral image classification has been intensively investigated and many related methods have been proposed, such as Bayesian methods [37], random forest [38], and kernel methods [39]. Particularly, SVM has shown impressive classification performance with limited labeled examples [40]. However, SVM independently treats every pixel (i.e., example) and fails to exploit the correlations among different image pixels. To address this limitation, spatial information is introduced. For instance, by directly incorporating spatial information into kernel design, Camps-Valls *et al.* [41] used SVM with composite kernels for hyperspectral image classification. Besides, filtering-based methods have also been applied to spectral–spatial classification. He and Chen [42] designed a 3-D filtering with a Gaussian kernel and its derivative for spectral–spatial information extraction. After that, they also proposed a discriminative low-rank Gabor filtering for spectral–spatial information extraction [43]. Additionally, MRF has been commonly used to exploit spatial context for hyperspectral image classification with the assumption that spatially neighboring pixels are more likely to take the same label [44]. However, when the neighboring pixels are highly correlated, the standard neighbor determination approaches will degrade the MRF models due to the insufficiently contained pixels [45]. Therefore, instead of modeling the joint distribution of spatially neighboring pixels, conditional random field directly models the class posterior probability given the hyperspectral image and has achieved encouraging performance [46], [47].

The aforementioned methods simply employ various manually extracted spectral–spatial features to represent the pixels, which highly depends on experts' experience and is not general. In contrast, deep learning-based methods [24], [48], [49], which can generate features automatically, have recently attracted increasing attention in hyperspectral image classification. The first attempt can be found in [23], where stacked autoencoder was utilized for high-level feature extraction. Subsequently, Li *et al.* [50] used restricted Boltzmann machine and deep belief network for hyperspectral image feature extraction and pixel classification, by which the information contained in the original data can be well retained. Meanwhile, the RNN model has been applied to hyperspectral image classification [51]. Shi and Pun [51] exploited multiscale spectral–spatial features via hierarchical RNN, which can learn the spatial dependence of nonadjacent image patches in a 2-D spatial domain. Among these deep learning methods, CNN, which needs fewer parameters than fully connected networks with the same number of hidden layers, has drawn great attention for its breakthrough in hyperspectral image classification. For example, in [29] and [52], CNN was used to extract the spectral features, which performs better than SVM. Nonetheless, excavating spatial information is of great importance in hyperspectral image classification and many CNN-based methods have done explorations on this aspect. For instance, Yang *et al.* [53] proposed a two-channel deep CNN to jointly learn spectral–spatial features from hyperspectral images, where the channels are used for learning spectral and spatial features, respectively. Besides, Yue *et al.* [54] projected hyperspectral data to several principal components before adopting CNN to extract spectral–spatial features. In the recent work of Li *et al.* [55], deep CNN is used to learn pixel-pair features, and the classification results of pixels in different pairs from the neighborhood are then fused. Additionally, Zhang *et al.* [14] proposed a deep CNN model based on diverse regions, which employs different local or global regions inputs to learn joint representation of each pixel. Although CNN-based hyperspectral image classification methods can extract spectral-spatial features automatically, the effectiveness of the obtained features is still restricted by some issues. For example, they simply apply the fixed convolution kernels to different regions in a hyperspectral image, which does not consider the geometric appearance of various local regions and may result in undesirable misclassifications.

B. Graph Convolutional Network

The concept of neural network for graph data was first proposed by Gori *et al.* [56], of which the advantage over CNN and RNN is that it can work on the graph-structured non-Euclidean data. Specifically, the graph neural network (GNN) can collectively aggregate the node features in a graph and properly embed the entire graph in a new discriminative space. Subsequently, Scarselli *et al.* [57] made GNN trainable by a supervised learning algorithm for practical data. However, their algorithm is computationally expensive and runs inefficiently on large-scale graphs. Therefore, Bruna *et al.* [58] developed the operation of "graph convolution" based on spectral property, which convolves on the neighborhood of every graph node and produces a node-level output. After that, many extensions of graph convolution have been investigated and achieved advanced results [59], [60]. For instance, Hamilton *et al.* [61] presented an inductive framework called "GraphSAGE," which leverages node features to effectively generate node embeddings for previously unseen data. Apart from this, Defferrard *et al.* [33] proposed a formulation of CNNs in the context of spectral graph theory. Based on their

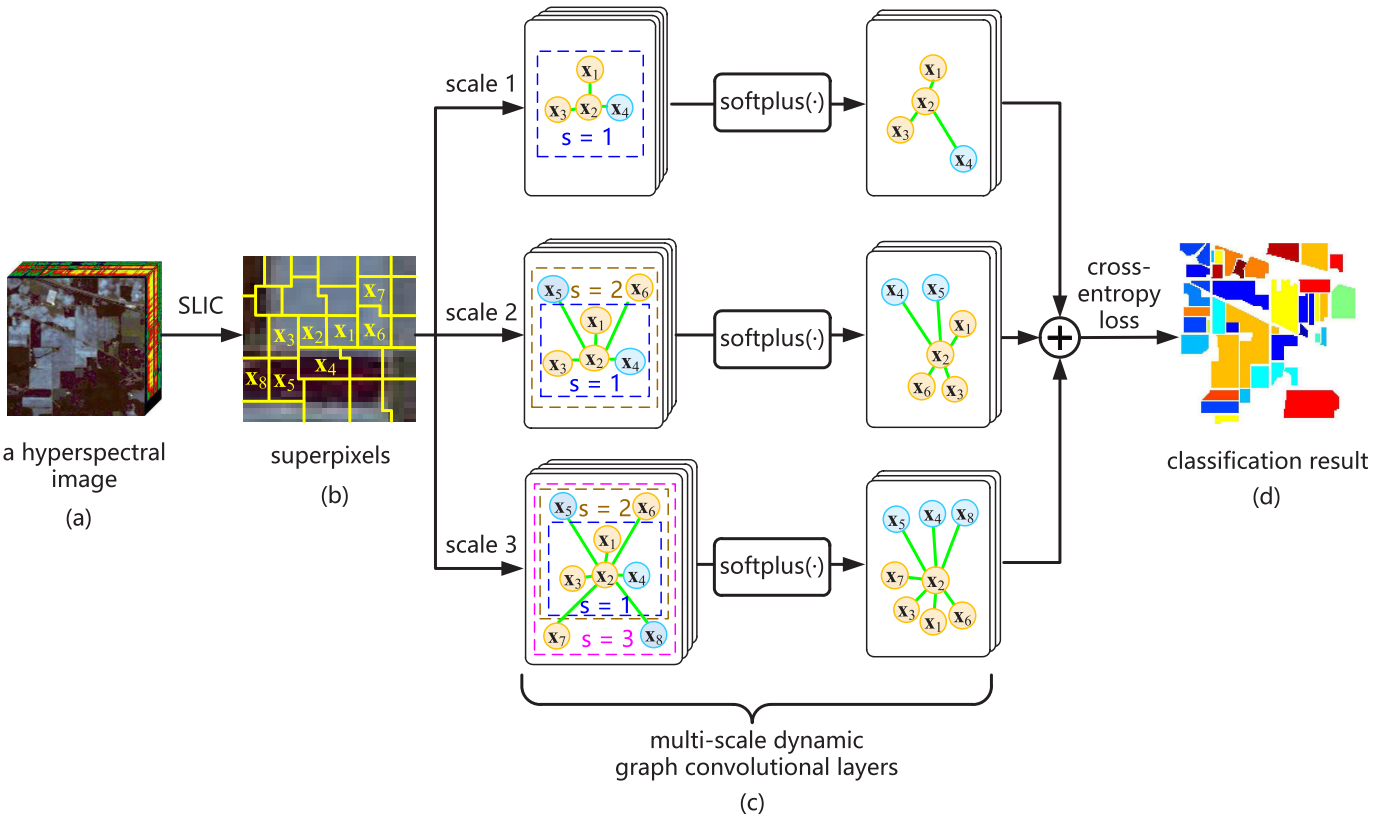

Fig. 1. Framework of our algorithm. (a) Original hyperspectral image. (b) Superpixels segmented by the SLIC algorithm [65], where a local region of the hyperspectral image is exhibited, which contains eight superpixels $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_8$. In (c), the circles and green lines represent the graph nodes and edges, respectively, where different colors of the nodes represent different land-cover types. Specifically, at each scale, the edge weight is updated gradually along with the convolution on graph nodes so that the graph can be dynamically refined. Here, two dynamic graph convolutional layers are employed for each scale, where softplus [66] is utilized as the activation function. In (d), the classification result is acquired by integrating the multiscale outputs, and the cross-entropy loss is used to penalize the label difference between the output and the seed superpixels.

work, Kipf and Welling [34] proposed a fast approximation localized convolution, which makes the GCN model able to encode both graph structure and node features. In their work, GCN was simplified by a first-order approximation of graph spectral convolution, which leads to more efficient filtering operations.

With the rapid development of graph convolution theories, GCN has been widely applied to various applications, such as recommender systems [62] and semantic segmentation [63]. Besides, to the best of our knowledge, GCN has been deployed for hyperspectral image classification in only one prior work [64]. However, [64] only utilizes a fixed graph during the node convolution process, and thus, the intrinsic relationship among the pixels cannot be precisely reflected. Moreover, the neighborhood size in their method is also fixed, and thus, the spectral–spatial information in different local regions cannot be flexibly captured. To cope with these issues, we propose a novel dynamic multiscale GCN that dynamically updates the graphs and fuses multiscale spectral–spatial information for hyperspectral image classification. As a result, the accurate node embeddings can be acquired, which ensures satisfactory classification performance.

III. Proposed Method

This section details our proposed MDGCN model (see Fig. 1). When an input hyperspectral image is given, it is preprocessed by the simple linear iterative clustering (SLIC) algorithm [65] to be segmented into several homogeneous superpixels. Then, graphs are constructed over these superpixels at different spatial scales. After that, the convolutions are conducted on these graphs, which simultaneously aggregates multiscale spectral–spatial features and also gradually refine the input graphs. The superpixels potentially belonging to the same class will be ideally clustered together in the embedding space. Finally, the classification result is produced by the well-trained network. Next, we detail the critical steps of our MDGCN by explaining the superpixel segmentation (see Section III-A), presenting the GCN backbone (see Section III-B), elaborating the dynamic graph evolution (see Section III-C), and describing the multiscale manipulation (see Section III-D).

A. Superpixel Segmentation

A hyperspectral image usually contains hundreds of thousands of pixels, which may result in unacceptable computational complexity for the subsequent graph convolution and classification. To address this problem, we adopt a segmentation algorithm named SLIC [65] to segment the entire image into a small amount of compact superpixels, and each superpixel represents a homogeneous image region with strong spectral–spatial similarity. Concretely, the SLIC algorithm starts from an initial grid on the image and then creates

segmentation through iteratively growing the local clusters using a k-means algorithm. When the segmentation is finished, each superpixel is treated as a graph node instead of the pixel in the input image; therefore, the amount of graph nodes can be significantly reduced, and the computational efficiency can be improved. Here, the feature of each node (i.e., superpixel) is the average spectral signatures of the pixels involved in the corresponding superpixel. Another advantage for implementing the superpixel segmentation is that the generated superpixels also help to preserve the local structural information of a hyperspectral image, as nearby pixels with high spatial consistency have a large probability to belong to the same land-cover type (i.e., label).

B. GCN

GCN [34] is a multilayer neural network, which operates directly on a graph and generates node embeddings by gradually fusing the features in the neighborhood. Different from traditional CNN that only applies to data represented by regular grids, GCN is able to operate on the data with arbitrary non-Euclidean structure. Formally, an undirected graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. The notation $\mathbf{A}$ denotes the adjacency matrix of $\mathcal{G}$, which indicates whether each pair of nodes is connected and can be calculated as

$$\mathbf{A}_{ij} = \begin{cases} e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, & \text{if } \mathbf{x}_i \in Nei(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in Nei(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where the parameter γ is empirically set to 0.2 in the experiments, $\mathbf{x}_i$ represents a superpixel and $Nei(\mathbf{x}_j)$ is the set of neighbors of the example $\mathbf{x}_j$.

To conduct node embeddings for $\mathcal{G}$, spectral filtering on graphs is defined, which can be expressed as a signal $\mathbf{x}$ filtered by $g_{\boldsymbol{\theta}} = \text{diag}(\boldsymbol{\theta})$ in the Fourier domain, namely

$$g_{\boldsymbol{\theta}} \star \mathbf{x} = \mathbf{U} g_{\boldsymbol{\theta}} \mathbf{U}^\top \mathbf{x} \tag{2}$$

where $\mathbf{U}$ is the matrix composed of the eigenvectors of the normalized graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{U}\Lambda\mathbf{U}^\top$. Here, Λ is a diagonal matrix containing the eigenvalues of $\mathbf{L}$, $\mathbf{D}$ is the degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, and $\mathbf{I}$ denotes the identity matrix with proper size throughout this article. Then, we can understand $g_{\boldsymbol{\theta}}$ as a function of the eigenvalues of $\mathbf{L}$, i.e., $g_{\boldsymbol{\theta}}(\Lambda)$. To reduce the computational consumption of eigenvector decomposition in (2), Hammond *et al.* [67] approximated $g_{\boldsymbol{\theta}}(\Lambda)$ by a truncated expansion in terms of Chebyshev polynomials $T_k(\mathbf{x})$ up to Kth order, which is

$$g_{\boldsymbol{\theta}'}(\Lambda) \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k(\widetilde{\Lambda}) \tag{3}$$

where $\boldsymbol{\theta}'$ is a vector of Chebyshev coefficients; $\widetilde{\Lambda} = 2/(\lambda_{\max})\Lambda - \mathbf{I}$ with $\lambda_{\max}$ being the largest eigenvalue of $\mathbf{L}$. According to [67], the Chebyshev polynomials are defined as $T_k(\mathbf{x}) = 2\mathbf{x}T_{k-1}(\mathbf{x}) - T_{k-2}(\mathbf{x})$ with $T_0(\mathbf{x}) = 1$ and $T_1(\mathbf{x}) = \mathbf{x}$. Therefore, the convolution of a signal $\mathbf{x}$ by the filter $g_{\boldsymbol{\theta}'}$ can be written as

$$g_{\boldsymbol{\theta}'} \star \mathbf{x} \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k(\widetilde{\mathbf{L}})\mathbf{x} \tag{4}$$

where $\widetilde{\mathbf{L}} = 2/(\lambda_{\max})\mathbf{L} - \mathbf{I}$ denotes the scaled Laplacian matrix. Equation (4) can be easily verified by using the fact $(\mathbf{U}\Lambda\mathbf{U}^\top)^k = \mathbf{U}\Lambda^k\mathbf{U}^\top$. It can be observed that this expression is a Kth-order polynomial regarding the Laplacian (i.e., K-localized), that is to say, the filtering only depends on the nodes that are at most K steps away from the central node. In this article, we consider the first-order neighborhood, i.e., $K = 1$, and thus, (4) becomes a linear function on the graph Laplacian spectrum with respect to $\mathbf{L}$.

After that, we can build a neural network based on graph convolutions by stacking multiple convolutional layers in the form of (4), and each layer is followed by an element-wise nonlinear operation softplus$(\cdot)$ [66]. In this way, we can acquire a diverse class of convolutional filter functions by stacking multiple layers with the same configuration. With the linear formulation, Kipf and Welling [34] further approximated $\lambda_{\max} \approx 2$, as the neural network parameters can adapt to this change in scale during the training process. Therefore, (4) can be simplified to

$$g_{\boldsymbol{\theta}'} \star \mathbf{x} \approx \boldsymbol{\theta}'_0\mathbf{x} + \boldsymbol{\theta}'_1(\mathbf{L} - \mathbf{I})\mathbf{x} = \boldsymbol{\theta}'_0\mathbf{x} - \boldsymbol{\theta}'_1\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x} \tag{5}$$

where $\boldsymbol{\theta}'_0$ and $\boldsymbol{\theta}'_1$ are two free parameters. Since reducing the number of parameters is beneficial to address overfitting, (5) is converted to

$$g_{\boldsymbol{\theta}} \star \mathbf{x} \approx \boldsymbol{\theta}(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{x} \tag{6}$$

by letting $\boldsymbol{\theta} = \boldsymbol{\theta}'_0 = -\boldsymbol{\theta}'_1$. Since $\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ has the eigenvalues in the range $[0, 2]$, repeatedly applying this operator will lead to numerical instabilities and exploding/vanishing gradients in a deep neural network. To cope with this problem, Kipf and Welling [34] performed the renormalization trick $\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \rightarrow \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}$ with $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$. As a result, the convolution operation of GCN model can be expressed as

$$\mathbf{H}^{(l)} = \sigma(\widetilde{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}) \tag{7}$$

where $\mathbf{H}^{(l)}$ is the output (namely, embedding result) of the lth layer; $\sigma(\cdot)$ represents an activation function, such as the softplus function [66] used in this article; and $\mathbf{W}^{(l)}$ denotes the trainable weight matrix included by the lth layer.

C. Dynamic Graph Evolution

As mentioned in Section I, one major disadvantage of conventional GCN is that the graph is fixed throughout the convolution process, which will degrade the final classification performance if the input graph is not accurate. To remedy this defect, in this article, we propose a dynamic GCN in which the graph can be gradually refined during the convolution process. The main idea is to find an improved graph by fusing the information of current data embeddings and the graph in the previous layer.

In the lth layer, we define an n-dimensional random variable $\overline{\mathbf{x}}^{(l)} \in \mathbb{R}^n$ corresponding to the data $\mathbf{x}^{(l)} \in \mathbb{R}^d$, where d is the number of spectral bands. Specifically, the elements of $\overline{\mathbf{x}}^{(l)}$ represent the similarities between $\mathbf{x}^{(l)}$ and all the n examples [i.e., the row of $\boldsymbol{\mathcal{A}}^{(l)}$ corresponding to $\mathbf{x}^{(l)}$]. Here, $\boldsymbol{\mathcal{A}}^{(l)} \in \mathbb{R}^{n\times n}$ is the adjacency matrix in the lth layer that carries

full pair-wise similarities among all the n examples. Then, according to [68] and [69], we may assume that the random variable $\overline{\mathbf{x}}^{(l)} \in \mathbb{R}^n$ satisfies Gaussian distribution, namely, $p(\overline{\mathbf{x}}^{(l)}) = \mathcal{N}(\overline{\mathbf{x}}^{(l)}|\overline{\boldsymbol{\mu}}^{(l)}, \boldsymbol{\mathcal{A}}^{(l)})$, where $\overline{\boldsymbol{\mu}}^{(l)}$ is the unknown mean. Similarly, we can also assume that the random variable $\overline{\mathbf{h}}^{(l)}$ corresponding to the embedding result $\mathbf{h}^{(l)}$ is Gaussian distributed with the embedding kernel $\mathbf{K}_E = \mathbf{H}^{(l)}\mathbf{H}^{(l)\top}$ being the covariance, where $\mathbf{K}_E \in \mathbb{R}^{n\times n}$ encodes the full pair-wise similarities among the embeddings generated by the lth layer. Based on the definitions mentioned earlier, the fused kernel can be obtained by linearly combining $\boldsymbol{\mathcal{A}}^{(l)}$ and $\mathbf{K}_E$, namely

$$\mathbf{F}^{(l)} = \boldsymbol{\mathcal{A}}^{(l)} + \alpha \mathbf{K}_E \tag{8}$$

where α is the weight assigned to the embedding kernel $\mathbf{K}_E$. The operation in (8) actually corresponds to the addition operator: $\overline{\mathbf{z}}^{(l)} = \overline{\mathbf{x}}^{(l)} + \sqrt{\alpha}\overline{\mathbf{h}}^{(l)}$, where $\overline{\mathbf{z}}^{(l)}$ is the fused result. Referring to the kernel fusion technique in [68], we obtain $p(\overline{\mathbf{z}}^{(l)}) = \mathcal{N}(\overline{\mathbf{z}}^{(l)}|\overline{\boldsymbol{\mu}}^{(l)}, \boldsymbol{\mathcal{A}}^{(l)} + \alpha\mathbf{H}^{(l)}\mathbf{H}^{(l)\top}) = \mathcal{N}(\overline{\mathbf{z}}^{(l)}|\overline{\boldsymbol{\mu}}^{(l)}, \mathbf{F}^{(l)})$, where $\overline{\boldsymbol{\mu}}^{(l)}$ is the unknown mean.

From (8), we can see that this fusion technique utilizes the information of the embedding results encoded in $\mathbf{K}_E$ and also the previous adjacency matrix $\boldsymbol{\mathcal{A}}^{(l)}$ to refine the graph. The advantages of such strategy are twofold. First, the introduction of embedding information helps to find a more accurate graph. Second, the improved graph will in turn make the embeddings more discriminative. However, there are still some problems regarding the fused kernel $\mathbf{F}^{(l)}$. The fusion in (8) will lead to performance degradation if the embeddings are not sufficiently accurate to characterize the intrinsic similarities of the input data. As a result, according to [69], we need to reemphasize the inherent structure among the input data carried by the initial adjacency matrix. Therefore, we do the following projection on the fused result $\overline{\mathbf{z}}^{(l)}$ by using the initial adjacency matrix $\mathbf{A}$, which leads to:

$$\overline{\mathbf{x}}^{(l+1)} = \mathbf{A}\overline{\mathbf{z}}^{(l)} + \beta^{(l)}\boldsymbol{\varepsilon} \tag{9}$$

where $\boldsymbol{\varepsilon}$ denotes white noise, i.e., $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon}|0, 1)$; the parameter $\beta^{(l)}$ is used to control the relative importance of $\boldsymbol{\varepsilon}$. With this projection, we have

$$p(\overline{\mathbf{x}}^{(l+1)}|\overline{\mathbf{z}}^{(l)}) = \mathcal{N}(\overline{\mathbf{x}}^{(l+1)}|\mathbf{A}\overline{\mathbf{z}}^{(l)}, \beta^{(l)}\mathbf{I}) \tag{10}$$

where $\mathbf{I}$ is an identity matrix. Therefore, the marginal distribution of $\overline{\mathbf{x}}^{(l+1)}$ is

$$\begin{aligned} p(\overline{\mathbf{x}}^{(l+1)}) &= \int \mathcal{N}(\overline{\mathbf{z}}^{(l)}|\overline{\boldsymbol{\mu}}^{(l)}, \mathbf{F}^{(l)})\mathcal{N}(\overline{\mathbf{x}}^{(l+1)}|\mathbf{A}\overline{\mathbf{z}}^{(l)}, \beta^{(l)}\mathbf{I})d\overline{\mathbf{z}}^{(l)} \\ &= \mathcal{N}(\overline{\mathbf{x}}^{(l+1)}|\mathbf{A}\overline{\boldsymbol{\mu}}^{(l)}, \mathbf{A}\mathbf{F}^{(l)}\mathbf{A}^{\top} + \beta^{(l)}\mathbf{I}). \end{aligned} \tag{11}$$

Since $\overline{\mathbf{x}}^{(l+1)}$ is Gaussian distributed with the covariance $\boldsymbol{\mathcal{A}}^{(l+1)}$, the adjacency matrix $\boldsymbol{\mathcal{A}}^{(l+1)}$ can be dynamically updated as

$$\boldsymbol{\mathcal{A}}^{(l+1)} \leftarrow \mathbf{A}(\boldsymbol{\mathcal{A}}^{(l)} + \alpha\mathbf{H}^{(l)}\mathbf{H}^{(l)\top})\mathbf{A}^{\top} + \beta^{(l)}\mathbf{I}. \tag{12}$$

To start the iteration depicted by (12), $\boldsymbol{\mathcal{A}}^{(1)}$ can be calculated as $\boldsymbol{\mathcal{A}}_{ij}^{(1)} = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ with γ being the tuning parameter. Here, in order to reduce the disturbance of noise and improve the quality of graph convolution, we only consider the correlations among the neighboring examples. Therefore, the adjacency matrix used in the lth graph convolutional layer can be computed as

$$\mathbf{A}_{ij}^{(l)} = \begin{cases} \boldsymbol{\mathcal{A}}_{ij}^{(l)}, & \text{if } \mathbf{x}_i \in Nei(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in Nei(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

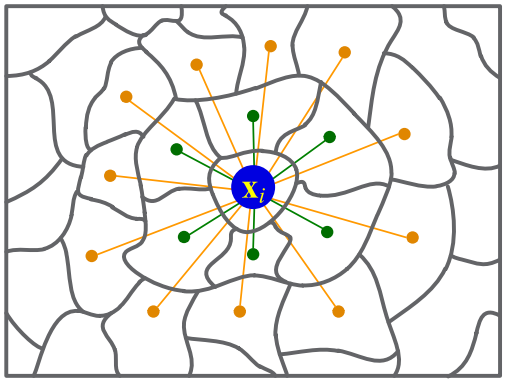

Fig. 2. Illustration of multiple scales considered by our method. The green nodes denote the one-hop neighbors of $\mathbf{x}_i$, and the orange nodes together with the green nodes represent $\mathbf{x}_i$'s two-hop neighbors.

D. Multiscale Manipulation

Multiscale information has been widely demonstrated to be useful for hyperspectral image classification problems [70], [71]. This is because the objects in a hyperspectral image usually have different geometric appearances, and the contextual information revealed by different scales helps to exploit the abundant local property of image regions from diverse levels. In our method, the multiscale spectral–spatial information is captured by constructing graphs at different neighborhood scales. Specifically, at the scale s, every superpixel $\mathbf{x}_i$ is connected to its s-hop neighbors. Fig. 2 exhibits the one- and two-hop neighbors of a central example $\mathbf{x}_i$ to illustrate the multiscale design. Then, the receptive field of $\mathbf{x}_i$ at the scale s is formed as

$$R_s(\mathbf{x}_i) = R_{s-1}(\mathbf{x}_i) \cup R_1(R_{s-1}(\mathbf{x}_i)) \tag{14}$$

where $R_0(\mathbf{x}_i) = \mathbf{x}_i$ and $R_1(\mathbf{x}_i)$ is the set of one-hop neighbors of $\mathbf{x}_i$. By considering both the effectiveness and the efficiency, in our method, we construct the graphs at scales 1–3. Therefore, the formulation of the graph convolutional layer is expressed as

$$\mathbf{H}_s^{(l)} = \sigma\left(\mathbf{A}_s^{(l)}\mathbf{H}_s^{(l-1)}\mathbf{W}_s^{(l)}\right) \tag{15}$$

where $\mathbf{A}_s^{(l)}, \mathbf{H}_s^{(l)}$, and $\mathbf{W}_s^{(l)}$ denote the adjacency matrix, the output matrix, and the trainable weight matrix of the lth graph convolutional layer at the scale s, respectively. Note that the input matrix $\mathbf{H}^{(0)}$ is shared by all scales. Based on (15), the output of MDGCN can be obtained by

$$\mathbf{O} = \sum\nolimits_s \mathbf{H}_s^{(L)} \tag{16}$$

where L is the number of graph convolutional layers shared by all scales and $\mathbf{O}$ is the output of MDGCN. The convolution process of MDGCN is summarized in Algorithm 1. In our model, the cross-entropy error is adopted to penalize the

Algorithm 1 Multiscale Dynamic Convolution Process of MDGCN

Input: Input matrix $\mathbf{H}^{(0)}$; number of scales S; number of graph convolutional layers L; initial adjacency matrices $\mathbf{A}_s^{(1)}$ $(1 \le s \le S)$;

1: **for** $l = 1$ to L **do**
2: Calculate the outputs of the l^{th} layer $\mathbf{H}_s^{(l)}$ $(1 \le s \le S)$ according to Eq. (15);
3: Update the graphs $\mathbf{A}_s^{(l+1)}$ $(1 \le s \le S)$ according to Eq. (12) and Eq. (13);
4: **end for**
5: Calculate the network output according to Eq. (16);

Output: Network output $\mathbf{O}$.

Algorithm 2 Proposed MDGCN for Hyperspectral Image Classification

Input: Input image; number of iterations $\mathcal{T} = 5000$; learning rate $\eta = 0.0005$; number of scales $S = 3$; number of graph convolutional layers $L = 2$;

1: Segment the whole image into superpixels via SLIC algorithm;
2: Construct the initial adjacency matrices $\mathbf{A}_s^{(1)}$ $(1 \le s \le S)$ according to Eq. (1);
3: // Train the MDGCN model
4: **for** $t = 1$ to $\mathcal{T}$ **do**
5: Conduct multi-scale dynamic convolution by Algorithm 1;
6: Calculate the error term according to Eq. (17), and update the weight matrices $\mathbf{W}_s^{(l)}$ $(1 \le l \le L, 1 \le s \le S)$ using full-batch gradient descent;
7: **end for**
8: Conduct label prediction by Algorithm 1;

Output: Predicted label for each superpixel.

difference between the network output and the labels of the original labeled examples, which is

$$\mathcal{L} = -\sum_{g \in \mathbf{y}_G} \sum_{f=1}^{C} \mathbf{Y}_{gf} \ln \mathbf{O}_{gf} \tag{17}$$

where $\mathbf{y}_G$ is the set of indices corresponding to the labeled examples, C denotes the number of classes, and $\mathbf{Y}$ denotes the label matrix. Similar to [34], the network parameters here are learned by using full-batch gradient descent, where all superpixels are utilized to perform gradient descent in each iteration. The implementation details of our MDGCN are shown in Algorithm 2.

IV. Experimental Results

In this section, we conduct exhaustive experiments to validate the effectiveness of the proposed MDGCN method and also provide the corresponding algorithm analyses. To be specific, we first compare MDGCN with other state-of-the-art approaches on three publicly available hyperspectral image data sets, where four metrics, including per-class accuracy, overall accuracy (OA), average accuracy (AA), and kappa coefficient, are adopted. Then, we demonstrate that both the multiscale manipulation and the dynamic graph design in our MDGCN are beneficial to obtaining the promising performance. After that, we validate the effectiveness of our method in dealing with the boundary regions. Finally, we compare the computational time of various methods to show the efficiency of our algorithm.

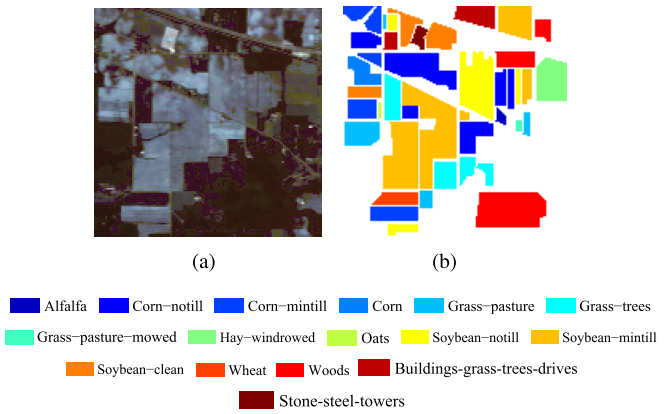

Fig. 3. Indian Pines. (a) False-color image. (b) Ground-truth map.

A. Data Sets

The performance of the proposed MDGCN is evaluated on three data sets, i.e., the Indian Pines, the University of Pavia, and the Kennedy Space Center, which will be introduced next.

1) Indian Pines: The Indian Pines data set was collected by the Airborne Visible/Infrared Imaging Spectrometer sensor in 1992, which records northwestern India. It consists of 145 × 145 pixels with a spatial resolution of 20 m × 20 m and has 220 spectral channels covering the range from 0.4 to 2.5 μm. As a usual step, 20 water absorption and noisy bands are removed, and 200 bands are reserved. The original ground truth includes 16 land-cover classes, such as "Alfalfa," "Corn-notill," and "Corn-mintill." Fig. 3 shows the false-color image and ground-truth map of the Indian Pines data set. The amounts of labeled and unlabeled pixels of various classes are listed in Table I.

2) University of Pavia: The University of Pavia data set captured the Pavia University in Italy with the ROSIS sensor in 2001. It consists of 610 × 340 pixels with a spatial resolution of 1.3 m × 1.3 m and has 103 spectral channels in the wavelength range from 0.43 to 0.86 μm after removing noisy bands. This data set includes nine land-cover classes, such as "Asphalt," "Meadows," and "Gravel," which are shown in Fig. 4. Table II lists the amounts of labeled and unlabeled pixels of each class.

3) Kennedy Space Center: The Kennedy Space Center data set was taken by the AVIRIS sensor over Florida with a spectral coverage ranging from 0.4 to 2.5 μm. This data set contains 224 bands and 614 × 512 pixels with a spatial resolution of 18 m. After removing water absorption and noisy bands, the remaining 176 bands of the image have been preserved. The Kennedy Space Center data set includes 13 land-cover classes, such as "Srub," "Willow swamp," and "CP hammock." Fig. 5 shows the false-color image and

TABLE I

NUMBERS OF LABELED AND UNLABELED PIXELS OF ALL CLASSES IN THE INDIAN PINES DATA SET

ID	Class	#Labeled	#Unlabeled
1	Alfalfa	30	16
2	Corn-notill	30	1398
3	Corn-mintill	30	800
4	Corn	30	207
5	Grass-pasture	30	453
6	Grass-trees	30	700
7	Grass-pasture-mowed	15	13
8	Hay-windrowed	30	448
9	Oats	15	5
10	Soybean-notill	30	942
11	Soybean-mintill	30	2425
12	Soybean-clean	30	563
13	Wheat	30	175
14	Woods	30	1235
15	Buildings-grass-trees-drives	30	356
16	Stone-steel-towers	30	63

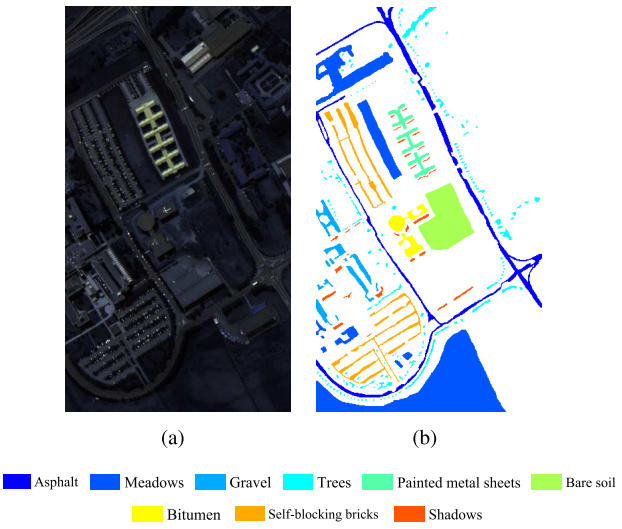

Fig. 4. University of Pavia. (a) False-color image. (b) Ground-truth map.

TABLE II

NUMBERS OF LABELED AND UNLABELED PIXELS OF ALL CLASSES IN THE UNIVERSITY OF PAVIA DATA SET

ID	Class	#Labeled	#Unlabeled
1	Asphalt	30	6601
2	Meadows	30	18619
3	Gravel	30	2069
4	Trees	30	3034
5	Painted metal sheets	30	1315
6	Bare soil	30	4999
7	Bitumen	30	1300
8	Self-blocking bricks	30	3652
9	Shadows	30	917

ground-truth map of the Kennedy Space Center data set. The numbers of labeled and unlabeled pixels of different classes are listed in Table III.

B. Experimental Settings

In our experiments, the proposed algorithm is implemented via TensorFlow with Adam optimizer. For all the adopted

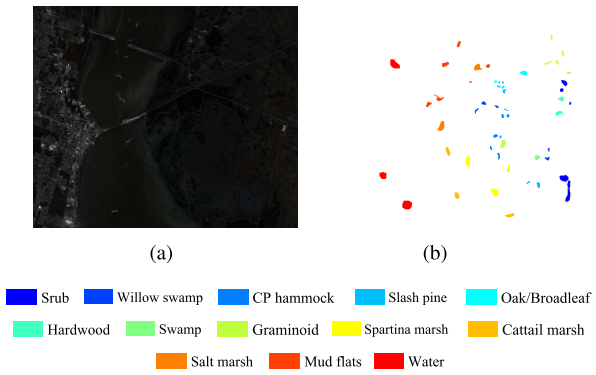

Fig. 5. Kennedy Space Center. (a) False-color image. (b) Ground-truth map.

TABLE III

NUMBERS OF LABELED AND UNLABELED PIXELS OF ALL CLASSES IN THE KENNEDY SPACE CENTER DATA SET

ID	Class	#Labeled	#Unlabeled
1	Srub	30	728
2	Willow swamp	30	220
3	CP hammock	30	232
4	Slash pine	30	228
5	Oak/Broadleaf	30	146
6	Hardwood	30	207
7	Swamp	30	96
8	Graminoid	30	393
9	Spartina marsh	30	469
10	Cattail marsh	30	365
11	Salt marsh	30	378
12	Mud flats	30	454
13	Water	30	836

three data sets introduced in Section IV-A, usually, 30 labeled pixels (i.e., examples) are randomly selected in each class for training, and only 15 labeled examples are chosen if the corresponding class has less than 30 examples. During training, 90% of the labeled examples are used to learn the network parameters and 10% are used as validation set to tune the hyperparameters. Meanwhile, all the unlabeled examples are used as the test set to evaluate the classification performance. The network architecture of our proposed MDGCN is kept identical for all the data sets. Specifically, three neighborhood scales, namely, $s = 1$, $s = 2$, and $s = 3$, are employed for graph construction to incorporate multiscale spectral–spatial information into our model. For each scale, we employ two graph convolutional layers with 20 hidden units, as GCN-based methods usually do not require deep structure to achieve satisfactory performance [64], [72]. Besides, the learning rate and the number of training epochs are set to 0.0005 and 5000, respectively.

To evaluate the classification ability of our proposed method, other recent state-of-the-art hyperspectral image classification methods are also used for comparison. Specifically, we employ two CNN-based methods, i.e., diverse region-based deep CNN (DR-CNN) [14] and recurrent 2-D-CNN (R-2D-CNN) [20], together with two GCN-based methods, i.e., GCN [34] and spectral–spatial graph convolutional network (S^2GCN) [64]. Meanwhile, we compare

TABLE IV
PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON THE INDIAN PINES DATA SET

ID	GCN [34]	S^2GCN [64]	R-2D-CNN [20]	DR-CNN [14]	MDA [8]	HiFi [73]	JSDF [74]	MDGCN
1	95.00±2.80	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.38±1.92	**100.00±0.00**	**100.00±0.00**
2	56.71±4.42	84.43±2.50	54.94±2.23	80.38±1.50	75.08±7.35	87.42±4.29	**90.75±3.19**	80.18±0.84
3	51.50±2.56	82.87±5.53	73.31±4.33	82.21±3.53	81.44±5.03	93.39±2.81	77.84±3.81	**98.26±0.00**
4	84.64±3.16	93.08±1.95	84.06±12.98	99.19±0.74	95.29±3.02	97.68±2.76	**99.86±0.33**	98.57±0.00
5	83.71±3.20	**97.13±1.34**	87.64±0.31	96.47±1.10	91.72±3.56	94.33±2.71	87.20±2.73	95.14±0.33
6	94.03±2.11	97.29±1.27	91.21±4.34	98.62±1.90	95.46±1.76	**98.71±1.86**	98.54±0.28	97.16±0.57
7	92.31±0.00	92.31±0.00	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	95.00±3.36	**100.00±0.00**	**100.00±0.00**
8	96.61±1.86	99.03±0.93	99.11±0.95	99.78±0.22	**99.80±0.28**	99.59±0.48	**99.80±0.31**	98.89±0.00
9	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
10	77.47±1.24	**93.77±3.72**	70.81±5.11	90.41±1.95	81.95±4.68	91.37±3.49	89.99±4.24	90.02±1.02
11	56.56±1.53	84.98±2.82	56.35±1.08	74.46±0.37	69.13±7.07	84.33±3.55	76.75±5.12	**93.35±1.47**
12	58.29±6.58	80.05±5.17	63.06±12.81	91.00±3.14	76.58±5.12	**95.02±3.10**	87.10±2.82	93.05±2.30
13	**100.00±0.00**	99.43±0.00	98.86±1.62	**100.00±0.00**	99.43±0.75	99.29±0.25	99.89±0.36	**100.00±0.00**
14	80.03±3.93	96.73±0.92	88.74±2.58	91.85±3.40	90.92±3.21	98.32±0.76	97.21±2.78	**99.72±0.05**
15	69.55±6.66	86.80±3.42	87.08±2.78	99.44±0.28	91.57±5.27	96.71±2.06	99.58±0.68	**99.72±0.00**
16	98.41±0.00	**100.00±0.00**	97.62±1.12	**100.00±0.00**	96.63±4.91	99.13±0.81	**100.00±0.00**	95.71±0.00
OA	69.24±1.56	89.49±1.08	72.11±1.28	86.65±0.59	81.91±1.33	91.90±1.36	88.34±1.39	**93.47±0.38**
AA	80.93±1.71	92.99±1.04	84.55±1.79	93.99±0.25	90.31±0.71	95.60±0.58	94.03±0.55	**96.24±0.21**
Kappa	65.27±1.80	88.00±1.23	68.66±1.46	84.88±0.67	79.54±1.46	90.77±1.53	86.80±1.55	**92.55±0.43**

the proposed MDGCN with three traditional hyperspectral image classification methods, namely, matrix-based discriminant analysis (MDA) [8], hierarchical guidance filtering-based ensemble classification (HiFi) [73], and joint collaborative representation and SVM with decision fusion (JSDF) [74], respectively. All these methods are implemented ten times on each data set, and the mean accuracies and standard deviations over these ten independent implementations are reported.

C. Classification Results

To show the effectiveness of our proposed MDGCN, here we quantitatively and qualitatively evaluate the classification performance by comparing MDGCN with the aforementioned baseline methods.

1) Results on the Indian Pines Data Set: The quantitative results obtained by different methods on the Indian Pines data set are summarized in Table IV, where the highest value in each row is highlighted in bold. We observe that the proposed MDGCN achieves the top-level performance among all the methods in terms of OA, AA, and Kappa coefficient, and the standard deviations are also very small. Therefore, it is reasonable to infer that the proposed MDGCN is more stable and effective than the compared methods.

Fig. 6 shows a visual comparison of the classification results generated by different methods on the Indian Pines data set, and the ground-truth map is provided in Fig. 6(b). Compared with the ground-truth map, it can be seen that some pixels of "Soybean-mintill" are misclassified into "Corn-notill" in all the classification maps because these two land-cover types have similar spectral signatures. Meanwhile, due to the lack of spatial context, the classification map obtained by GCN suffers from pepper-noise-like mistakes within certain regions. Comparatively, the result of the proposed MDGCN method yields a smoother visual effect and shows fewer misclassifications than other compared methods.

2) Results on the University of Pavia Data Set: Table V presents the quantitative results of different methods on the University of Pavia data set. Similar to the results on the Indian Pines data set, the results in Table V indicate that the proposed MDGCN is in the first place and outperforms the compared methods by a substantial margin, which again validates the strength of our proposed multiscale dynamic graph convolution. Besides, it is also notable that DR-CNN performs better than HiFi, which is different from the results on the Indian Pines data set. The main reason is that DR-CNN exploits diverse predefined convolutional kernels, which can flexibly capture the variations of contextual distribution around objects. Therefore, it can effectively perceive and handle the irregular class boundaries, and thus, the advantage of DR-CNN will become prominent on the data sets that contain many irregular class boundaries, such as the University of Pavia data set. Furthermore, from the classification results in Fig. 7, stronger spatial correlation and fewer misclassifications can be observed in the classification map of the proposed MDGCN when compared with DR-CNN and other competitors.

3) Results on the Kennedy Space Center Data Set: Table VI presents the experimental results of different methods on the Kennedy Space Center data set. It is apparent that the performance of all methods is better than that on the Indian Pines and University of Pavia data sets. This could be due to that the Kennedy Space Center data set has higher spatial resolution and contains less noise than the Indian Pines and the University of Pavia data sets and thus is more suitable for classification. Note that slight gaps can still be observed between HiFi and our MDGCN in terms of OA. For the proposed MDGCN, it is also worth noting that misclassifications only occur in the sixth class ("Hardwood"), which further demonstrates the advantage of our proposed MDGCN. Fig. 8 shows the classification results of the eight different methods, where some critical regions of each classification

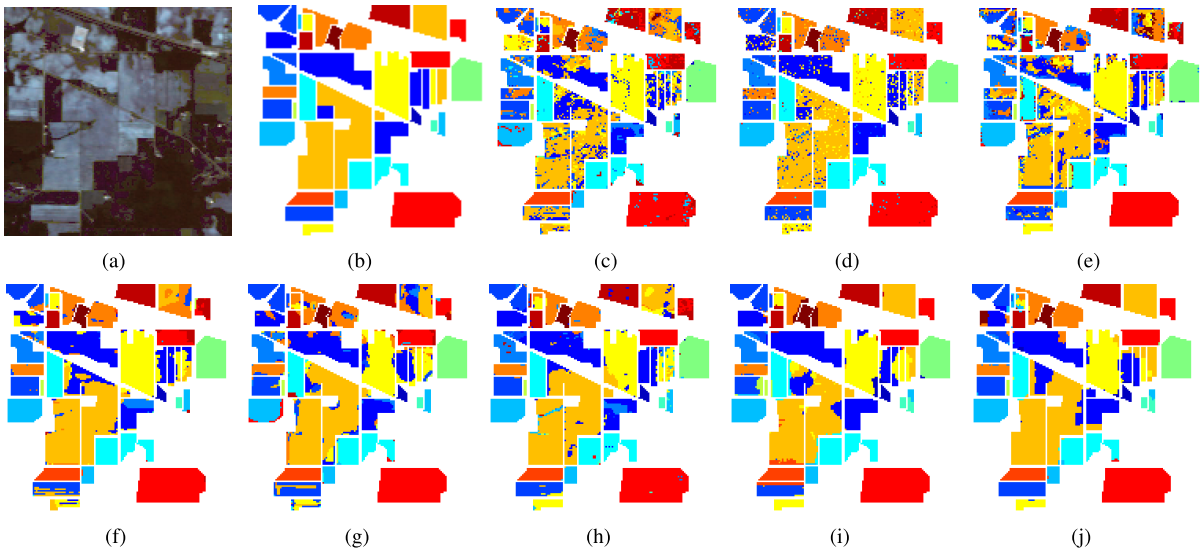

Fig. 6. Classification maps obtained by different methods on the Indian Pines data set. (a) False-color image. (b) Ground-truth map. (c) GCN. (d) S^2GCN. (e) R-2D-CNN. (f) DR-CNN. (g) MDA. (h) HiFi. (i) JSDF. (j) MDGCN.

TABLE V

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON THE UNIVERSITY OF PAVIA DATA SET

ID	GCN [34]	S^2GCN [64]	R-2D-CNN [20]	DR-CNN [14]	MDA [8]	HiFi [73]	JSDF [74]	MDGCN
1	69.78±4.71	92.87±3.79	84.96±0.56	92.10±3.34	78.84±1.99	77.77±4.52	82.40±4.07	**93.55±0.37**
2	54.10±10.54	87.06±4.47	79.99±2.29	96.39±3.20	79.96±3.41	95.49±2.07	90.76±3.74	**99.25±0.23**
3	69.69±4.48	87.97±4.77	89.49±0.17	84.23±0.71	85.58±3.46	**94.12±3.26**	86.71±4.14	92.03±0.24
4	91.23±7.02	90.85±0.94	**98.12±0.65**	95.26±0.67	90.90±2.29	82.68±4.25	92.88±2.16	83.78±1.55
5	98.74±0.11	**100.00±0.00**	99.85±0.11	97.77±0.00	99.93±0.08	97.25±1.67	**100.00±0.00**	99.47±0.09
6	65.34±10.53	88.69±2.64	76.79±7.40	90.44±2.27	75.52±7.11	**99.63±0.78**	94.30±4.55	95.26±0.50
7	86.64±4.68	98.88±1.08	88.69±4.57	89.05±1.76	84.28±5.11	97.77±2.43	96.62±1.37	**98.92±1.04**
8	72.26±2.63	89.97±3.28	67.54±5.67	78.49±1.53	81.82±6.92	**95.12±2.34**	94.69±3.74	94.99±1.33
9	**99.93±0.06**	98.89±0.53	99.84±0.08	96.34±0.22	97.50±1.48	83.86±3.40	99.56±0.36	81.03±0.49
OA	66.19±3.43	89.74±1.70	82.38±0.88	92.62±1.15	81.60±2.07	92.08±1.28	90.82±1.30	**95.68±0.22**
AA	78.63±1.23	92.80±0.47	87.25±0.68	91.12±0.12	86.04±1.67	91.52±0.99	93.10±0.65	**93.15±0.28**
Kappa	58.39±3.28	86.65±2.06	77.31±0.97	90.27±1.44	76.24±2.63	89.60±1.65	88.02±1.62	**94.25±0.29**

TABLE VI

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON THE KENNEDY SPACE CENTER DATA SET

ID	GCN [34]	S^2GCN [64]	R-2D-CNN [20]	DR-CNN [14]	MDA [8]	HiFi [73]	JSDF [74]	MDGCN
1	86.91±3.46	95.12±0.32	94.71±0.16	98.72±0.21	96.88±2.22	97.28±1.72	**100.00±0.00**	**100.00±0.00**
2	83.29±3.08	95.15±5.15	79.03±0.98	97.97±1.36	97.84±2.31	99.66±0.89	92.07±1.59	**100.00±0.00**
3	87.57±4.31	96.17±0.51	80.24±4.11	97.49±2.00	88.45±6.24	**100.00±0.00**	95.13±4.01	**100.00±0.00**
4	24.86±12.31	71.17±8.58	42.19±5.88	62.46±3.94	78.29±6.98	99.03±1.12	59.01±8.13	**100.00±0.00**
5	63.36±5.47	97.71±2.64	79.39±3.33	94.66±2.75	86.76±5.06	**100.00±0.00**	85.34±7.82	**100.00±0.00**
6	61.01±4.43	89.95±3.48	77.05±7.85	97.65±1.05	94.27±3.95	**99.21±1.00**	86.48±3.63	94.91±0.25
7	91.20±5.63	98.22±3.08	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	98.93±1.51	**100.00±0.00**
8	78.20±6.45	89.11±0.58	98.17±0.76	97.42±1.77	94.08±3.60	**100.00±0.00**	94.76±3.56	**100.00±0.00**
9	85.39±3.96	99.59±0.35	96.67±0.62	99.93±0.12	98.79±2.94	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
10	84.28±4.93	98.04±1.08	98.30±1.30	98.84±0.56	98.02±3.41	97.78±2.60	**100.00±0.00**	**100.00±0.00**
11	94.68±1.95	99.23±0.45	89.03±1.16	**100.00±0.00**	98.62±1.53	99.78±0.23	**100.00±0.00**	**100.00±0.00**
12	82.14±2.42	95.63±0.24	94.64±0.80	98.94±0.73	94.98±1.96	99.97±0.08	95.52±2.02	**100.00±0.00**
13	98.99±0.67	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
OA	83.60±0.81	95.44±0.92	91.11±0.60	97.21±0.27	95.92±0.81	99.30±0.39	95.69±0.34	**99.79±0.01**
AA	78.60±1.01	94.24±1.84	86.88±1.03	95.70±0.33	94.38±0.96	99.44±0.31	92.87±0.63	**99.61±0.02**
Kappa	81.70±0.90	94.91±1.03	90.06±0.66	96.88±0.30	95.44±0.91	99.22±0.44	95.17±0.37	**99.77±0.01**

map are enlarged for better performance comparison. From Fig. 8, we can see that our MDGCN is able to produce precise classification results on these small and difficult regions.

D. Impact of the Number of Labeled Examples

In this experiment, we investigate the classification accuracies of the proposed MDGCN and the other competitors under different numbers of labeled examples. To this end, we vary

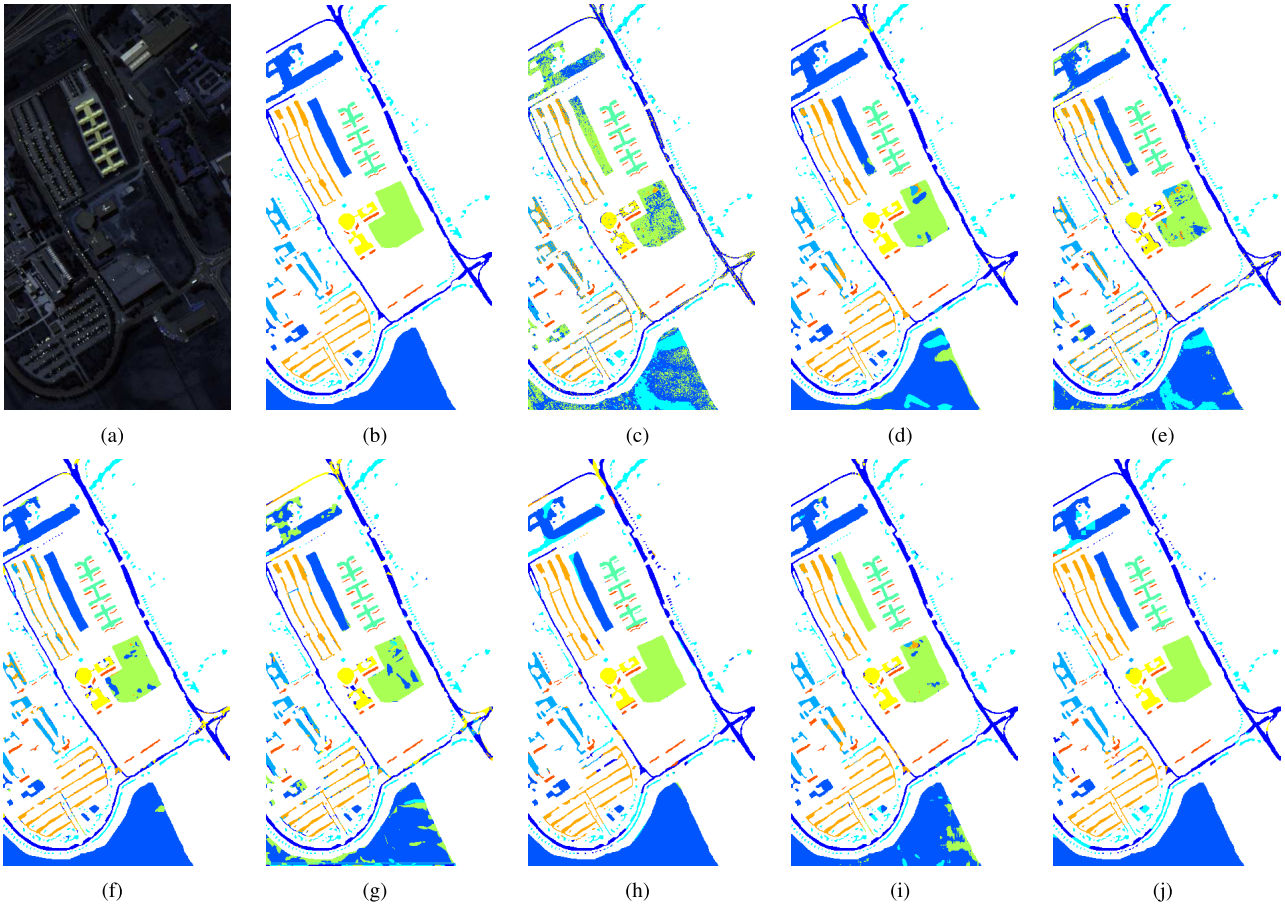

Fig. 7. Classification maps obtained by different methods on the University of Pavia data set. (a) False-color image. (b) Ground-truth map. (c) GCN. (d) S^2GCN. (e) R-2D-CNN. (f) DR-CNN. (g) MDA. (h) HiFi. (i) JSDF. (j) MDGCN.

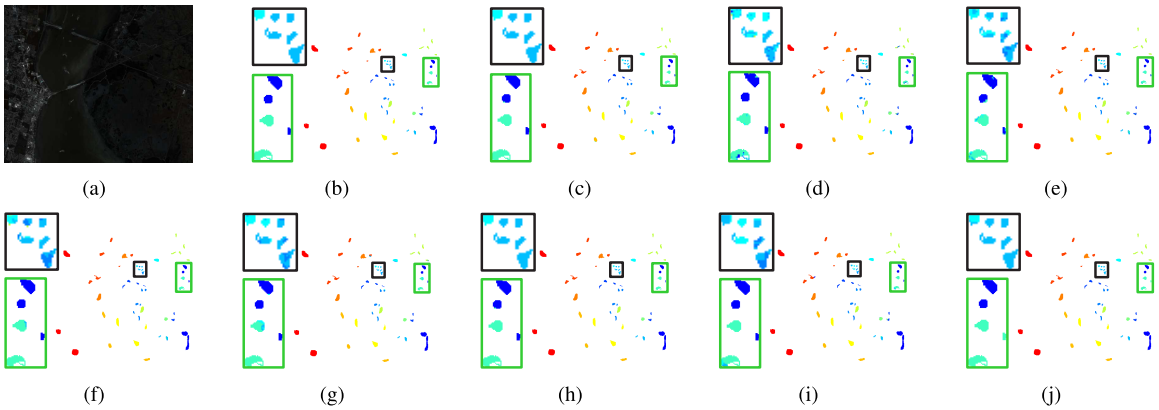

Fig. 8. Classification maps obtained by different methods on the Kennedy Space Center data set. (a) False-color image. (b) Ground-truth map. (c) GCN. (d) S^2GCN. (e) R-2D-CNN. (f) DR-CNN. (g) MDA. (h) HiFi. (i) JSDF. (j) MDGCN. In (b)–(j), zoomed-in views of the regions enclosed in black and green boxes are shown at the left side of each map.

the number of labeled examples per class from 5 to 30 and report the OA gained by all the methods on three data sets, i.e., the Indian Pines, the University of Pavia, and the Kennedy Space Center (see Fig. 9). We can make the observation from Fig. 9 that the performance of all methods can be improved by increasing the number of labeled examples. It is also noteworthy that the proposed MDGCN consistently yields higher OA than all the baseline methods with different numbers of labeled examples. Besides, the performance of MDGCN is more stable than the compared methods with the changed number of labeled examples. All these observations indicate the effectiveness and stability of our MDGCN method.

E. Ablation Study

As mentioned in Section I, our proposed MDGCN contains two critical parts for boosting the classification performance, i.e., multiscale operation and dynamic graph convolution.

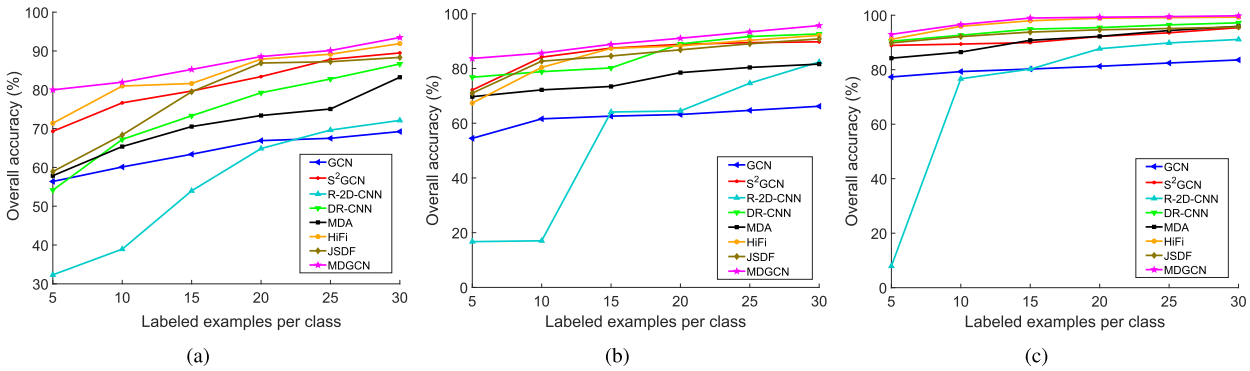

Fig. 9. Overall accuracies of various methods under different numbers of labeled examples per class. (a) Indian Pines data set. (b) University of Pavia data set. (c) Kennedy Space Center data set.

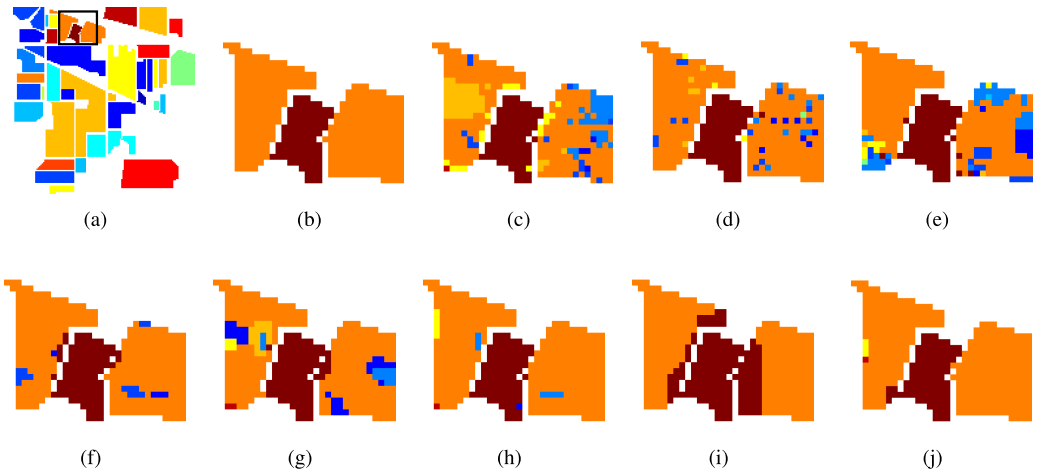

Fig. 10. Classification maps obtained by different methods regarding a boundary region in the Indian Pines data set. (a) Studied boundary region. (b) Ground-truth map. (c) GCN. (d) S^2GCN. (e) R-2D-CNN. (f) DR-CNN. (g) MDA. (h) HiFi. (i) JSDF. (j) MDGCN.

TABLE VII

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT GRAPH CONVOLUTION APPROACHES ON THE INDIAN PINES DATA SET

ID	$s=1$	$s=2$	$s=3$	MGCN	MDGCN
1	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
2	**81.15±1.06**	76.89±4.17	71.22±1.35	78.74±2.59	80.18±0.84
3	92.81±2.21	96.70±1.62	96.02±2.81	94.12±1.00	**98.26±0.00**
4	**100.00±0.00**	98.76±0.55	98.47±2.37	98.55±0.00	98.57±0.00
5	89.62±0.31	93.76±2.22	89.51±2.54	90.18±0.22	**95.14±0.33**
6	93.36±0.91	96.78±0.56	95.21±2.27	94.57±1.49	**97.16±0.57**
7	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
8	98.66±0.32	98.57±1.40	99.81±0.46	**100.00±0.00**	98.89±0.00
9	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
10	84.50±0.30	88.03±2.01	76.26±4.30	85.67±2.01	**90.02±1.02**
11	79.59±0.93	91.51±1.27	91.70±1.46	90.37±0.64	**93.35±1.47**
12	91.21±0.38	91.73±3.41	86.35±4.17	90.90±3.15	**93.05±2.30**
13	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
14	99.55±0.06	99.78±0.08	**99.87±0.07**	99.76±0.07	99.72±0.05
15	98.31±1.99	99.64±0.21	99.63±0.15	98.67±1.38	**99.72±0.00**
16	**98.41±0.00**	96.15±1.55	96.83±1.74	**98.41±0.00**	95.71±0.00
OA	88.53±0.54	92.03±0.39	89.53±0.55	91.24±0.70	**93.47±0.38**
AA	94.20±0.28	95.52±0.32	93.81±0.30	95.00±0.58	**96.24±0.21**
Kappa	86.97±0.60	90.90±0.44	88.06±0.63	90.00±0.80	**92.55±0.43**

TABLE VIII

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT GRAPH CONVOLUTION APPROACHES ON THE UNIVERSITY OF PAVIA DATA SET

ID	$s=1$	$s=2$	$s=3$	MGCN	MDGCN
1	84.90±3.63	76.49±5.58	63.79±6.31	85.70±5.56	**93.55±0.37**
2	93.46±3.79	94.26±2.86	95.88±1.19	92.94±3.34	**99.25±0.23**
3	84.87±7.82	90.87±4.43	**94.59±4.46**	92.11±4.31	92.03±0.24
4	81.48±3.29	65.28±4.59	62.49±7.11	**83.98±4.89**	83.78±1.55
5	98.01±1.33	98.40±0.99	98.24±1.68	96.53±2.38	**99.47±0.09**
6	98.46±2.55	**99.78±0.15**	97.71±2.08	98.65±2.82	95.26±0.50
7	97.38±2.43	96.38±2.79	97.80±1.16	96.61±2.37	**98.92±1.04**
8	92.84±1.49	91.40±3.50	87.47±5.11	88.56±5.34	**94.99±1.33**
9	88.63±2.97	79.14±3.08	73.28±4.10	**90.19±3.58**	81.03±0.49
OA	91.55±1.48	89.54±1.43	87.59±1.20	91.60±1.96	**95.68±0.22**
AA	91.11±0.95	88.00±1.26	85.69±0.98	91.70±1.53	**93.15±0.28**
Kappa	88.92±1.87	86.26±1.83	83.65±1.54	89.00±2.52	**94.25±0.29**

Here, we use the three data sets, i.e., the Indian Pines, the University of Pavia, and Kennedy Space Center, to demonstrate the usefulness of these two operations, where the number of labeled pixels per class is kept identical to the abovementioned experiments in Section IV-C. To show the importance of multiscale technique, we exhibit the classification results in Tables VII–IX, by using the dynamic graphs with three different neighborhood scales, i.e., $s = 1$, $s = 2$, and $s = 3$. It can be observed that higher neighborhood scale does not necessarily result in better performance, since the spectral–spatial information cannot be sufficiently exploited with only a single-scale graph, especially in complex image

TABLE IX

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT GRAPH CONVOLUTION APPROACHES ON THE KENNEDY SPACE CENTER DATA SET

ID	$s=1$	$s=2$	$s=3$	MGCN	MDGCN
1	98.77±1.70	98.69±1.70	98.54±2.75	98.42±1.85	**100.00±0.00**
2	99.06±0.00	99.34±2.08	98.08±4.31	99.60±1.58	**100.00±0.00**
3	**100.00±0.00**	98.67±2.80	95.62±4.48	98.52±4.56	**100.00±0.00**
4	98.59±1.95	**100.00±0.00**	**100.00±0.00**	97.79±3.16	**100.00±0.00**
5	99.43±1.62	98.63±2.21	98.63±2.21	98.40±2.24	**100.00±0.00**
6	99.06±1.30	**99.25±1.21**	96.13±3.89	98.54±2.42	94.91±0.25
7	99.67±0.94	**100.00±0.00**	98.67±1.41	98.53±1.36	**100.00±0.00**
8	**100.00±0.00**	**100.00±0.00**	95.81±2.68	99.00±1.25	**100.00±0.00**
9	**100.00±0.00**	**100.00±0.00**	99.27±2.32	99.45±1.70	**100.00±0.00**
10	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.92±0.20	**100.00±0.00**
11	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
12	**100.00±0.00**	99.83±0.09	98.12±2.92	99.90±0.11	**100.00±0.00**
13	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
OA	99.65±0.29	99.62±0.26	98.66±0.73	99.29±0.39	**99.79±0.01**
AA	99.58±0.22	99.57±0.34	98.37±0.67	99.08±0.49	**99.61±0.02**
Kappa	99.61±0.32	99.58±0.29	98.50±0.82	99.21±0.43	**99.77±0.01**

scenes. Comparatively, we find that MDGCN consistently performs better than the settings of $s=1$, $s=2$, and $s=3$ in terms of OA, AA, and Kappa coefficient, which indicates the usefulness of incorporating the multiscale spectral–spatial information into the graphs.

To show the effectiveness of dynamic graphs, Tables VII–IX also list the results acquired by only using multiscale graph convolution network (MGCN), where the graphs are fixed throughout the classification procedure. Compared with the results of MDGCN, there is a noticeable performance drop in the OA, AA, and Kappa coefficient of MGCN, which indicates that utilizing fixed graph convolution is not ideal for accurate classification. Therefore, the dynamically updated graph in our method is critical to rendering good classification results.

F. Classification Performance in the Boundary Region

One of the defects in traditional CNN-based methods is that the weights of each convolution kernel are identical when convolving all image patches, which may produce misclassifications in boundary regions. Different from the coarse convolution of traditional CNN-based methods with fixed size and weights, the graph convolution of the proposed MDGCN can be flexibly applied to irregular image patches and thus will not significantly "erase" the boundaries of objects during the convolution process. Therefore, the boundary information will be preserved and our MDGCN will perform better than the CNN-based methods in boundary regions. To reveal this advantage, in Fig. 10, we show the classification maps of a boundary region in the Indian Pines data set obtained by different methods, where the number of labeled pixels per class is kept identical to the abovementioned experiments in Section IV-C. The investigated boundary region is indicated by a black box in Fig. 10(a). Note that the results near the class boundaries are quite confusing and inaccurate in the classification maps of GCN, S^2GCN, R-2D-CNN, DR-CNN, MDA, HiFi, and JSDF, since the spatial information is very limited to distinguish the pixels around class boundaries. In contrast, the classification map of the proposed MDGCN [see Fig. 10(j)] is more compact and accurate than those of other methods.

TABLE X

RUNNING TIME COMPARISON (IN SECONDS) OF DIFFERENT METHODS. "IP" DENOTES THE INDIAN PINES DATA SET, "PAVIAU" DENOTES UNIVERSITY OF PAVIA DATA SET, AND "KSC" DENOTES THE KENNEDY SPACE CENTER DATA SET

Dataset	GCN [34]	S^2GCN [64]	R-2D-CNN [20]	DR-CNN [14]	MDGCN
IP	58	71	2156	2753	95
PaviaU	1783	1803	2272	3251	244
KSC	28	33	1470	2670	51

G. Running Time

Table X reports the running time of deep models, including GCN, S^2GCN, R-2D-CNN, DR-CNN, and the proposed MDGCN on the three data sets adopted earlier, where the number of labeled pixels per class is kept identical to the abovementioned experiments in Section IV-C. The codes for all methods are written in Python, and the running time is reported on a server with a 3.60-GHz Intel Xeon CPU with 264 GB of RAM and a Tesla P40 GPU. Although the time consumption of our proposed MDGCN is higher than GCN and S^2GCN on the Indian Pines and the Kennedy Space Center data sets, the classification performance of MDGCN is obviously better than those two methods. Moreover, on the large-scale data set (namely, the University of Pavia), our MDGCN requires much less time than GCN and S^2GCN to achieve satisfactory results. Due to the utilization of superpixel technique, the size of the graphs used in MDGCN can be significantly reduced. Consequently, the time consumption of the proposed MDGCN becomes quite low on large-scale data sets, even though MDGCN employs multiple graphs at different neighborhood scales.

V. CONCLUSION

In this article, we propose a novel MDGCN for hyperspectral image classification. Different from prior works that depend on a fixed input graph for convolution, the proposed MDGCN critically employs dynamic graphs that are gradually refined during the convolution process. Therefore, the graphs can faithfully encode the intrinsic similarities among the image regions and help to find accurate region representations. Meanwhile, multiple graphs with different neighborhood scales are constructed to fully exploit the multiscale information, which comprehensively discover the hidden spatial context carried by different scales. The experimental results on three widely used hyperspectral image data sets demonstrate that the proposed MDGCN is able to yield better performance when compared with the state-of-the-art methods.

REFERENCES

[1] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[2] P. Zhong, Z. Gong, and J. Shan, "Multiple instance learning for multiple diverse hyperspectral target characterizations," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[3] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[4] B.-C. Kuo, C.-S. Huang, C.-C. Hung, Y.-L. Liu, and I.-L. Chen, "Spatial information based support vector machine for hyperspectral image classification," in *Proc. IEEE IGARSS*, Jul. 2010, pp. 832–835.

[5] L. Shi, L. Zhang, J. Yang, L. Zhang, and P. Li, "Supervised graph embedding for polarimetric SAR image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 216–220, Mar. 2013.

[6] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[7] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[8] R. Hang, Q. Liu, H. Song, and Y. Sun, "Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 783–794, Feb. 2016.

[9] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018.

[10] Z. Zhong *et al.*, "Discriminant tensor spectral–spatial feature extraction for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 1028–1032, May 2015.

[11] Y. Wang, K.-F. Loe, T. Tan, and J.-K. Wu, "A dynamic hidden Markov random field model for foreground and shadow segmentation," in *Proc. WACV*, vol. 1, Jan. 2005, pp. 474–480.

[12] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.

[13] B. Song *et al.*, "Remotely sensed image classification using sparse representations of morphological attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 5122–5136, Aug. 2014.

[14] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE ICCV*, Dec. 2015, pp. 3730–3738.

[16] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

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

[18] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[19] 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.

[20] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[21] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[22] L. Fang, Z. Liu, and W. Song, "Deep hashing neural networks for hyperspectral image feature extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1412–1416, Sep. 2019.

[23] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[24] X. Ma, J. Geng, and H. Wang, "Hyperspectral image classification via contextual deep learning," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, p. 20, Dec. 2015.

[25] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[26] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, Apr. 2018.

[27] 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.

[28] P. Jia, M. Zhang, W. Yu, F. Shen, and Y. Shen, "Convolutional neural network based classification for hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 5075–5078.

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

[30] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE IGARSS*, Jul. 2015, pp. 4959–4962.

[31] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, 2017.

[32] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[33] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–14.

[35] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE ICIP*, Sep. 2017, pp. 3904–3908.

[36] X. J. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1530, 2005.

[37] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.

[38] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[39] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[40] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 1, Jul. 2003, pp. 288–290.

[41] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[42] L. He and X. Chen, "A three-dimensional filtering method for spectral-spatial hyperspectral image classification," in *Proc. IEEE IGARSS*, Jul. 2016, pp. 2746–2748.

[43] L. He, J. Li, A. Plaza, and Y. Li, "Discriminative low-rank Gabor filtering for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1381–1395, Mar. 2017.

[44] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[45] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[46] G. Zhang and X. Jia, "Simplified conditional random fields with class boundary constraint for spectral-spatial based remote sensing image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 5, pp. 856–860, Sep. 2012.

[47] Y. Zhong, X. Lin, and L. Zhang, "A support vector conditional random fields classifier with a Mahalanobis distance boundary constraint for high spatial resolution remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 7, no. 4, pp. 1314–1330, Apr. 2014.

[48] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo, "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3368–3379, 2015.

[49] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[50] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE ICIP*, Oct. 2014, pp. 5132–5136.
[51] C. Shi and C.-M. Pun, "Multi-scale hierarchical recurrent neural networks for hyperspectral image classification," *Neurocomputing*, vol. 294, pp. 82–93, Jun. 2018.
[52] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.
[53] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *Proc. IEEE IGARSS*, Jul. 2016, pp. 5079–5082.
[54] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral–spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 6, pp. 468–477, May 2015.
[55] 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.
[56] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2005, pp. 729–734.
[57] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational Capabilities of Graph Neural Networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 81–102, Jan. 2009.
[58] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*. [Online]. Available: https://arxiv.org/abs/1312.6203
[59] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *Proc. ICML*, 2016, pp. 2702–2711.
[60] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3697–3707.
[61] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[62] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. ACM SIGKDD*, 2018, pp. 974–983.
[63] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for RGBD semantic segmentation," in *Proc. IEEE ICCV*, Oct. 2017, pp. 5199–5208.
[64] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.
[65] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[66] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units," in *Proc. IJCNN*, Jul. 2015, pp. 1–4.
[67] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
[68] B. Wang, Z. Tu, and J. K. Tsotsos, "Dynamic label propagation for semi-supervised multi-class multi-label classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 425–432.
[69] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2997–3004.
[70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[71] S. Zhang and S. Li, "Spectral-spatial classification of hyperspectral images via multiscale superpixels based sparse representation," in *Proc. IEEE IGARSS*, Jul. 2016, pp. 2423–2426.
[72] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proc. SIGKDD KDD*, 2018, pp. 1416–1424.
[73] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4177–4189, Jul. 2017.
[74] C. Bo, H. Lu, and D. Wang, "Hyperspectral image classification via JCR and SVM models with decision fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 177–181, Feb. 2016.

Sheng Wan received the B.S. degree from Nanjing Agricultural University (NJAU), Nanjing, China, in 2016. He is currently pursuing the Ph.D. degree with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, and the School of Computer Science and Engineering, Nanjing University of Science and Technology (NJUST), Nanjing, under the supervision of Dr. C. Gong.

His research interests include deep learning and hyperspectral image processing.

Chen Gong (M'16) received the B.E. degree from the East China University of Science and Technology (ECUST), Shanghai, China, in 2010, and the dual Ph.D. degrees from Shanghai Jiao Tong University (SJTU), Shanghai, and the University of Technology Sydney (UTS), Ultimo, NSW, Australia, in 2016 and 2017, respectively.

He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has published over 70 technical articles at prominent journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON CYBERNETICS (TCYB), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), the IEEE TRANSACTIONS ON MULTIMEDIA (TMM), the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), the ACM TIST, NeurIPS, CVPR, the Association for the Advance of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), and ICDM. His research interests include machine learning, data mining, and learning-based vision problems.

Dr. Gong serves as a SPC/PC member for several top-tier conferences, such as ICML, NeurIPS, AAAI, IJCAI, ICDM, and AISTATS. He was a recipient of the Excellent Doctoral Dissertation Award by Shanghai Jiao Tong University (SJTU) and the Chinese Association for Artificial Intelligence (CAAI). He was also enrolled by the Young Elite Scientists Sponsorship Program of Jiangsu Province and the China Association for Science and Technology. He also serves as a Reviewer for over 20 international journals, such as AIJ, IEEE TPAMI, IEEE TNNLS, and IEEE TIP.

Ping Zhong (M'09–SM'18) received the M.S. degree in applied mathematics and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2003 and 2008, respectively.

From 2015 to 2016, he was a Visiting Scholar with the Department of Applied Mathematics and Theory Physics, University of Cambridge, Cambridge, U.K. He is currently a Professor with the ATR National Laboratory, NUDT. He has authored over 30 peer-reviewed articles in international journals such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (JSTSP), and the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). His research interests include computer vision, machine learning, and pattern recognition.

Dr. Zhong was a recipient of the National Excellent Doctoral Dissertation Award of China in 2011 and the New Century Excellent Talents in University of China in 2013. He is a Referee of the IEEE TNNLS, IEEE TIP, IEEE TGRS, IEEE JSTARS, IEEE JSTSP, and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL).

Bo Du (M'10–SM'15) received the B.S. and Ph.D. degrees in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China, in 2005 and 2010, respectively.

He was with the Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Ultimo, NSW, Australia. He is currently a Professor with the School of Computer, Wuhan University. He has authored over 40 research articles in the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL). His research interests include pattern recognition, hyperspectral image processing, and signal processing.

Dr. Du was a recipient of the Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society (GRSS) for his services to the IEEE JSTARS in 2011 and the ACM Rising Star Award for his academic progress in 2015. He was the Session Chair of the IEEE IGARSS 2016 and the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing. He also serves as a Reviewer for 20 Science Citation Index magazines, including the IEEE TGRS, IEEE TIP, IEEE JSTARS, and the IEEE GRSL.

Lefei Zhang (S'11–M'14) received the B.S. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

He was a Big Data Institute Visitor with the Department of Statistical Science, University College London, London, U.K., in 2016, and a Hong Kong Scholar with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2017. He is currently a Professor with the School of Computer Science, Wuhan University. His research interests include pattern recognition, image processing, and remote sensing.

Dr. Zhang is a Reviewer/PC member of numerous journals/conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), the Association for the Advance of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), and International Conference on Knowledge Discovery and Data Mining (KDD).

Jian Yang (M'06) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

In 2003, he was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a Post-Doctoral Fellow with the Biometrics Center, Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, NY, USA. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He has authored over 200 scientific articles in pattern recognition and computer vision. His articles have been cited over 6000 times in the Web of Science, and 17 000 times in the Scholar Google. His research interests include pattern recognition, computer vision, and machine learning.

Dr. Yang is a fellow of IAPR. He is/was an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*.