Dual Interactive Graph Convolutional Networks for Hyperspectral Image Classification

Sheng Wan, Shirui Pan, *Member, IEEE*, Ping Zhong, *Senior Member, IEEE*, Xiaojun Chang, *Member, IEEE*, Jian Yang, *Member, IEEE*, and Chen Gong, *Member, IEEE*,

***Abstract*—Recently, Graph Convolutional Network (GCN) has progressed significantly and gained increasing attention in hyperspectral image (HSI) classification due to its impressive representation power. However, existing GCN-based methods do not give full consideration to the multi-scale spatial information, since the convolution operations are governed by fixed neighborhood. As a result, their performances can be limited, particularly in the regions with diverse land cover appearances. In this paper, we develop a new Dual Interactive GCN (DIGCN) which introduces the dual GCN branches to capture spatial information at different scales. More significantly, the dual interactive module is embedded across the GCN branches, so that the correlation of multi-scale spatial information can be leveraged to refine the graph information. To be concrete, the edge information contained in one GCN branch can be refined by incorporating the feature representations from the other branch. Analogously, improved feature representations can be generated in one GCN branch by fusing the edge information from the other branch. As such, the refined graph information can help enhance the representation power of the model. Furthermore, to avoid the negative effects of the manually constructed graph, our proposed model adaptively learns a discriminative region-induced graph, which also accelerates the convolution operation. We comprehensively evaluate the proposed method on four commonly used HSI benchmark datasets, and state-of-the-art results can be achieved when compared with several typical HSI classification methods.**

***Index Terms*—Hyperspectral image classification, graph convolutional network (GCN), deep learning, graph refinement.**

I. Introduction

HYPERSPECTRAL remote sensors capture digital images from hundreds of consecutive geographic object segments, thereby providing detailed spectral information and revealing enhanced ability to distinguish geographic objects [1]. Over the past few decades, hyperspectral image (HSI) classification has been an active topic in various fields, such as military target detection and vegetation monitoring [2], and it aims to categorize each image pixel into a certain meaningful class according to the image contents.

During the past few decades, diverse kinds of methods have been proposed for HSI classification. Among them, spectral-spatial methods have gained some success [2]. One kind of these approaches extracts spectral-spatial information before learning a classifier, such as the patch-based feature extraction technique [3]. Another popular research trend incorporates spatial information with Markov Random Field (MRF)-based models to post-process the classification map. Other spatial-based methods include morphological profiles [4], and manifold learning methods [5].

Inspired by the success of deep learning models, HSI classification with deep networks, such as Convolutional Neural Network (CNN), have been extensively investigated [6], [7]. For instance, Zhang *et al.* [6] developed a deep CNN model based on diverse region types, where different local or global region inputs are utilized to jointly learn the pixel representations. Following the recent developments of 3D CNN for video analysis [8], Chen *et al.* [9] introduced an l_2 regularized 3D CNN for learning spectral-spatial features. While powerful, the CNN-based methods still suffer from some limitations when dealing with HSI. Specifically, CNN fails to precisely perceive geometric variations between different object regions, since it only conducts convolution on squared image grids and cannot apply to arbitrarily shaped regions. Apart from this, the convolution kernels constrained with fixed shapes and weights cannot be well adaptive to the object regions with diverse sizes or shapes.

To better handle the irregular class boundaries, the recently proposed Graph Convolutional Network (GCN) [10] has been adopted for HSI classification. Different from CNN, GCN operates on graph structured data, such as social network data [11], and is able to pass, transform, and aggregate feature information across the graph [12]. In this way, GCN can be naturally applied to non-Euclidean data based on the predefined graph, which helps preserve the class boundaries of different object regions flexibly. Most recently, a handful of works on GCN-based HSI classification have emerged and achieved promising performance [13], [14], [15], [16], [17].

This work was supported by NSF of China (No: 61973162, U1713208, 61971428, 61671456), the Fundamental Research Funds for the Central Universities (No: 30920032202), the "Young Elite Scientists Sponsorship Program" by CAST (No: 2018QNRC001), Guangdong Key Area Research Project (No: 2018B010108003), the Program for Changjiang Scholars, and the "111" Program (No: B13022). (*Corresponding author: Chen Gong, Shirui Pan.*)

S. Wan and J. Yang are 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, Nanjing, 210094, P.R. China (e-mail: wansheng315@hotmail.com; csjyang@njust.edu.cn).

S. Pan and X. Chang are with Department of Data Science and AI, Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (Email: shirui.pan@monash.edu; cxj273@gmail.com).

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).

C. Gong is with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, and is also with the Department of Computing, Hong Kong Polytechnic University (e-mail: chen.gong@njust.edu.cn).

For instance, Qin *et al.* [13] devised a spectral-spatial GCN which makes full use of spatial information via using adjacent pixels to approximate convolution. However, the research on GCN-based HSI classification is still in its primary stage, especially on the way to characterize the spatial information. To be concrete, the existing GCN models usually perform graph convolution based on fixed single neighborhood, thereby failing to exploit the multi-scale cues in HSI. As a consequence, the representation power of these single-input style GCN models become limited, especially in the regions with various land cover appearances.

Most recently, in [15], the Multi-scale Dynamic GCN (MDGCN) is proposed to incorporate the spatial information at different scales. However, the graph convolution is performed separately at different spatial scales, and thus is limited in the fixed neighborhood. Different from MDGCN, we introduce a new model called Dual Interactive GCN (DIGCN), where the interaction of multi-scale spatial information can be leveraged to refine the input graph and help guide the graph convolution operation. To be specific, in our dual interactive module, the edge information contained in one GCN branch can be refined with the feature representations from the other branch. Analogously, the representations produced by one GCN branch can also be improved by fusing the edge information from the other branch. Owing to the dual interaction between the GCN branches, feature representations inheriting the advantages of multi-scale spatial information can be obtained, which helps better represent the regions with diverse appearances.

To perform the dual interactive graph convolution properly, one crucial step is to construct an appropriate graph. Although constructing a graph based on image pixels is an intuitive solution, some issues may ensue with such approaches. Firstly, the graph may not be optimal for the convolution operation, as it is constructed independently from the graph convolution step. Secondly, the manually constructed graphs are sensitive to the noise contained in pixel features. Lastly, the huge number of image pixels may introduce intractable computational problems, especially when performing multi-scale graph convolution. In this paper, our proposed DIGCN adaptively learns to project the original HSI into a region-induced graph, by which the graph size gets significantly reduced and the noise of individual pixels is also suppressed [18]. Moreover, a discriminative loss function is embedded into the graph learning process, which aims to enhance the representation power of the projected features.

To sum up, the main contributions of our proposed DIGCN are as follows: 1) the interaction of multi-scale spatial information is leveraged to help refine the graph convolution operation, by which the regions with diverse appearances can be better represented; 2) by learning a discriminative region-induced graph, the representation power of the generated region features can be further enhanced and the computational cost is significantly reduced. Extensive experiments on four real-world datasets demonstrate the effectiveness of the proposed DIGCN.

II. Related Work

In this section, we review some typical works on HSI classification and GCN, as they are related to this work.

A. Hyperspectral Image Classification

At the beginning, research efforts are only concentrated on spectral-based HSI classification, which ignores the critical spatial information and thereby results in poor classification performance [19]. To mitigate this limitation, spatial context, which has been observed to be arguably more effective than spectral information, is introduced for HSI classification.

Structural filtering is one of the typical spectral-spatial-based HSI classification methods, which has been widely applied to generate spatial features for HSI preprocessing [20]. A simple and powerful practice is to extract spatial information via moment criteria, such as the mean or standard deviation of adjacent pixels in a window [21]. In structural filtering, local harmonic analysis is also a research direction in addition to moment criteria, which includes spatial translation-invariant spectral-spatial wavelet features, spectral-spatial Gabor features, and empirical mode decomposition-based features [22].

Mathematical morphology is an effective tool for analyzing and processing geometrical structures in spatial domain. For example, the Morphological Profile (MP) is introduced for classifying images with very high spatial resolutions via using a set of geodesic opening and closing operations [23]. Furthermore, Extended MP (EMP) is developed by Benediktsson *et al.* [4], where an MP is calculated on each component after performing dimensionality reduction. To better exploit the spectral information contained in HSI, Fauvel *et al.* [24] proposed an approach to fuse the spectral and spatial information based on EMP and the original data.

Furthermore, a handful of recently proposed studies [25], [26] have also achieved promising results on HSI classification. For instance, Cao *et al.* [25] devised a powerful semi-supervised wrapper method with the local smoothness of HSI, which is also robust to label noise. Meanwhile, in [26], an effective minority class oversampling technique is devised using generative adversarial learning and aims to handle the class imbalance problem of HSI classification. Additionally, a series of image classification models have also shown their potentials in HSI classification, such as the Gabor convolutional networks [27] and the deep confidence network [28].

B. Graph Convolutional Network

As one of the powerful techniques for data representation, graph based methods [29], [30], [31], [32] have been extensively investigated for HSI classification. In [33], Camps-Valls *et al.* first proposed a semi-supervised graph-based method for HSI classification, where the graphs are constructed based on different spectral and spatial kernels. After that, Gao *et al.* [34] devised a bilayer graph-based learning method for HSI classification. Meanwhile, Cui *et al.* [35] employed an extended random walker on a superpixel-induced graph to optimize the classification map.

In recent years, there has been a surge of research interest in applying convolutions on graphs [36], [37], which can be roughly divided into spatial and spectral methods. In spatial methods, a weighted average function is utilized to perform convolution over the neighbors of each graph node. For example, the weighting function is built via using various aggregators over the neighboring nodes in GraphSAGE [11]. Different from spatial methods, the spectral methods employ spectral graph theories to define parameterized filters. Spectral CNN, which is the pioneering work of spectral methods, converts the signals defined in node domain into spectral domain by leveraging graph Fourier transform. However, the computational complexity gets extremely high on large-scale graphs due to the eigendecomposition of Laplacian matrix. Afterwards, in [38], the convolution kernel is considered as a polynomial function of the diagonal matrix containing the eigenvalues of Laplacian matrix. Subsequently, Kipf and Welling proposed GCN via using a localized first-order approximation to ChebyNet [10], which contributes to more efficient filtering operation than that in spectral CNN.

During the past few years, GCN has achieved remarkable success in various fields, such as social network mining [39] and natural language processing [40]. For instance, in [41], GCN is integrated with MRF to solve the problem of semi-supervised community detection in attributed networks. Inspired by this great success, GCN has also been studied for processing 2D images, such as HSI [13], [14], [15], [16]. For example, Mou *et al.* [14] built a dense fully-connected graph based on all image pixels for convolution, in order to obtain a large receptive field. In addition, Hong *et al.* [16] proposed a mini-batch GCN to allow the network training on large-scale hyperspectral data in a mini-batch fashion. However, there still exists a potential limitation in these early-staged methods. To be specific, they cannot sufficiently exploit the diverse types of spatial information at different scales. In this paper, we introduce a novel dual interactive GCN model that takes advantages of the interaction of multi-scale spatial information. As such, the representation power of the GCN model can be enhanced, especially in the regions with diverse land cover appearances.

III. Problem Definition

We start by formally introducing the problem of HSI classification. Suppose that we have a HSI composed of $n = p + q$ pixels $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_p, \mathbf{z}_{p+1}, \cdots, \mathbf{z}_n\}$, where the labels (i.e., land-cover types) of the first p pixels are available and the remaining q pixels constitute the unlabeled set. Here $\mathbf{z}_i$ is characterized by a d-dimensional vector of features which denote the measurement in spectral bands acquired by the sensors. We also denote $\mathbf{Y} \in \mathbb{R}^{n \times C}$ as the label matrix with its $(i,j)^{\text{th}}$ element $\mathbf{Y}_{ij} = 1$ if $\mathbf{z}_i$ belongs to the j^{th} class and $\mathbf{Y}_{ij} = 0$ otherwise, where C is the number of land-cover types. The purpose of HSI classification task is to infer the class labels of the unlabeled pixels $\mathbf{z}_{p+1}, \mathbf{z}_{p+2}, \cdots, \mathbf{z}_n$ based on the feature values of all pixels and the labels of $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_p$.

IV. The Proposed Method

This section details our proposed DIGCN algorithm, of which the architecture is illustrated in Fig. 1. When an input HSI (Fig. 1(a)) is given, we first learn to project the original 2D grids into image regions (Fig. 1(b)). Afterwards, dual interactive graph convolution (Fig. 1(c) and Fig. 1(d)) is performed to acquire representative features through information integration and propagation. Finally, the classification result (Fig. 1(e)) is generated via interpolating the learned graph feature back into 2D image grids based on the region-to-pixel assignment. Next the critical steps will be detailed by elaborating the graph projection operation (Section IV-A), explaining the GCN formulation (Section IV-B), describing the dual interactive architecture (Section IV-C), and presenting the graph re-projection step (Section IV-D).

A. Discriminative Pixel-to-Region Assignment

For GCN, an accurate graph plays a critical role in obtaining expressive representations. In some applications such as chemical molecules and social networks, one can directly use the existing graph from domain knowledge as network input. In many other applications where the graph structure is not available, one popular way is to construct a graph (e.g., nearest-neighbor graph) in advance [43]. However, the graphs obtained from domain knowledge or constructed in advance may not be optimal for GCN, as they are generally independent of the graph convolution process [44]. Besides, the manually constructed graphs are sensitive to local noise and outliers. To overcome these problems, [44] proposed to learn an optimal graph that best serves the GCN, where both given labels and the estimated labels are facilitated to refine the graph construction operation. Apart from this, in [45], the graph structure and features are explored to help guide the robust graph learning. Inspired by these techniques, we intend to adaptively learn the graph information by the network. Nevertheless, directly performing graph convolution on the huge number of pixels will lead to extremely high computational complexity. Fortunately, region-based GCN [15] has been demonstrated to be effective in reducing the computational cost. Therefore, we aim at learning a region-induced graph transformed from the original HSI, which is termed graph projection.

First, a soft assignment matrix parameterized by $\mathbf{V} \in \mathbb{R}^{d \times c}$ can be learned to assign each pixel $\mathbf{z}_i \in \mathbb{R}^d$ to its neighboring regions, where d is the feature dimensionality of each pixel, c denotes the number of image regions, and each column $\mathbf{v}_i \in \mathbb{R}^d$ of $\mathbf{V}$ corresponds to the anchor point of a certain region. Specifically, the soft assignment matrix $\mathbf{P} \in \mathbb{R}^{n \times c}$ (n denotes the number of image pixels) can be computed as

$$\mathbf{P}_{ij} = \begin{cases} e^{-\gamma \|\mathbf{z}_i - \mathbf{v}_j\|^2} & \text{if } \mathbf{v}_j \in \text{Anc}(\widetilde{N}(\mathbf{z}_i)) \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where γ is a temperature parameter and has been empirically set to 0.2 throughout the paper, $\widetilde{N}(\mathbf{z}_i)$ represents the set of neighboring regions that are connected to the pixel $\mathbf{z}_i$, and $\text{Anc}(\widetilde{N}(\mathbf{z}_i))$ indicates the anchor points of $\widetilde{N}(\mathbf{z}_i)$. More concretely, not only the central region where $\mathbf{z}_i$ resides, but

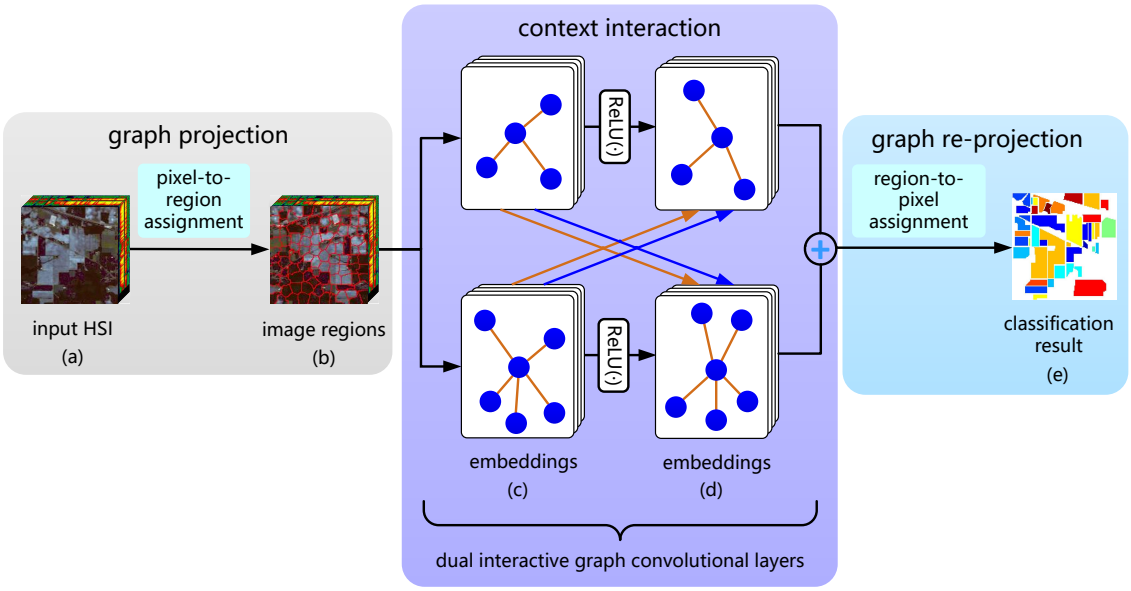


Fig. 1. The framework of our algorithm. (a) is the original HSI. (b) shows the regions obtained via discriminative pixel-to-region assignment. (c) and (d) denote the dual interactive graph convolutional layers, where the blue circles and orange lines represent the graph nodes and edges, respectively. Besides, the blue and orange arrows indicate the interaction of graph nodes and edges, respectively. Specifically, the two GCN branches characterize spatial information at different scales and are forced to perform dual interactive graph convolution. Here ReLU [42] is used as the activation function. In (e), the region features are interpolated back into 2D image grids after dual interactive graph convolution, by which the classification results can be acquired.

also the regions adjacent to the central one are included in $\widetilde{N}(\mathbf{z}_i)$, and then the region feature $\mathbf{x}_j$ can be encoded as

$$\mathbf{x}_j = \frac{\sum_i \mathbf{P}_{ij}\mathbf{z}_i}{\sum_i \mathbf{P}_{ij}}. \tag{2}$$

This assignment is soft and likely to group pixels with similar features to the same region, and thus improved region features can be adaptively learned by the network.

To enhance the discriminative power of the learned region features, we regularize the graph learning process with a label consistency loss. The intuition behind is that data from the same class often have similar features and those from different classes usually have different features. To achieve this, we first design an ideal discriminative adjacency matrix $\mathbf{Q}$ in advance, namely

$$\mathbf{Q}_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ share the same label} \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Then the discriminative loss function can be defined as

$$\mathcal{L}_{dis} = \left\| \mathbf{S} \otimes (\mathbf{Q} - \mathbf{A}^{(1)}) \right\|, \tag{4}$$

where $\otimes$ means element-wise multiplication, $\mathbf{S}$ is a 0-1 sparse matrix with the element $\mathbf{S}_{ij}$ equaling 1 if the supervised information of $\mathbf{x}_i$ and $\mathbf{x}_j$ is available during training, and $\mathbf{A}^{(1)}$ is the adjacency matrix calculated based on the region features $\mathbf{X}$. Here $\mathbf{x}_i$ is the i^{th} row of $\mathbf{X}$, and the calculation of $\mathbf{A}^{(1)}$ will be introduced in Eq. (5). Besides, the majority voting rule [46] is adopted to determine the region labels. Specifically, the most frequent pixel class within a region is considered as the corresponding region label. With Eqs. (3) and (4), the scarce but valuable label information contained in $\mathbf{Q}$ can help regularize the learning of region features and further enhance the discriminative power of the adjacency matrix $\mathbf{A}^{(1)}$.

However, the graph projection step still faces an optimization challenge that most or even all of the image pixels might be assigned to a single region in some extreme circumstances. This is probably due to the improper initialization of the anchor point matrix $\mathbf{V}$, which will subsequently result in an ill-posed assignment matrix $\mathbf{P}$. Moreover, the imbalanced assignment will lead to unfavorable graph structure, thereby weakening the expressive power of the generated representations. To avoid this problem, we take advantage of the spatial information by initializing $\mathbf{V}$ via a segmentation technique, instead of resorting to random initialization. Here the Simple Linear Iterative Clustering (SLIC) algorithm [47] is employed to obtain the initial anchor point matrix $\mathbf{V}$. Specifically, the average spectral signatures of the pixels involved in the corresponding region will be utilized to initialize each $\mathbf{v}_i$. Note that the matrix $\mathbf{V}$ comprising anchor points of the regions can be further updated via using gradient descent. Afterwards, the soft assignment matrix $\mathbf{P}$ can be calculated via Eq. (1), and then the region features can be naturally obtained with Eq. (2). This segmentation-based initialization technique can yield more stable training performance and produce more meaningful graph representations than random initialization [48]. Fig. 2 exhibits the pixel-to-region assignment regarding a pixel $\mathbf{z}_i$. By learning the discriminative region-induced graph, the negative impact of inaccurate pre-computed region features

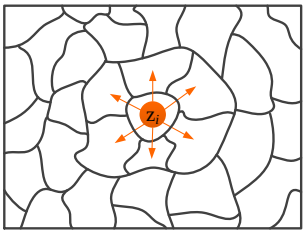


Fig. 2. Illustration of the soft pixel-to-region assignment used in our DIGCN algorithm. Each of the initialized image regions is surrounded by black lines, and the orange arrows denote the assignment regarding the pixel $\mathbf{z}_i$ to its neighboring regions.

can be reduced and expressive region features can be flexibly obtained.

B. Formulation of Graph Convolutional Network

GCN [10] is a multi-layer neural network inspired by CNN, which can operate on graph-structured data and extract high-level features via aggregating feature information from the neighborhood of each node. Formally, an undirected graph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ and $\mathcal{E}$ denoting the sets of nodes and edges, respectively. The notation $\mathbf{A}$ represents the adjacency matrix of the graph $\mathcal{G}$ and indicates the existence of an edge between each pair of nodes, which can be calculated as

$$\mathbf{A}_{ij} = \begin{cases} e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2} & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where γ is a temperature parameter and has been empirically set to 0.2 throughout the paper, $\mathbf{x}_i$ and $\mathbf{x}_j$ are two graph nodes (i.e., image regions in this paper), $\|\mathbf{x}_i - \mathbf{x}_j\|$ calculates the Euclidean distance between each pair of neighboring regions, and $N(\mathbf{x}_j)$ is the set of neighbors of $\mathbf{x}_j$. In our proposed DIGCN, $N(\mathbf{x}_j)$ is composed of the spatial neighbors of $\mathbf{x}_j$, which will be illustrated in Section IV-C.

First, to conduct node embedding for $\mathcal{G}$, spectral filtering on the graph, which can be expressed as the multiplication of a signal $\mathbf{x}$ with a filter $g_{\boldsymbol{\theta}} = \mathrm{diag}(\boldsymbol{\theta})$ in the Fourier domain, can be defined as

$$g_{\boldsymbol{\theta}} \star \mathbf{x} = \mathbf{U} g_{\boldsymbol{\theta}} \mathbf{U}^\top \mathbf{x}, \tag{6}$$

where $\mathbf{U}$ denotes the matrix of eigenvectors of normalized graph Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$. Here the diagonal matrix $\boldsymbol{\Lambda}$ is composed of the eigenvalues of $\mathbf{L}$, $\mathbf{D}$ is the degree matrix with each diagonal element $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, and $\mathbf{I}$ is the identity matrix with proper size throughout this paper. Then $g_{\boldsymbol{\theta}}$ can be understood as a function of the eigenvalues of $\mathbf{L}$, i.e., $g_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$. To reduce the computational complexity of eigen-decomposition in Eq. (6), Hammond *et al.* [49] approximated $g_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$ via using a truncated expansion in terms of Chebyshev polynomials $T_k(\mathbf{x})$ up to K^{th}-order, namely

$$g_{\boldsymbol{\theta}'}(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k(\widetilde{\boldsymbol{\Lambda}}), \tag{7}$$

where $\boldsymbol{\theta}'$ denotes a vector of Chebyshev coefficients; $\widetilde{\boldsymbol{\Lambda}} = \frac{2}{\lambda_{\max}}\boldsymbol{\Lambda} - \mathbf{I}$ with $\lambda_{\max}$ being the largest eigenvalue of $\mathbf{L}$. According to [49], the Chebyshev polynomials can be defined as $T_k(\mathbf{x}) = 2\mathbf{x}T_{k-1}(\mathbf{x}) - T_{k-2}(\mathbf{x})$, where $T_0(\mathbf{x}) = 1$ and $T_1(\mathbf{x}) = \mathbf{x}$. Afterwards, the convolution of a signal $\mathbf{x}$ can be defined as

$$g_{\boldsymbol{\theta}'} \star \mathbf{x} \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k(\widetilde{\mathbf{L}})\mathbf{x}, \tag{8}$$

where $\widetilde{\mathbf{L}} = \frac{2}{\lambda_{\max}}\mathbf{L} - \mathbf{I}$ is the scaled Laplacian matrix. Eq. (8) can be easily verified by considering the fact that $(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top)^k = \mathbf{U}\boldsymbol{\Lambda}^k\mathbf{U}^\top$. This expression is a K^{th}-order polynomial with respect to the Laplacian (i.e., K-localized). In our proposed DIGCN, only the first-order neighborhood is considered, i.e., $K = 1$, and thus Eq. (8) turns to a linear function on the graph Laplacian spectrum with respect to $\mathbf{L}$.

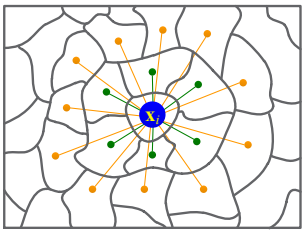


Fig. 3. The illustration of neighborhood sizes utilized in our method. The green nodes denote the 1-hop neighbors of the region $\mathbf{x}_i$, and the orange nodes together with the green nodes represent $\mathbf{x}_i$'s 2-hop neighbors.

Afterwards, a neural network based on graph convolution can be constructed by stacking multiple convolutional layers in the form of Eq. (8), where each layer is followed by an element-wise non-linear operation (i.e., ReLU$(\cdot)$ [42]). By this way, diverse classes of convolutional filter functions can be derived through stacking multiple layers with the same configuration. With the linear formulation, Kipf and Welling [10] further approximated $\lambda_{\max} \approx 2$, considering that the network parameters can adapt to this change in scale during training. Consequently, Eq. (8) is simplified to

$$g_{\boldsymbol{\theta}'} \star \mathbf{x} \approx \boldsymbol{\theta}'_0\mathbf{x} + \boldsymbol{\theta}'_1(\mathbf{L} - \mathbf{I})\mathbf{x} = \boldsymbol{\theta}'_0\mathbf{x} - \boldsymbol{\theta}'_1\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x}, \tag{9}$$

where $\boldsymbol{\theta}'_0$ and $\boldsymbol{\theta}'_1$ are two free parameters. Since reducing the number of parameters helps avoid overfitting, Eq. (9) is further converted to $g_{\boldsymbol{\theta}} \star \mathbf{x} \approx \boldsymbol{\theta}(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{x}$ by letting $\boldsymbol{\theta} = \boldsymbol{\theta}'_0 = -\boldsymbol{\theta}'_1$. As $\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ has the eigenvalues in the range $[0, 2]$, repeatedly applying this operator will result in numerical instabilities and exploding/vanishing gradients in a deep network. To solve this deficiency, Kipf and Welling [10] performed the re-normalization trick $\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \rightarrow \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$ with $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$.Hence the convolution operation can be expressed as

$$\mathbf{H}^{(l)} = \sigma(\widetilde{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}), \tag{10}$$

where $\mathbf{H}^{(l)}$ denotes the output of the l^{th} layer with $\mathbf{H}^{(0)} = \mathbf{X}$, $\sigma(\cdot)$ represents an activation function, such as the ReLU function [42] used in our proposed DIGCN, and $\mathbf{W}^{(l)}$ is the trainable weight matrix involved in the l^{th} layer.

C. Dual Interactive Architecture

As mentioned in the introduction, there often exist various object appearances in HSI, where the object regions from the same class even have diverse sizes and shapes. Conventional GCN models cannot well adapt to this difficult situation, since they fail to extract the contextual information within different spatial scales. Concretely, the convolution operation of GCN models is usually governed by fixed neighborhood. Although stacking more layers can incorporate contextual information from larger spatial scales, a deep GCN model is very likely to cause over-smoothing problem [50].

In this paper, we develop a dual interactive graph convolution module, where the interaction of multi-scale spatial information can be leveraged to help better represent the regions with diverse appearances. Considering that GCN is

able to aggregate information within the corresponding neighborhood, graphs constructed with different neighborhood sizes are adopted in the two GCN branches, in order to characterize spatial information at different scales. Here, the s-hop neighbors of each region are considered to construct the graph with neighborhood size s. To better describe the definition of neighborhood size, Fig. 3 is used to exhibit the 1-hop and 2-hop neighbors of a central region $\mathbf{x}_i$. Note that an oversized neighborhood size is likely to incorporate ineffective spatial information and result in heavy computational cost, and thus it is essential to choose the reasonable neighborhood sizes for graph construction, which will be further discussed in the experiments. By aggregating the spatial information based on different neighborhood sizes, feature representations $\mathbf{H}_1^{(l)}$ and $\mathbf{H}_2^{(l)}$ can be generated. Here $\mathbf{H}_1^{(l)}$ and $\mathbf{H}_2^{(l)}$ denote the outputs of the l^{th} graph convolutional layer in two branches, respectively.

To leverage the interaction of multi-scale spatial information, the dual interactive architecture is applied across the GCN branches, as illustrated in Fig. 4. Here, the graph information in each GCN branch can be helpful to refine the convolution operation of the other branch. For example, in branch 1, the adjacency matrix of the l^{th} graph convolutional layer $\mathbf{A}_1^{(l)}$ receives the information from branch 2 as follows:

$$\left[\mathbf{A}_1^{(l)}\right]_{ij} \leftarrow \left[\mathbf{A}_1^{(l)}\right]_{ij} + \beta_1 e^{-\gamma\left\|\left[\mathbf{H}_2^{(l-1)}\right]_{i,:}-\left[\mathbf{H}_2^{(l-1)}\right]_{j,:}\right\|^2}, \quad (11)$$

where $\mathbf{H}_2^{(l-1)}$ denotes the representations generated from the $(l-1)^{\text{th}}$ layer in branch 2, β_1 is the weight assigned to the information from branch 2 and can be learned via gradient descent. Similar to Eq. (11), the adjacency matrix $\mathbf{A}_2^{(l)}$ in branch 2 can be updated by

$$\left[\mathbf{A}_2^{(l)}\right]_{ij} \leftarrow \left[\mathbf{A}_2^{(l)}\right]_{ij} + \beta_2 e^{-\gamma\left\|\left[\mathbf{H}_1^{(l-1)}\right]_{i,:}-\left[\mathbf{H}_1^{(l-1)}\right]_{j,:}\right\|^2}. \quad (12)$$

With Eqs. (11) and (12), the similarities among graph nodes serve as the supplement to the edge information and help refine the graph structure. In the meanwhile, the node representations can also get updated by adopting information from a different spatial scale, namely

$$\left[\mathbf{H}_1^{(l)}\right]_{i,:} \leftarrow \text{concat}\left(\left[\mathbf{H}_1^{(l)}\right]_{i,:}, \text{MaxPool}\left(\left[\mathbf{A}_2^{(l)}\right]_{i,:}\right)\right), \quad (13)$$

$$\left[\mathbf{H}_2^{(l)}\right]_{i,:} \leftarrow \text{concat}\left(\left[\mathbf{H}_2^{(l)}\right]_{i,:}, \text{MaxPool}\left(\left[\mathbf{A}_1^{(l)}\right]_{i,:}\right)\right), \quad (14)$$

where concat$(\cdot,\ \cdot)$ is the concatenation of two vectors, MaxPool$(\cdot)$ represents the max-pooling operation, and $\mathbf{H}_1^{(l)}$ denotes the representations generated from the l^{th} layer in branch 1. In Fig. 4, the blue arrows correspond to the interaction in Eqs. (11)-(12), and the orange arrows correspond to the interaction in Eqs. (13)-(14), respectively. With this dual interactive module, spatial information at different scales can be sufficiently exploited to help guide the graph convolution operation and enhance the representation power of the GCN model.

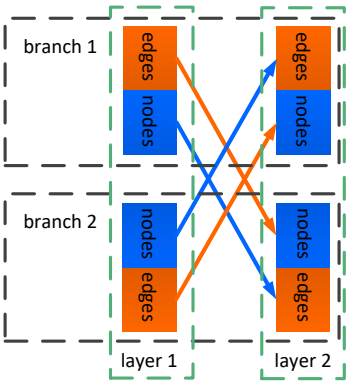


Fig. 4. The interactive graph convolution layers used in our method. The graphs in different branches are constructed based on different neighborhood sizes. The blue and orange arrows denote the spatial information transferred through graph nodes and edges, respectively.

D. Region-to-Pixel Assignment

By performing dual interactive graph convolution, we can acquire improved region features from each GCN branch, i.e., $\mathbf{H}_1^{(L)}$ and $\mathbf{H}_2^{(L)}$, where L is the number of graph convolutional layers. The region-level output can then be calculated by

$$\widetilde{\mathbf{O}} = \mathbf{H}_1^{(L)} + \mathbf{H}_2^{(L)}, \quad (15)$$

in order to fuse the feature information at different spatial scales. To obtain the class predictions of image pixels, we need to re-project the region-level output $\widetilde{\mathbf{O}}$ back into the 2D image with grids of pixels, which is termed 'graph re-projection'. This region-to-pixel assignment is accomplished by

$$\mathbf{O} = \mathbf{P}\widetilde{\mathbf{O}}, \quad (16)$$

where $\mathbf{P}$ has also been used for graph projection in Eq. (2). Through graph re-projection, all the re-projected pixels will have diverse features, even if some of them were assigned to identical regions, and the contextual details can be well preserved as a result.

In addition to the discriminative loss function $\mathcal{L}_{dis}$ mentioned above, we also employ the widely-used cross-entropy loss function to penalize the differences between the network output and the labels of labeled pixels, i.e.,

$$\mathcal{L}_{cro} = -\sum_{i\in\mathbf{y}_G}\sum_{j=1}^{C}\mathbf{Y}_{ij}\ln\mathbf{O}_{ij}, \quad (17)$$

where $\mathbf{y}_G$ is the set of indices corresponding to the labeled pixels, C is the number of classes, and $\mathbf{Y}$ represents the label matrix. As a result, we have the following overall loss function:

$$\mathcal{L} = \mathcal{L}_{cro} + \alpha\mathcal{L}_{dis}, \quad (18)$$

where α is the coefficient assigned to the discriminative loss function and can be learned via gradient descent. Note that our DIGCN algorithm is trained in an end-to-end way, where full-batch gradient descent is used to update the network parameters [10]. The implementation details of our DIGCN are shown in Algorithm 1.

Finally, we will summarize the computational complexity of our proposed method. To be concrete, the graph projection step has a complexity of $\mathcal{O}(ndc)$, where n is the number of

Algorithm 1 The Proposed DIGCN for HSI Classification

Input: Input image; number of iterations $\mathcal{T}$; learning rate η; number of graph convolutional layers L;

Initialize the anchor point matrix $\mathbf{V}$ with SLIC algorithm;
// Train the DIGCN model
for $t = 1$ to $\mathcal{T}$ **do**
    Learn the region features $\mathbf{X}$ through Eq. (1) and Eq. (2);
    Perform dual interactive graph convolution based on Eqs. (11)-(14);
    Interpolate the region features back into the original 2D grids by Eq. (16);
    Calculate the error terms according to Eq. (18), and update the network parameters using full-batch gradient descent;
end for
Conduct label prediction based on the trained network;

Output: Predicted label for each pixel.

image pixels, d is the dimensionality of pixel features, and c is the number of projected graph nodes. Similarly, the graph projection step takes $\mathcal{O}(Cnc)$ with C representing the number of classes. Besides, the computational complexity of performing the dual interactive graph convolution is $\mathcal{O}(|\mathcal{E}|u^2dC)$, where $|\mathcal{E}|$ is the number of edges in the region-induced graph and u is the number of hidden units. As a consequence, the total computational complexity of our method is acceptable.

V. Experimental Results

In this section, extensive experiments will be conducted to test the effectiveness of the proposed DIGCN method, and the corresponding algorithm analyses will also be provided. Specifically, we first compare DIGCN with other state-of-the-art approaches on four public HSI datasets, where four metrics including per-class accuracy, Overall Accuracy (OA), Average Accuracy (AA), and Kappa coefficient are utilized for the evaluation of model performance. After that, we investigate the impact of the number of labeled pixels on classification accuracy. Then we analyze the convergence of the overall loss empirically. Afterwards, we demonstrate that both the interaction manipulation and discriminative regularization in our DIGCN are beneficial to obtaining the promising performance.

A. Experimental Settings

In our experiments, the performance of the proposed DIGCN is evaluated on four real-world benchmark datasets, i.e., the Indian Pines, University of Pavia, Salinas, and Houston University, which will be introduced in the appendix part. The DIGCN algorithm is implemented via TensorFlow with Adam optimizer. For all the utilized four real-world datasets, we randomly selected 30 labeled pixels (i.e., examples) per class for network training, and 15 labeled examples will be randomly chosen if the corresponding class contains less than 30 pixels. During training, 90% of the labeled examples are used to learn the network parameters and the remaining 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. As GCN-based methods usually do not require deep structure to achieve satisfactory performance [13], [51], we fix the number of graph convolutional layers to two for all the datasets. The selection of other hyperparameters used in our DIGCN, including the number of iterations $\mathcal{T}$, the learning rate η, the number of hidden units u, and the neighborhood sizes s_1 and s_2 have been shown in Table I. Meanwhile, we will investigate the parametric sensitivity of these hyperparameters in the appendix part.

TABLE I
The Hyperparameter Settings of Our Method on Different Datasets

Dataset	$\mathcal{T}$	η	u	s_1	s_2
Indian Pines	1500	0.001	60	1	2
University of Pavia	500	0.001	80	1	5
Salinas	2000	0.0001	100	1	4
Houston University	500	0.001	240	1	2

To evaluate the classification ability of our proposed DIGCN, several recent state-of-the-art HSI classification methods are employed for comparison. To be concrete, we employ two CNN-based methods, namely Diverse Region-based deep CNN (DR-CNN) [6] and CNN-Pixel-Pair Features (CNN-PPF) [7], together with three GCN-based methods, i.e., Graph Convolutional Network (GCN) [10], Spectral-Spatial Graph Convolutional Network (S^2GCN) [13], and miniGCN [16]. Besides, the classification performance of DIGCN is also compared with three traditional machine learning methods, i.e., Multiple Feature Learning (MFL) [52], Joint collaborative representation and SVM with Decision Fusion (JSDF) [53], and Edge-Preserving Filtering (EPF) [54]. Note that the hyperparameters of these baseline methods have been carefully tuned, which is presented in the appendix part. Meanwhile, all the methods are implemented ten times, and the mean accuracies and standard deviations over these ten independent implementations are reported. Moreover, to statistically validate the superiority of our proposed DIGCN to the remaining baselines, the paired t-test with significance level 0.05 is adopted in our experiments.

B. Classification Results

To reveal the effectiveness of the proposed DIGCN, we perform comparison evaluation of DIGCN against the aforementioned baseline methods.

1) Results on the Indian Pines Dataset: The quantitative results obtained by different methods on the Indian Pines dataset are tabulated in Table II, where the highest value in each row is marked in bold. We can observe that the proposed DIGCN outperforms all the other compared methods by a substantial margin in terms of OA, AA, and Kappa coefficient, and the standard deviations are relatively small as well. Meanwhile, one can also find that the improvement of our DIGCN is statistically significant revealed by the t-test. We can reasonably infer that the interaction of multi-scale spatial information can help guide the graph convolution operation, which further produces powerful representations.

A visual comparison of the classification results produced by various methods on the Indian Pines dataset is presented in Fig. 5, where the ground-truth map is exhibited in Fig. 5(b). As can be observed, some of the pixels belonging to 'Soybean-

TABLE II
PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON INDIAN PINES DATASET. THE '✓' DENOTES THAT OUR DIGCN APPROACH IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHODS REVEALED BY THE PAIRED T-TEST WITH SIGNIFICANCE LEVEL 0.05

ID	GCN [10]	S^2GCN [13]	miniGCN [16]	DR-CNN [6]	CNN-PPF [7]	MFL [52]	JSDF [53]	EPF [54]	DIGCN
1	95.00±2.80	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	95.00±2.64	98.06±0.58	**100.00±0.00**	90.12±15.01	99.15±1.90
2	56.71±4.42	84.43±2.50	56.66±8.14	80.38±1.50	73.53±5.61	74.72±0.66	90.75±3.19	82.34±5.83	**90.92±4.27**
3	51.50±2.56	82.87±5.53	61.97±5.92	82.21±3.53	81.34±3.76	82.14±0.70	77.84±3.81	84.25±9.75	**94.87±3.92**
4	84.64±3.16	93.08±1.95	87.52±3.79	99.19±0.74	91.84±3.53	93.60±0.55	**99.86±0.33**	46.93±12.40	98.32±1.93
5	83.71±3.20	**97.13±1.34**	90.12±1.87	96.47±1.10	93.69±0.84	92.54±0.43	87.20±2.73	96.44±3.26	95.29±3.44
6	94.03±2.11	97.29±1.27	96.66±1.68	**98.62±1.90**	97.46±1.01	98.40±0.27	98.54±0.28	96.46±3.53	95.95±1.85
7	92.31±0.00	92.31±0.00	98.75±3.95	**100.00±0.00**	75.38±8.73	97.28±0.45	**100.00±0.00**	96.31±8.13	96.36±4.03
8	96.61±1.86	99.03±0.93	94.67±4.22	99.78±0.22	98.01±0.69	99.82±0.05	99.80±0.31	99.82±0.48	**99.84±0.37**
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**	59.50±18.05	**100.00±0.00**
10	77.47±1.24	**93.77±3.72**	70.92±4.63	90.41±1.95	82.30±1.55	84.59±0.53	89.99±4.24	69.01±7.22	89.58±4.95
11	56.56±1.53	84.98±2.82	65.06±7.19	74.46±0.37	62.64±3.32	83.73±0.39	76.75±5.12	91.01±5.15	**91.75±3.52**
12	58.29±6.58	80.05±5.17	66.97±7.89	91.00±3.14	88.92±2.50	83.68±0.72	87.10±2.82	61.11±7.55	**93.81±3.73**
13	**100.00±0.00**	99.43±0.00	99.55±0.35	**100.00±0.00**	98.80±0.57	99.20±0.06	99.89±0.36	**100.00±0.00**	99.65±0.68
14	80.03±3.93	96.73±0.92	86.26±5.23	91.85±3.40	86.49±2.23	96.80±0.40	97.21±2.78	97.71±1.75	**98.94±1.73**
15	69.55±6.66	86.80±3.42	70.45±5.13	99.44±0.28	86.71±4.36	97.86±0.20	**99.58±0.68**	77.34±7.74	98.58±1.60
16	98.41±0.00	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	92.70±3.45	98.72±0.35	**100.00±0.00**	82.28±10.11	99.48±0.72
OA	69.24±1.56 ✓	89.49±1.08 ✓	73.45±1.34 ✓	86.65±0.59 ✓	80.09±1.56 ✓	87.38±0.12 ✓	88.34±1.39 ✓	82.83±1.55 ✓	**94.16±1.07**
AA	80.93±1.71 ✓	92.99±1.04 ✓	84.10±0.90 ✓	93.99±0.25 ✓	87.80±1.53 ✓	92.57±0.10 ✓	94.03±0.55 ✓	83.17±0.89 ✓	**96.41±0.54**
Kappa	65.27±1.80 ✓	88.00±1.23 ✓	69.92±1.51 ✓	84.88±0.67 ✓	77.52±1.74 ✓	85.64±0.14 ✓	86.80±1.55 ✓	80.57±1.74 ✓	**93.34±1.21**

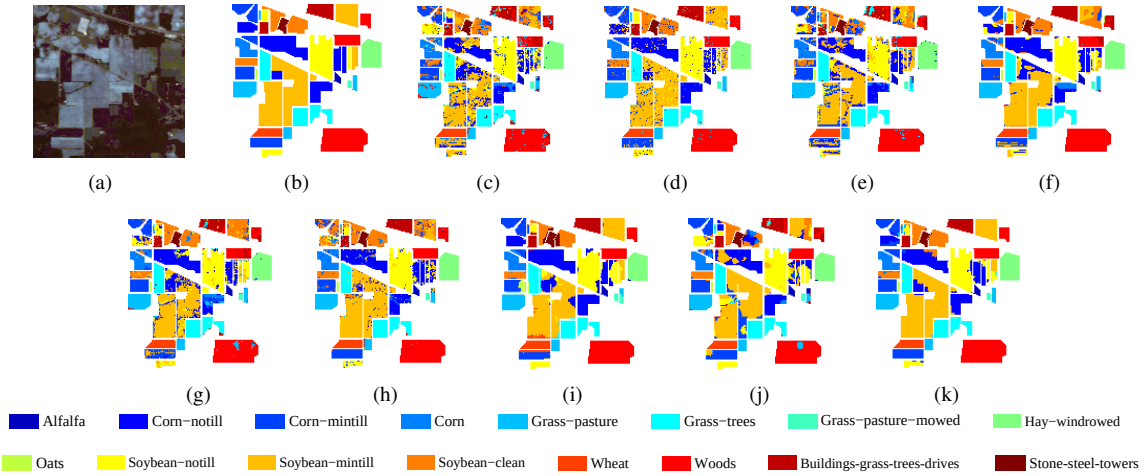


Fig. 5. Classification maps obtained by different methods on Indian Pines dataset. (a) False color image; (b) Ground-truth map; (c) GCN; (d) S^2GCN; (e) miniGCN; (f) DR-CNN; (g) CNN-PPF; (h) MFL; (i) JSDF; (j) EPF; (k) DIGCN.

TABLE III
PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON UNIVERSITY OF PAVIA DATASET. THE '✓' DENOTES THAT OUR DIGCN APPROACH IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHODS REVEALED BY THE PAIRED T-TEST WITH SIGNIFICANCE LEVEL 0.05

IID	GCN [10]	S^2GCN [13]	miniGCN [16]	DR-CNN [6]	CNN-PPF [7]	MFL [52]	JSDF [53]	EPF [54]	DIGCN
1	69.78±4.71	92.87±3.79	79.43±2.13	92.10±3.34	95.73±0.80	94.45±0.25	82.40±4.07	**97.78±1.27**	84.59±3.64
2	54.10±10.54	87.06±4.47	90.70±1.94	96.39±3.20	84.01±1.99	90.17±0.65	90.76±3.74	**97.30±2.03**	94.75±2.79
3	69.69±4.48	87.97±4.77	81.07±2.33	84.23±0.71	86.45±1.94	85.05±0.54	86.71±4.14	87.69±7.20	**95.10±3.31**
4	91.23±7.02	90.85±0.94	**96.84±0.74**	95.26±0.67	91.70±2.06	93.31±0.28	92.88±2.16	73.83±15.63	84.89±4.02
5	98.74±0.11	**100.00±0.00**	99.74±0.18	97.77±0.00	99.93±0.04	99.38±0.02	**100.00±0.00**	95.31±2.90	99.17±0.61
6	65.34±10.53	88.69±2.64	84.15±2.57	90.44±2.27	93.57±1.28	93.31±0.20	94.30±4.55	73.19±10.16	**99.97±0.05**
7	86.64±4.68	98.88±1.08	87.47±2.35	89.05±1.76	93.53±0.72	**99.39±0.04**	96.62±1.37	84.66±12.04	98.23±1.77
8	72.26±2.63	89.97±3.28	75.16±3.89	78.49±1.53	83.83±1.60	85.30±0.53	**94.69±3.74**	88.17±5.58	93.94±2.67
9	99.93±0.06	98.89±0.53	**99.98±0.05**	96.34±0.22	99.47±0.34	99.76±0.02	99.56±0.36	98.67±0.90	93.21±5.05
OA	66.19±3.43 ✓	89.74±1.70 ✓	87.17±1.05 ✓	92.62±1.15	88.72±0.95 ✓	91.54±0.30 ✓	90.82±1.30 ✓	89.08±3.70 ✓	**93.24±1.24**
AA	78.63±1.23 ✓	92.80±0.47 ✓	88.26±0.57 ✓	91.12±0.12 ✓	92.02±0.37 ✓	93.35±0.10 ✓	93.10±0.65 ✓	88.51±2.87 ✓	**93.76±0.90**
Kappa	58.39±3.28 ✓	86.65±2.06 ✓	83.21±1.33 ✓	90.27±1.44	85.43±1.18 ✓	88.98±0.38 ✓	88.02±1.62 ✓	85.96±4.50 ✓	**91.14±1.58**

mintill' are misclassified into 'Corn-notill' in all the classification maps, from which it can be inferred that these two land-cover types have similar spectral signatures and are difficult to distinguish. Besides, the GCN-based methods, namely S^2GCN and DIGCN, achieve better performance than other methods, demonstrating the power of graph convolution in HSI classification. However, GCN suffers from pepper-noise-like mistakes in several regions, which is because that traditional GCN does not incorporate any spatial context. Comparatively, the result of the proposed DIGCN yields smoother visual effect and shows fewer misclassifications than all the other methods.

2) Results on the University of Pavia Dataset: Table III reports the quantitative results of different methods on the University of Pavia dataset. Similar to the observations on the Indian Pines dataset, the results in Table III indicate that the proposed DIGCN is still in the first place, which again validates the strength of our proposed dual interactive graph convolution. Moreover, it is worthwhile to note that

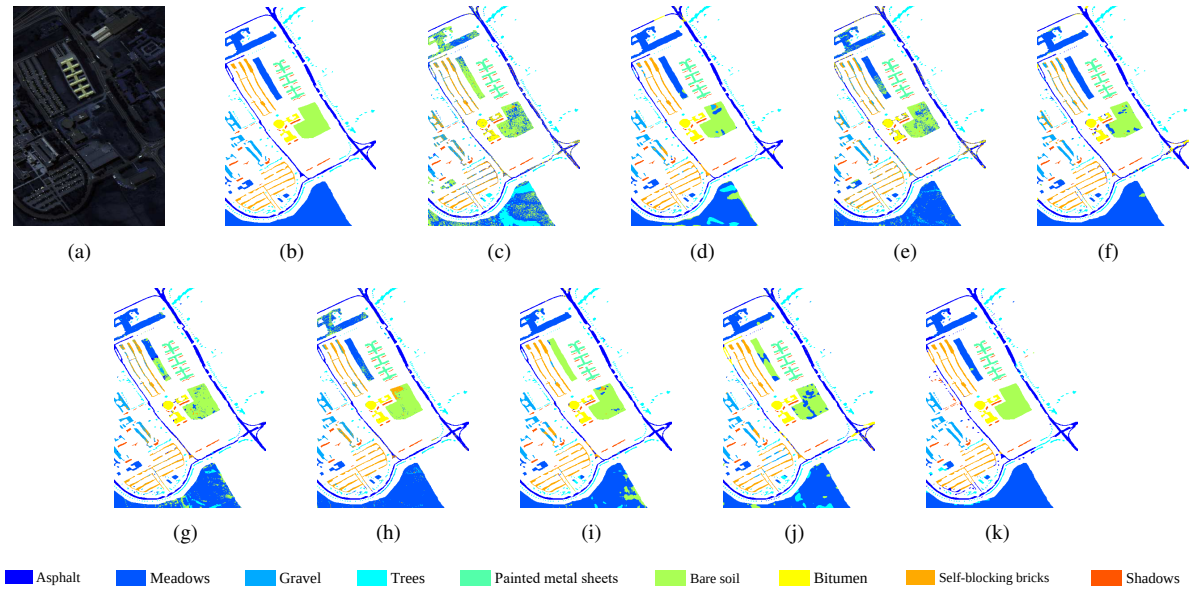


Fig. 6. Classification maps obtained by different methods on University of Pavia dataset. (a) False color image; (b) Ground-truth map; (c) GCN; (d) S^2GCN; (e) miniGCN; (f) DR-CNN; (g) CNN-PPF; (h) MFL; (i) JSDF; (j) EPF; (k) DIGCN.

DR-CNN achieves relatively high OA when compared with other baseline methods. The main reason is that DR-CNN can flexibly capture the variations of object appearances around objects by exploiting the diverse pre-defined convolutional kernels, which contributes to the effectiveness of DR-CNN in perceiving and handling the irregular class boundaries. Therefore, the advantage of DR-CNN becomes prominent on the datasets containing various irregular class boundaries, such as the University of Pavia dataset. Owing to the exploration of multi-scale spatial information, our DIGCN is able to precisely perceive the diverse types of image regions. As a consequence, the classification map of DIGCN exhibited in Fig. 6 reveals stronger spatial correlations and fewer mistakes than other competitors.

3) Results on the Salinas Dataset: Experimental results of different methods on the Salinas dataset are listed in Table IV. Compared with the best baseline method (i.e., JSDF), DIGCN exhibits improvement of approximately 3% in terms of OA and Kappa coefficient. Although DIGCN cannot reach the best result in terms of AA, it yields more stable performance than the competitors. Especially, in the classes 'Grapes untrained' (ID = 8) and 'Vineyard untrained' (ID = 15), the class-specific accuracies of the proposed DIGCN are even approximately 10% and 17% higher than those of the baseline methods. As a consequence, we can see that DIGCN is able to achieve promising results in different scenarios. Additionally, different from the CNN-based model (namely DR-CNN and CNN-PPF), our proposed DIGCN is able to achieve promising results with very few labeled pixels. For example, by using 30 labeled pixels per class for training, the proposed DIGCN outperforms CNN-PPF with 200 labeled pixels per class for training [7]. Fig. 7 visualizes the classification results generated by different methods, where the classification map obtained by the proposed DIGCN is noticeably closer to the ground-truth map (see Fig. 7(b)), compared with other methods.

4) Results on the Houston University Dataset: Table V gives the classification results obtained by different methods on the Houston University dataset. Note that the classification accuracies of the comparison methods are relatively low in some land-cover classes, such as 'Residential' (ID = 7) and 'Road' (ID = 9), which might be explained by the existence of a few noisy pixels in these two classes. One can observe that miniGCN is generally superior to GCN, owing to the incorporation of local spatial context. Remarkably, our proposed DIGCN achieves stable performance improvements when compared with other methods, which is in consistent with the observations on the above three datasets. The classification results produced by different methods are visualized in Fig. 8. Generally, the pixel-wise classification model, namely GCN, results in much salt and pepper noise in the classification map. As expected, the proposed DIGCN method obtains smoother and more compact map when compared with the competitors.

C. Impact of the Number of Labeled Examples

In this experiment, classification accuracies of the proposed DIGCN and the baseline methods under different numbers of initially labeled examples are shown in Fig. 9. Here we vary the number of labeled examples per class from 5 to 30 with an interval of 5 and report the OA gained by all the methods on four datasets, i.e., the Indian Pines, University of Pavia, Salinas, and Houston University. As observed in Fig. 9, the classification performance of all the methods can be generally improved by increasing the number of labeled examples. However, on the Salinas dataset, the performance of the CNN-based methods significantly declines when the label information is very limited. It can be inferred that the scarce supervision information might lead to overfitting problem. Unlike CNN, GCN is able to utilize the unlabeled data for model training, and thus the performance of GCN-based methods is relatively stable when faced with insufficient

TABLE IV
PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON SALINAS DATASET. THE '✓' DENOTES THAT OUR DIGCN APPROACH IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHODS REVEALED BY THE PAIRED T-TEST WITH SIGNIFICANCE LEVEL 0.05

ID	GCN [10]	S^2GCN [13]	miniGCN [16]	DR-CNN [6]	CNN-PPF [7]	MFL [52]	JSDF [53]	EPF [54]	DIGCN
1	98.62±0.86	99.01±0.44	99.40±0.85	99.40±1.54	99.77±0.21	99.63±0.07	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
2	99.07±1.21	99.18±0.59	99.90±0.06	99.46±0.16	98.69±0.89	99.34±0.06	**100.00±0.00**	**100.00±0.01**	98.10±1.89
3	97.03±1.10	97.15±2.76	95.98±3.19	98.58±1.69	99.50±0.49	99.77±0.03	**100.00±0.00**	95.19±2.15	99.91±0.25
4	99.28±0.49	99.11±0.55	99.76±0.25	99.70±0.45	99.81±0.04	98.88±0.07	**99.93±0.09**	97.71±0.40	94.28±4.99
5	98.58±0.79	97.55±2.35	98.37±0.88	98.90±0.74	96.64±1.26	98.72±0.04	99.77±0.31	**99.92±0.12**	94.86±2.00
6	99.58±0.30	99.32±0.35	99.85±0.23	99.57±0.78	99.32±0.86	99.18±0.11	**100.00±0.00**	99.98±0.05	96.48±2.24
7	99.13±0.25	99.06±0.27	99.74±0.21	99.50±0.66	99.59±0.13	98.61±0.12	**99.99±0.01**	99.57±0.81	**99.99±0.03**
8	67.94±8.33	70.68±5.20	64.42±7.73	75.59±8.19	74.77±4.01	76.57±0.71	87.79±4.89	86.91±6.12	**97.65±1.81**
9	98.50±0.85	98.32±1.79	99.37±0.39	**99.75±0.41**	98.99±0.18	99.01±0.05	99.67±0.33	99.48±0.16	99.58±1.15
10	89.64±1.57	90.97±2.59	93.40±2.06	94.29±2.24	89.32±3.04	93.10±0.30	**96.53±2.55**	88.66±4.11	95.42±3.46
11	94.80±2.98	98.00±1.65	96.66±2.75	97.57±2.19	97.65±1.49	96.81±0.30	**99.76±0.21**	94.63±3.60	96.26±3.55
12	99.71±0.08	99.56±0.59	**100.00±0.00**	99.99±0.05	99.82±0.30	98.84±0.20	**100.00±0.00**	99.79±0.32	91.86±3.74
13	97.99±0.61	97.83±0.72	99.93±0.17	99.95±0.09	97.70±0.50	99.38±0.08	**100.00±0.00**	99.34±1.19	92.19±4.26
14	93.58±2.60	95.75±1.65	96.75±1.46	98.57±1.13	94.14±1.22	96.20±0.32	**98.71±0.72**	98.11±1.33	95.95±2.76
15	66.18±9.08	70.36±3.62	68.90±7.99	72.18±9.28	79.12±1.99	78.85±0.56	81.86±5.26	71.82±11.68	**98.44±1.58**
16	97.24±1.21	96.90±1.97	97.35±0.60	98.45±0.57	98.65±0.31	99.69±0.06	98.99±0.63	99.43±2.06	**100.00±0.00**
OA	87.16±0.85 ✓	88.39±1.01 ✓	87.38±1.34 ✓	90.35±1.14 ✓	90.52±0.77 ✓	91.20±0.13 ✓	94.67±0.77 ✓	91.35±2.53 ✓	**97.61±0.69**
AA	93.55±0.39 ✓	94.30±0.47 ✓	94.36±0.58 ✓	95.72±0.39	95.22±0.34	95.79±0.04	**97.69±0.34**	95.66±0.90	96.94±0.66
Kappa	85.74±0.92 ✓	87.10±1.12 ✓	85.99±1.48 ✓	89.26±1.26 ✓	89.46±0.85 ✓	90.21±0.14 ✓	94.06±0.85 ✓	90.39±2.78 ✓	**97.34±0.76**

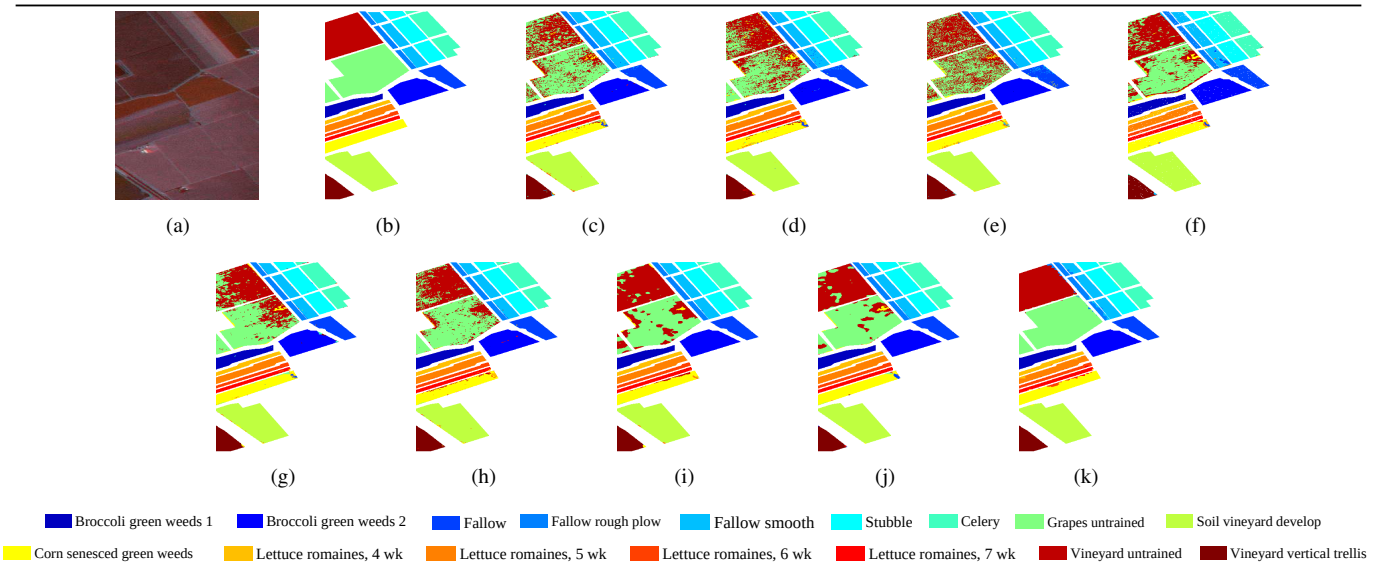


Fig. 7. Classification maps obtained by different methods on Salinas dataset. (a) False color image; (b) Ground-truth map; (c) GCN; (d) S^2GCN; (e) miniGCN; (f) DR-CNN; (g) CNN-PPF; (h) MFL; (i) JSDF; (j) EPF; (k) DIGCN.

TABLE V
PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT METHODS ON HOUSTON UNIVERSITY DATASET. THE '✓' DENOTES THAT OUR DIGCN APPROACH IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHODS REVEALED BY THE PAIRED T-TEST WITH SIGNIFICANCE LEVEL 0.05

ID	GCN [10]	S^2GCN [13]	miniGCN [16]	DR-CNN [6]	CNN-PPF [7]	MFL [52]	JSDF [53]	EPF [54]	DIGCN
1	88.16±1.90	96.30±3.07	94.85±3.58	95.62±5.41	**98.62±0.71**	91.000.90	97.41±1.21	94.96±3.10	93.07±2.73
2	97.20±0.48	98.57±1.47	98.35±1.71	96.78±3.92	98.15±0.53	94.97±0.62	**99.48±0.25**	95.30±4.43	94.17±2.93
3	97.91±0.13	98.88±0.43	98.09±1.74	96.75±1.83	99.01±0.33	99.74±0.01	**99.88±0.22**	98.87±5.50	95.00±1.68
4	96.55±0.41	97.68±2.89	95.60±2.13	93.41±3.23	93.21±0.48	93.14±0.45	**98.22±2.80**	96.01±6.45	90.47±4.09
5	89.79±0.71	97.66±1.12	98.64±0.72	99.15±0.78	99.13±0.73	98.36±0.15	**100.00±0.00**	96.19±6.36	**100.00±0.00**
6	98.21±1.15	96.84±1.17	96.58±1.80	93.83±2.22	91.26±5.25	97.24±0.40	**99.32±1.09**	97.84±6.21	94.10±3.86
7	73.67±1.94	83.48±5.89	76.05±1.53	80.71±6.26	81.54±5.84	88.02±0.52	91.93±4.91	88.65±5.51	**96.06±2.80**
8	65.71±4.64	76.15±4.37	77.28±3.75	78.32±5.43	68.15±3.97	64.28±0.71	68.82±6.16	**88.86±11.31**	73.36±5.63
9	70.27±3.03	82.17±1.78	78.98±2.24	76.90±5.62	77.17±1.65	67.91±0.57	69.47±8.56	85.37±12.00	**94.33±3.33**
10	74.71±2.32	86.85±8.32	82.92±3.80	81.99±7.04	**92.12±1.76**	87.64±0.95	85.63±9.32	91.01±12.99	88.76±7.63
11	75.36±2.37	88.57±5.06	70.07±3.69	84.04±4.86	81.05±3.31	89.20±0.47	**94.51±3.82**	89.52±12.87	90.68±4.32
12	79.29±4.80	78.64±4.79	85.87±3.99	81.92±8.31	78.10±5.07	77.65±0.46	84.33±5.33	85.40±12.79	**87.08±4.25**
13	12.09±2.68	75.62±6.93	80.93±2.57	86.54±2.58	72.55±4.36	80.76±0.40	**98.10±1.28**	63.43±12.60	92.79±4.34
14	86.03±3.31	99.45±0.44	97.73±1.87	99.31±1.18	99.85±0.16	98.01±0.29	**100.00±0.00**	95.22±4.55	**100.00±0.00**
15	95.29±1.67	98.03±1.07	99.04±0.76	99.60±0.50	98.60±0.34	98.47±0.11	**99.86±0.36**	99.79±4.20	97.90±1.62
OA	80.35±0.61 ✓	89.31±1.00 ✓	87.00±0.71 ✓	88.08±1.09 ✓	87.54±1.03 ✓	86.66±0.13 ✓	90.51±0.95 ✓	90.88±1.59	**91.72±0.64**
AA	80.02±0.46 ✓	90.33±1.06 ✓	88.73±0.58 ✓	89.66±0.95 ✓	88.57±0.77 ✓	88.43±0.11 ✓	92.46±0.75	91.09±1.63	**92.52±0.16**
Kappa	78.72±0.66 ✓	88.44±1.08 ✓	85.94±0.77 ✓	87.10±1.18 ✓	86.53±1.12 ✓	85.56±0.14 ✓	89.74±1.03 ✓	90.13±1.65	**91.03±0.69**

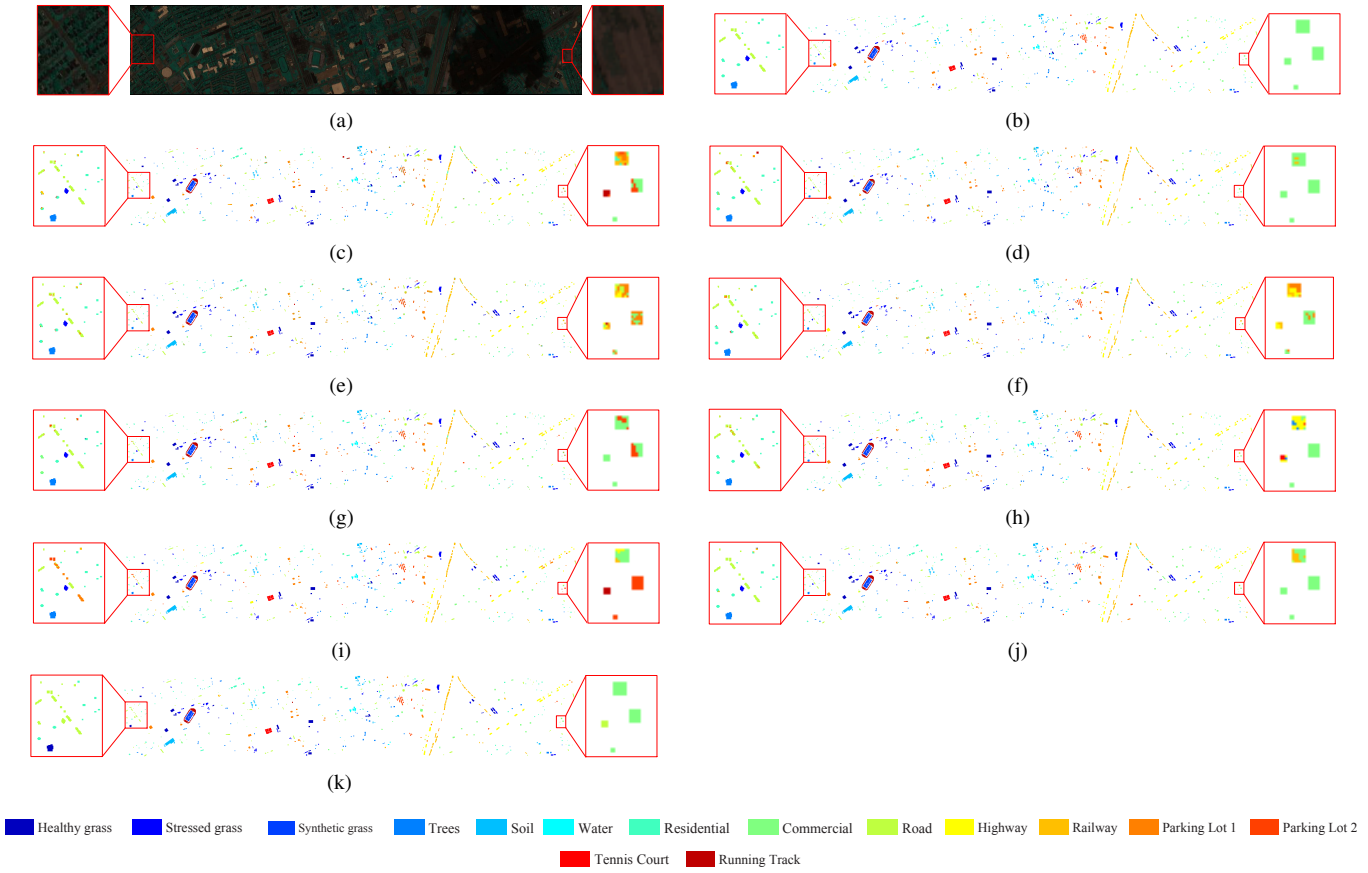


Fig. 8. Classification maps obtained by different methods on Houston University dataset. (a) False color image; (b) Ground-truth map; (c) GCN; (d) S^2GCN; (e) miniGCN; (f) DR-CNN; (g) CNN-PPF; (h) MFL; (i) JSDF; (j) EPF; (k) DIGCN. In (a)-(k), zoomed-in views of the regions are denoted by red boxes.

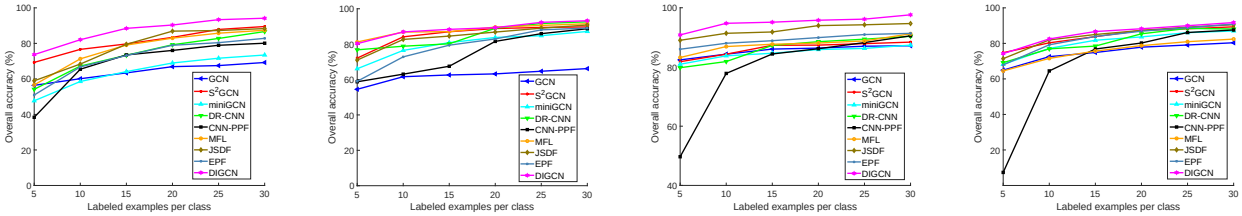


Fig. 9. Overall accuracies of various methods under different numbers of labeled examples per class. (a) Indian Pines dataset; (b) University of Pavia dataset; (c) Salinas dataset; (d) Houston University dataset.

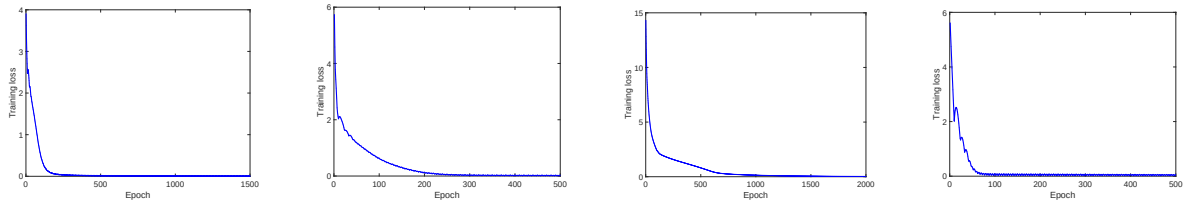
labels. Moreover, it is noteworthy that the proposed DIGCN can generally achieve superior classification results than the competitors, even when the label rate is extremely low (namely five labels per class). This is mainly due to the utilization of the large number of unlabeled pixels during graph convolution. Meanwhile, by leveraging the interaction of multi-scale spatial information, the graph information can be refined, which helps enhance the expressive power of the generated representations.

D. Convergence Analysis

To evaluate the convergence performance of the proposed DIGCN, in Fig. 10, we exhibit the loss curves on different datasets. The number of labeled pixels per class is kept identical to the above experiments in Section V-B. From the plots in Fig. 10, we can observe that our DIGCN converges very quickly within 500 epochs on the Indian Pines, University of Pavia dataset, and Houston University datasets. Since the learning rate adopted on the Salinas dataset is relatively small, the corresponding convergence speed can be slower than those on other datasets. However, the convergence can still be achieved within 1500 epochs. Owing to the satisfactory convergence performance, DIGCN is effective and efficient for HSI classification.

E. Ablation Study

As is mentioned in the introduction, the proposed DIGCN algorithm comprises two parts that are critical for boosting the classification performance, i.e., the dual interactive graph convolution and the discriminative loss function. Here we use the Indian Pines dataset to investigate the contributions of these two operations, where the number of labeled pixels per class is kept identical to the above experiments in Section V-B. Every time we report the OA obtained by DIGCN without one of the aforementioned operations. For simplicity, 'DIGCN-v_1'



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

Fig. 10. Training losses of DIGCN on different datasets. (a) Indian Pines dataset; (b) University of Pavia dataset; (c) Salinas dataset; (d) Houston University dataset.

TABLE VI
PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT MODEL SETTINGS ON INDIAN PINES DATASET

ID	DIGCN-v_1	DIGCN-v_2	DIGCN-s_1	DIGCN-s_2	DIGCN
1	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.15±1.90
2	86.39±4.81	82.68±5.79	82.94±5.81	82.23±6.95	**90.92±4.27**
3	94.74±3.06	93.85±2.86	94.63±3.73	88.84±4.72	**94.87±3.92**
4	97.88±1.80	**99.46±1.07**	96.76±3.11	98.77±2.79	98.32±1.93
5	92.55±4.36	92.94±4.09	83.30±8.13	84.69±0.75	**95.29±3.44**
6	94.54±3.37	94.81±5.25	94.14±3.73	85.90±2.91	**95.95±1.85**
7	97.85±2.78	97.46±3.82	98.25±4.74	**100.00±0.00**	96.36±4.03
8	99.91±0.21	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	99.84±0.37
9	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
10	85.24±2.95	**91.76±3.36**	81.58±7.17	82.47±5.69	89.58±4.95
11	85.67±4.98	91.13±2.78	86.73±3.98	**92.18±4.28**	91.75±3.52
12	93.48±3.34	**94.15±2.27**	88.10±6.69	85.55±3.02	93.81±3.73
13	99.23±1.91	**99.93±0.20**	99.89±0.23	98.88±1.94	99.65±0.68
14	98.71±1.81	99.94±0.08	97.77±2.52	**99.95±0.08**	98.94±1.73
15	**99.19±0.52**	98.47±3.12	95.57±3.52	98.06±3.99	98.58±1.60
16	99.24±1.29	99.43±1.61	**100.00±0.00**	**100.00±0.00**	99.48±0.72
OA	91.33±1.33	92.95±0.91	89.72±1.31	90.30±1.27	**94.16±1.07**
AA	95.29±0.61	96.00±0.43	93.73±0.79	93.60±0.55	**96.41±0.54**
Kappa	90.13±1.50	91.97±1.03	88.32±1.47	88.95±1.43	**93.34±1.21**

TABLE VII
RUNNING TIME COMPARISON (IN SECONDS) OF DIFFERENT METHODS. 'IP' DENOTES INDIAN PINES DATASET, 'PAVIAU' DENOTES UNIVERSITY OF PAVIA DATASET, AND 'UH' DENOTES HOUSTON UNIVERSITY DATASET. THE BEST RECORD IS HIGHLITED IN BOLD ON EACH DATASET

Dataset	GCN [10]	S^2GCN [13]	miniGCN [16]	DR-CNN [6]	CNN-PPF [7]	DIGCN
IP	58	71	65	2753	1495	**53**
PaviaU	1783	1803	240	3251	1545	**187**
Salinas	3497	3528	**594**	3477	1769	616
UH	901	971	179	2969	1223	**124**

and 'DIGCN-v_2' are utilized to indicate the reduced models that remove the interaction operation and the discriminative loss function, respectively. Table VI exhibits the comparative results on the Indian Pines dataset. We observe that DIGCN achieves higher OA over DIGCN-v_1 and DIGCN-v_2 by margins of 2.83% and 1.21%, respectively, which validates the contributions of the dual interactive graph convolution and the discriminative loss function to performance improvement. In addition, we also investigate the importance of using different neighborhood sizes for graph convolution. Concretely, Table VI exhibits the classification results of adopting only a single neighborhood size, where 'DIGCN-s_1' and 'DIGCN-s_2' indicate the reduced models using the neighborhood size s_1 and s_2 for graph convolution, respectively. We can observe that even DIGCN-v_1 which removes the interaction operation from DIGCN yields better results than DIGCN-s_1 and DIGCN-s_2. Hence it is critical to utilize the contextual information from different spatial scales.

F. Running Time

Table VII reports the running time of different deep models including GCN, S^2GCN, miniGCN, DR-CNN, CNN-PPF, and our DIGCN on the four datasets adopted above, where the number of labeled pixels per class is kept identical to the above experiments in Section V-B. 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 introducing the interaction of contextual information needs additional computation, it can be observed that DIGCN still shows high efficiency, especially on the University of Pavia and the Salinas datasets, which is owing much to the utilization of graph projection operation. Since graph size gets significantly reduced, the corresponding graph convolution step can be accelerated greatly. Meanwhile, the computational complexity of the dual interactive graph convolution is also acceptable. It is also notable that miniGCN shows higher efficiency than our DIGCN on the Salinas dataset. This is mainly due to that miniGCN adopts the mini-batch strategy for network training, which is more efficient than the full-batch fashion used by DIGCN. As a consequence, when coping with large-scale datasets, the time consuming of miniGCN can even be lower than DIGCN. Nevertheless, the time cost of our DIGCN is still comparable to that of miniGCN. All the comparison results indicate that the proposed DIGCN is highly effective and efficient for HSI classification.

VI. CONCLUSION

In this paper, we propose a novel Dual Interactive Graph Convolutional Network (DIGCN) for HSI classification. In our DIGCN model, the dual GCN branches are employed to explore the contextual information at different spatial scales and work interactively to refine the graph information. As such, the expressive power of the generated feature representations can be gradually enhanced with the interactive graph convolution. This dual interactive graph convolution offers a new way to effectively incorporate the multi-scale spatial information of HSI and the time complexity is acceptable. Experimental results on four real-world HSI datasets validate the effectiveness of our proposed DIGCN.

Since our proposed method is a bit sensitive to the selection of neighborhood sizes, we plan to adaptively determine them by resorting to the self-attention mechanism. Apart from this, it is also worthwhile to exploit the supervision signals contained in the hyperspectral data itself by utilizing some recent self-supervised learning techniques.

REFERENCES

[1] P. Zhong, Z. Gong, and J. Shan, "Multiple instance learning for multiple diverse hyperspectral target characterizations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 246–258, Jan. 2020.

[2] 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.

[3] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.

[4] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[5] M. Crawford and W. Kim, "Manifold learning for multi-classifier systems via ensembles," in *Multiple Classifier Systems.* Springer, 2009, pp. 519–528.

[6] 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.

[7] 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.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4489–4497.

[9] 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.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[11] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1024–1034.

[12] S. Pan, R. Hu, S. F. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020.

[13] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral-spatial graph convolutional networks for semi-supervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.

[14] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–12, 2020.

[15] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May 2020.

[16] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–13, 2020.

[17] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 597–612, Jan. 2021.

[18] S. Niu, Q. Chen, L. De Sisternes, Z. Ji, Z. Zhou, and D. L. Rubin, "Robust noise region-based active contour model via local similarity factor for image segmentation," *Pattern Recognit.*, vol. 61, pp. 104–119, Jan. 2017.

[19] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

[20] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectralspatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

[21] 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.

[22] Y. Y. Tang, Y. Lu, and H. Yuan, "Hyperspectral image classification based on three-dimensional scattering wavelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2467–2480, May 2015.

[23] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, Feb. 2001.

[24] 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.

[25] X. Cao, C. Wei, J. Han, and L. Jiao, "Hyperspectral band selection using improved classification map," *IEEE Trans. Geosci. Remote Sens.*, vol. 14, no. 11, pp. 2147–2151, Nov. 2017.

[26] S. K. Roy, J. M. Haut, M. E. Paoletti, S. R. Dubey, and A. Plaza, "Generative adversarial minority oversampling for spectral-spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2021.

[27] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, Sep. 2018.

[28] G. Ding, Y. Guo, K. Chen, C. Chu, J. Han, and Q. Dai, "DECODE: Deep confidence network for robust image classification," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3752–3765, Aug. 2019.

[29] C. Gong, J. Yang, J. J. You, and M. Sugiyama, "Centroid estimation with guaranteed efficiency: A general framework for weakly supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020.

[30] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[31] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, Jun. 2017.

[32] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021.

[33] G. Camps-Valls, T. V. Bandos Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3044–3054, Oct. 2007.

[34] Y. Gao, R. Ji, P. Cui, Q. Dai, and G. Hua, "Hyperspectral image classification through bilayer graph-based learning," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2769–2778, Jul. 2014.

[35] B. Cui, X. Xie, X. Ma, G. Ren, and Y. Ma, "Superpixel-based extended random walker for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3233–3243, Jun. 2018.

[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, 2020.

[37] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," *arXiv preprint arXiv:2002.00388*, 2020.

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

[39] T. Lei and H. Liu, "Relational learning via latent social dimensions," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. ACM, 2009, pp. 817–826.

[40] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention." in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 4623–4629.

[41] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet Markov random fields: Semi-supervised community detection in attribute networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, 2019, pp. 152–159.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf Mach. Learn. (ICML)*, 2010, pp. 807–814.

[43] B. Jiang, C. Ding, B. Luo, and J. Tang, "Graph-laplacian PCA: Closed-form solution and robustness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 3492–3498.

[44] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 11 313–11 320.

[45] M. Shi, Y. Tang, and X. Zhu, "Topology and content co-alignment graph convolutional learning," *arXiv preprint arXiv:2003.12806*, 2020.

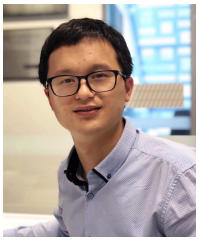
[46] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognit.*, vol. 43, no. 7, pp. 2367–2379, Jul. 2010.

[47] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "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.

[48] Y. Li and A. Gupta, "Beyond grids: Learning graph representations for visual recognition," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 9225–9235.

[49] 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, Dec. 2011.
[50] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 3538–3545.
[51] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. ACM, 2018, pp. 1416–1424.
[52] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1592–1606, Mar. 2015.
[53] 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.
[54] 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 2013.

Sheng Wan received his B.S. degree from Nanjing Agricultural University (NJAU) in 2016. He is currently working toward 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, China. His research interests include deep learning and hyperspectral image processing. His supervisor is Dr. Chen Gong.



Shirui Pan received a Ph.D. in computer science from the University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is currently a senior lecturer with Department of Data Science and AI, Faculty of Information Technology, Monash University, Australia. Prior to this, he was a Lecturer with the School of Software, University of Technology Sydney. His research interests include data mining and machine learning. To date, Dr. Pan has published over 80 research papers in top-tier journals and conferences, including the IEEE T-NNLS, IEEE T-KDE, IEEE T-CYB, WWW, AAAI, and ICDM.

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 more than 30 peer-reviewed papers in international journals such as the IEEE T-NNLS, the IEEE T-IP, the IEEE T-GRS, the IEEE J-STSP, and the IEEE J-STARS. 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 T-NNLS, the IEEE T-IP, the IEEE T-GRS, the IEEE J-STARS, the IEEE J-STSP, and IEEE GRSL.

Xiaojun Chang received the Ph.D. degree from the Centre for Artificial Intelligence and the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, in 2016. He is currently a Senior Lecturer with Department of Data Science and AI, Faculty of Information Technology, Monash University Clayton Campus, Australia. He is also a Distinguished Adjunct Professor with the Faculty of Computing and Information Technology, King Abdulaziz University. Before joining Monash, he was a Postdoctoral Research Associate with the School of Computer Science, Carnegie Mellon University, working with Prof. Alex Hauptmann. He has spent most of time working on exploring multiple signals (visual, acoustic, and textual) for automatic content analysis in unconstrained or surveillance videos. He has achieved top performance in various international competitions, such as TRECVID MED, TRECVID SIN, and TRECVID AVS. He is an ARC Discovery Early Career Researcher Award (DECRA) Fellow from 2019 to 2021.

Jian Yang received the PhD degree from Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a Postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a Chang-Jiang professor in the School of Computer Science and Technology of NUST. He is the author of more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited over 24000 times in the Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is/was an associate editor of Pattern Recognition, Pattern Recognition Letters, IEEE T-NNLS, and Neurocomputing. He is a Fellow of IAPR.

Chen Gong (M'16) received his B.E. degree from East China University of Science and Technology (ECUST) in 2010, and dual doctoral degree from Shanghai Jiao Tong University (SJTU) and University of Technology Sydney (UTS) in 2016 and 2017, respectively. Currently, he is a full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has published more than 100 technical papers at prominent journals and conferences such as IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, IEEE T-CYB, IEEE T-CSVT, IEEE T-MM, IEEE T-ITS, ACM T-IST, ICML, NeurIPS, CVPR, AAAI, IJCAI, ICDM, etc. He also serves as the reviewer for more than 20 international journals such as AIJ, IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, and also the SPC/PC member of several top-tier conferences such as ICML, NeurIPS, CVPR, AAAI, IJCAI, ICDM, AISTATS, etc. He received the "Excellent Doctorial Dissertation" awarded by Shanghai Jiao Tong University (SJTU) and Chinese Association for Artificial Intelligence (CAAI). He was enrolled by the "Young Elite Scientists Sponsorship Program" of Jiangsu Province and China Association for Science and Technology. He was also the recipient of "Wu Wen-Jun AI Excellent Youth Scholar Award".