Hyperspectral Image Classification With Context-Aware Dynamic Graph Convolutional Network

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

***Abstract*— In hyperspectral image (HSI) classification, spatial context has demonstrated its significance in achieving promising performance. However, conventional spatial context-based methods simply assume that spatially neighboring pixels should correspond to the same land-cover class, so they often fail to correctly discover the contextual relations among pixels in complex situations, and thus leading to imperfect classification results on some irregular or inhomogeneous regions such as class boundaries. To address this deficiency, we develop a new HSI classification method based on the recently proposed graph convolutional network (GCN), as it can flexibly encode the relations among arbitrarily structured non-Euclidean data. Different from traditional GCN, there are two novel strategies adopted by our method to further exploit the contextual relations for accurate HSI classification. First, since the receptive field of traditional GCN is often limited to fairly small neighborhood, we proposed to capture long-range contextual relations in HSI by performing successive graph convolutions on a learned region-induced graph which is transformed from the original 2-D image grids. Second, we refine the graph edge weight and the connective relationships among image regions simultaneously by learning the improved similarity measurement and the "edge filter," so that the graph can be gradually refined to adapt to the representations generated by each graph convolutional layer. Such updated graph will in turn result in faithful region representations, and vice versa. The experiments carried out on four real-world benchmark data sets demonstrate the effectiveness of the proposed method.**

***Index Terms*— Contextual relations, graph convolutional network (GCN), graph updating, hyperspectral image (HIS) classification.**

I. Introduction

HYPERSPECTRAL image (HSI) has recently received considerable attention in a variety of applications such as military target detection, mineral identification, and disaster prevention [1], [2]. In contrast to traditional panchromatic and multispectral remote-sensing images, HSI consists of hundreds of contiguous spectral bands, which are helpful to distinguishing the targets with different materials. Thanks to the high spectral resolution, HSI has shown its advantages in identifying various land-cover types or targets [3].

Up to now, significant efforts have been made in developing diverse kinds of HSI classification methods. The early-staged algorithms are mainly based on the simple combination of spectral signatures and conventional pattern recognition methods, such as nearest neighbor classifier and support vector machines (SVMs) [4], [5]. However, these methods isolatedly classify each image pixel without considering the spatial correlation among the pixels, so they will encounter the spectral variability problem [6] and generate imperfect classification results.

To address this shortcoming, the spatial context naturally becomes another type of useful information in addition to the spectra. It is now commonly acknowledged that the introduction of spatial context offers probability to improve HSI classification results and is the key to generating discriminative features for classification [7]; hence, there is a huge demand for the algorithms which can effectively discover and incorporate spatial context. During the past decades, researchers have reported various HSI classification methods utilizing spatial context. The first attempt was accomplished

Manuscript received February 23, 2020; accepted April 26, 2020. This work was supported in part by the NSF of China under Grant 61973162, Grant U1713208, Grant 61971428, and Grant 61671456, in part by the NSF of Jiangsu Province under Grant BK20171430 and Grant BK20190444, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319 and Grant 30919011228, 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, the "Young Elite Scientists Sponsorship Program" by CAST under Grant 2018QNRC001, in part by the Guangdong Key Area Research Project under Grant 2018B010108003, in part by the Program for Changjiang Scholars, and in part by the "111" Program under Grant AH92005. *(Corresponding author: Chen Gong.)*

Sheng Wan, Guangyu Li, and Jian Yang are with the PCA Laboratory, Nanjing University of Science and Technology, Nanjing 210094, China, also 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, in part by the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, Nanjing University of Science and Technology, Nanjing 210094, China, and in part by the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wansheng315@hotmail.com; liguangyu0627@hotmail.com; csjyang@njust.edu.cn).

Chen Gong is with the PCA Laboratory, Nanjing University of Science and Technology, Nanjing 210094, China, also 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 Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: chen.gong@njust.edu.cn).

Ping 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).

Shirui Pan is with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (e-mail: shirui.pan@monash.edu).

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

Digital Object Identifier 10.1109/TGRS.2020.2994205

by Kettig and Landgrebe [8], where the well-known ECHO classifier is proposed to extract contextual information. After that, Markov random field (MRF), which is an undirected graphical model [9], became a popular approach to include spatial context for HSI classification. For instance, in [10], a relative homogeneity index for each pixel is introduced in MRF-based classification to determine an appropriate weighting coefficient for the contextual contribution. Apart from this, a novel framework combining SVM and MRF is proposed for contextual HSI classification [11]. Although the spatial context exploited by MRF improves the classification results in smooth image areas [12], the aforementioned methods do not explicitly investigate the contextual relations among individual pixels (or regions), and they only implicitly assume that nearby pixels have a large probability to take the same class label regardless whether they are in object boundary regions or homogeneous regions [13]. As a result, the semantic meaning carried by image patches cannot be well preserved, and the classification errors may appear within the object area of a certain class. Moreover, the pixels around some irregular or inhomogeneous regions are also very likely to be misclassified due to the inappropriate utilization of local contextual information. Consequently, the simple assumption of smoothness and homogeneity over the whole image is unreasonable.

Different from MRF-based models which are usually utilized for postprocessing-based classification, texture plays an important role in spatial preprocessing for HSI, particularly when the classes of interest are quite similar [14], [15]. A widely adopted approach to extract texture is processing each spectral band independently via using the existing texture methods [16], which inevitably ignores the fact that each pixel is characterized in a multidimensional space. To take into account the multidimensional nature, some researchers focused on introducing information of interband dependencies. For example, in [17], the texture is approached from the perspective of a Gaussian mixture generative process by assuming that a given image can be generated with a set of texture image primitives. Besides, Safia and He [14] proposed a texture descriptor to extract interband dependencies among a large number of spectral bands and exploit the spatial variations among different bands simultaneously. However, the texture information fails to capture the detailed relations among image pixels and can only be used to characterize the homogeneity of image regions. As a consequence, high-level information cannot be directly generated using texture. Moreover, the contextual relations exploited by conventional texture-based methods are often restricted in a small local region, and thus it is unable to capture the long-range dependencies among faraway pixels.

To alleviate the aforementioned defects and effectively exploit the contextual relations, in this article, we propose a novel “context-aware dynamic graph convolutional network" (CAD-GCN). In our CAD-GCN, the recently proposed GCN [18] is employed as the backbone. As the extension of convolutional neural network (CNN) for nongrid data, GCN is able to aggregate features and propagate information across graph nodes. Consequently, the convolution operation of GCN is adaptively dominated by the neighborhood structure and can be applied to the non-Euclidean irregular data based on the graph which encodes contextual relations among the graph nodes. As a result, the complex regions such as target boundaries in HSI can be flexibly preserved by GCN. Meanwhile, through successively aggregating feature information based on the contextual relations, high-level features can be naturally extracted with GCN.

To capture the long-range dependencies, our proposed method learns to project the original HSI into a region-induced graph and encodes contextual relations among image regions, by which the receptive field of GCN will not be merely limited to a fairly small region. Then inference can be performed on the graph through passing messages between regions and along the edges connecting them. Therefore, this inference can not only update the region features, but also connect the regions which are originally far away in the 2-D space by successive graph convolutions. As a result, the long-range relations between faraway image regions can be effectively exploited. After that, the proposed CAD-GCN can learn an effective graph representation with only a small number of nodes. Finally, the learned region-level features can be interpolated into the 2-D feature map by reverting the pixel-to-region assignment from the previous graph projection step, so that the pixel-level features can be obtained to fully comply with the existing networks. With the above graph projection and reprojection framework, the formed regions as well as their features are flexibly learned by the network in an end-to-end way, by which the negative impact of inaccurate precomputed region features can be effectively rectified.

Furthermore, in the proposed CAD-GCN, we also enable the graph to be updated dynamically, to iteratively refine the contextual relations among regions. The update process can be divided into two parts, namely the refinement of node similarities and connective relationships, respectively. Considering that the predefined graph based on the Euclidean distance may not be suitable for measuring their intrinsic similarities [19], we intend to learn the improved similarity measurement. To be specific, the graph can be dynamically updated to adapt to the region representations generated by each graph convolutional layer, which will in turn produce the improved representations. In addition, we further refine the learned graph by introducing the “edge filter" which can filter out the incorrect interclass edges, since the graph may contain improper interclass connections, especially around the boundaries between different land covers.

It is noted that one previous work “multiscale dynamic GCN" (MDGCN) [20] also utilizes GCN for HSI classification. However, this article is very different from MDGCN in two aspects. First, in MDGCN, the regions are coarsely formed via using a heuristic superpixel generation technique, which might be imprecise and will not change throughout the classification process. In contrast, the regions in our CAD-GCN are adaptively learned by the projection and reprojection steps, so that they can well fit the object appearances in the image. Second, different from MDGCN which fails to refine the connective relationships among graph nodes, our

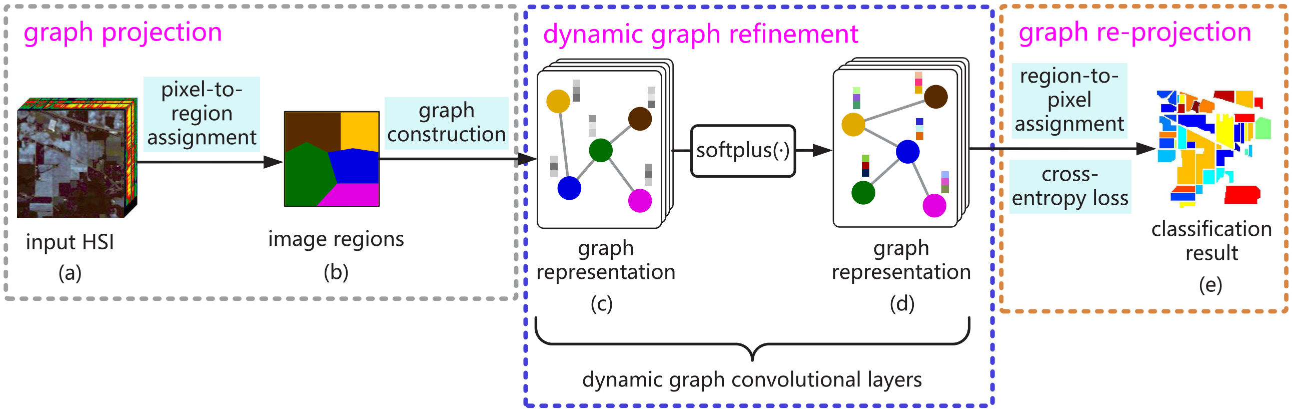


Fig. 1. Framework of our proposed CAD-GCN. (a) Original HSI. (b) Five regions from the original HSI obtained via pixel-to-region assignment. (c) and (d) Two dynamic graph convolutional layers, where the circles with different colors correspond to different image regions (i.e., graph nodes) and the gray lines represent graph edges. From (c) to (d), both the edge weight and the connective relationships among regions can be dynamically refined during the convolution operation, and thus the improved graph structure and node representations can be obtained. In our model, softplus [21] is utilized as the activation function. In (e), the learned graph representation can be interpolated back into 2-D image grids based on the region-to-pixel assignment, and then the cross-entropy loss is used to penalize the label differences between the network output and the originally labeled pixels.

CAD-GCN dynamically updates the graph edges connecting different image regions, so that the contextual relations can be exploited more properly.

To sum up, the proposed CAD-GCN employs three key techniques to effectively and precisely characterize contextual relations: 1) the incorporation of GCN for sufficiently exploiting contextual relations among pixels; 2) the employment of the flexible graph projection and reprojection framework for exploring long-range contextual relations and generating faithful region features; and 3) The utilization of dynamic graph refinement for accurately characterizing contextual relations and timely finding precise region representations. Intensive experimental results on four typically used data sets reveal the effectiveness of the proposed CAD-GCN.

II. Proposed Method

This section details the proposed CAD-GCN model, of which the pipeline is presented in Fig. 1. Given an input image [Fig. 1(a)], we first obtain its region features [Fig. 1(b)] by learning to project the original image with 2-D pixel grids into graph data. Then dynamic graph convolution [Fig. 1(c) and (d)] is conducted to refine the acquired region graph, along with encoding features for each region. Finally, the classification result [Fig. 1(e)] is produced by interpolating the learned graph representation into 2-D grids based on the region-to-pixel assignment. The critical operations in the proposed CAD-GCN will be detailed by presenting the GCN backbone (Section II-A), explaining the graph projection with pixel-to-region assignment (Section II-B), describing the dynamic graph refinement (Section II-C), and elaborating the graph reprojection with region-to-pixel assignment (Section II-D).

A. GCN

Inspired by CNN, GCN [18] is a multilayer neural network which directly operates on a graph and aims to extract high-level features through aggregating feature information from the neighborhoods of graph nodes. In GCN, an undirected graph is formally defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ and $\mathcal{E}$ denoting the sets of nodes and edges, respectively. The notation $\mathbf{A}$ denotes the adjacency matrix of $\mathcal{G}$ which indicates the existence of an edge between each pair of nodes, and its (i, j)th element can be calculated as

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

where the parameter γ is empirically set to 0.2 in our experiments, $\mathbf{x}_i$ and $\mathbf{x}_j$ represent two graph nodes (i.e., image regions in this article), and $N(\mathbf{x}_j)$ is the set of neighbors of $\mathbf{x}_j$.

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

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

where $\mathbf{U}$ is the matrix of eigenvectors of 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 denotes a diagonal matrix composed of the eigenvalues of $\mathbf{L}$, $\mathbf{D}$ is the degree matrix with the diagonal element $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, and $\mathbf{I}$ represents the identity matrix with proper size throughout this article. Then $g_{\boldsymbol{\theta}}$ can be understood as a function of eigenvalues of $\mathbf{L}$, i.e., $g_{\boldsymbol{\theta}}(\Lambda)$. To reduce the computational cost of eigendecomposition in (2), Hammond *et al.* [22] approximated $g_{\boldsymbol{\theta}}(\Lambda)$ using a truncated expansion in terms of Chebyshev polynomials $T_k(\mathbf{x})$ up to Kth order, namely

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

where $\boldsymbol{\theta}'$ denotes a vector of Chebyshev coefficients, and $\widetilde{\Lambda} = (2/\lambda_{\max})\Lambda - \mathbf{I}$ with $\lambda_{\max}$ being the largest eigenvalue of $\mathbf{L}$. According to [22], 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

$T_1(\mathbf{x}) = \mathbf{x}$. Hence, we have the convolution of a signal $\mathbf{x}$ 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 $\widetilde{\mathbf{L}} = (2/\lambda_{\max})\mathbf{L} - \mathbf{I}$ is the scaled Laplacian matrix. Equation (4) can be easily verified according to the fact that $(\mathbf{U}\Lambda\mathbf{U}^{\top})^k = \mathbf{U}\Lambda^k\mathbf{U}^{\top}$. As is can be observed, this expression is a Kth-order polynomial regarding the Laplacian (i.e., K-localized). In other words, the filtering only depends on the nodes that are at most K steps away from the central node. In our CAD-GCN model, the first-order neighborhood is considered, that is $K = 1$, and thus (4) turns to a linear function on the graph Laplacian spectrum with respect to $\mathbf{L}$.

Afterward, a neural network based on graph convolutions can be built by stacking multiple convolutional layers in the form of (4), where each layer is followed by an elementwise nonlinear operation (i.e., softplus$(\cdot)$ [21]). By this way, we can derive diverse classes of convolutional filter functions through stacking multiple layers of the same configuration. With the linear formulation, Kipf and Welling [18] further approximated $\lambda_{\max} \approx 2$, considering that the network parameters can adapt to this change in scale during the training process. Therefore, (4) is simplified to

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

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

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

by letting $\boldsymbol{\theta} = \boldsymbol{\theta}'_0 = -\boldsymbol{\theta}'_1$. As $\mathbf{I}+\mathbf{D}^{-(1/2)}\mathbf{A}\mathbf{D}^{-(1/2)}$ has the eigenvalues in the range $[0, 2]$, repeatedly applying this operator will result in numerical instabilities and exploding/vanishing gradients in a deep network. To solve this deficiency, Kipf and Welling [18] 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 then be expressed as

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

where $\mathbf{H}^{(l)}$ denotes the output of the lth layer, $\sigma(\cdot)$ represents an activation function, such as the softplus function [21] used in our proposed CAD-GCN, and $\mathbf{W}^{(l)}$ is the trainable weight matrix involved in the lth layer.

B. Pixel-to-Region Assignment

Although GCN is able to capture contextual relations among image pixels, the receptive field of pixel-level graph convolution is often limited [23]. To effectively characterize long-range relations among pixels, we intend to move beyond regular 2-D image grids and encode contextual relations among regions, since the dependencies among image regions are of much longer than those captured by pixel-level convolutions [23]. The main idea is learning pixel-to-region assignment which groups pixels with similar features into coherent regions, to capture contextual relations among the regions originally far away in the original 2-D space.

Different from the conventional region-based methods [13], [24] which start by coarsely grouping pixels into certain regions, we aim at learning to transform the original HSI into a region graph, and this process is called graph projection. Specifically, a soft assignment matrix which is parameterized by $\mathbf{V} \in \mathbb{R}^{d\times c}$ will be learned by the network to assign each pixel $\mathbf{z}_i \in \mathbb{R}^d$ to its neighboring regions, where d denotes the spectral dimensionality of each pixel, c is 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 region. Then the soft assignment matrix $\mathbf{P} \in \mathbb{R}^{n\times c}$ can be computed as

$$\mathbf{P}_{ij} = \begin{cases} e^{-\gamma\|\mathbf{z}_i - \mathbf{v}_j\|^2}, & \text{if } \mathbf{v}_j \in \widetilde{N}(\mathbf{z}_i) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where n is the number of image pixels, $\widetilde{N}(\mathbf{z}_i)$ denotes the set of neighboring regions connected to the pixel $\mathbf{z}_i$. To be more specific, $\widetilde{N}(\mathbf{z}_i)$ includes not only the central region where $\mathbf{z}_i$ resides, but also the regions adjacent that are to the central one. In (8), the element $\mathbf{P}_{ij}$ defines the soft assignment of a pixel $\mathbf{z}_i$ to $\mathbf{v}_j$. With the learned pixel-to-region assignment, the region feature $\mathbf{x}_j$ can be encoded by

$$\mathbf{x}_j = \frac{\sum_i \mathbf{P}_{ij}\mathbf{z}_i}{\sum_i \mathbf{P}_{ij}}. \tag{9}$$

By learning the features for each region, the negative impact of inaccurate precomputed region features can be reduced.

However, there still exists an optimization challenge, since most or even all of the image pixels may be assigned to a single region in some extreme circumstances. This is probably because the anchor point matrix for image regions $\mathbf{V}$ is initialized improperly, which will subsequently result in an ill-posed assignment matrix $\mathbf{P}$. Moreover, the imbalanced assignment will lead to unfavorable graph structure, and thus the contextual relations cannot be sufficiently exploited. To cope with this problem, instead of initializing $\mathbf{V}$ randomly, we take the spatial information into consideration and initialize $\mathbf{V}$ by utilizing a segmentation technique. Specifically, the simple linear iterative clustering (SLIC) algorithm [25], which has been widely used for image segmentation, is employed to obtain the initial regions. Herein, the average spectral signatures of the pixels involved in the corresponding region will be utilized to initialize each $\mathbf{v}_i$, and the matrix $\mathbf{V}$ can be further updated via using gradient descent. This segmentation-based initialization technique can yield more stable training performance and produce more meaningful graph representation than random initialization [23]. Fig. 2 exhibits the pixel-to-region assignment regarding a pixel $\mathbf{z}_i$. With the learned region features, the corresponding region graph can be naturally acquired using (1). After that, the region features $\mathbf{X}$ will be recomputed by performing graph convolution [18] which aggregates information along the edges. Moreover, through successive graph convolutions, long-range dependencies among the regions that are far away in the original 2-D space can be captured.

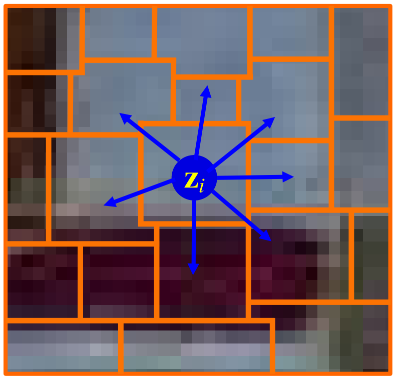


Fig. 2. Illustration of the soft pixel-to-region assignment used in our CAD-GCN model. Each of the initialized image regions are surrounded by yellow lines, and the blue arrows denote the assignment regarding the pixel $\mathbf{z}_i$ to its neighboring regions.

C. Dynamic Graph Refinement

The performance of graph convolution largely depends on the quality of the predefined graph which encodes the similarities and connective relationships among graph nodes. However, the Euclidean distance, which is widely used for characterizing node similarities [e.g., in (1)], may not be a good metric for graph structured data [19]. To address this weakness, we aim to learn an improved distance metric. Specifically, we construct a symmetric positive semidefinite matrix $\mathbf{M} = \mathbf{W}_d\mathbf{W}_d^\top$ with $\mathbf{W}_d$ being a trainable weight matrix. Then the generalized Mahalanobis distance can be formulated as follows:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)}. \tag{10}$$

Afterward, the adjacency matrix $\mathbf{A}$ in (1) can be rewritten as

$$\mathbf{A}_{ij} = \begin{cases} e^{-\gamma(\mathcal{D}(\mathbf{x}_i,\mathbf{x}_j))^2}, & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Since the graph representation is updated along with the graph convolutional layers, learning a single matrix $\mathbf{M}$ is insufficient to accurately measure node similarities for all the layers. Therefore, we adaptively learn the symmetric positive semidefinite parameter matrix $\mathbf{M}^{(l)}$ for the adjacency matrix $\mathbf{A}^{(l)}$ which is utilized in the lth layer, to acquire the improved node similarities. Then (11) can be rewritten as

$$\mathbf{A}_{ij}^{(l+1)} = \begin{cases} e^{-\gamma(\mathcal{D}^{(l)}(\mathbf{h}_i^{(l)},\mathbf{h}_j^{(l)}))^2}, & \text{if } \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $\mathbf{h}_i^{(l)}$ is the representation of $\mathbf{x}_i$ generated by the lth layer with $\mathbf{h}_i^{(0)} = \mathbf{x}_i$, and $\mathcal{D}^{(l)}(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)})$ can be formulated as $((\mathbf{h}_i^{(l)} - \mathbf{h}_j^{(l)})^\top \mathbf{M}^{(l)}(\mathbf{h}_i^{(l)} - \mathbf{h}_j^{(l)}))^{1/2}$.

During graph construction, connections among the regions from different classes may be incorporated, which will lead to the aggregation of interclass feature information and further degrade the discriminability of graph convolution results. To overcome this deficiency, we propose to use the edge filter, which aims to refine the contextual relations by reducing undesirable interclass edges of the graph. Since the intraclass examples are generally more similar than the interclass ones, it is believed that the element $\mathbf{A}_{ij}^{(l)}$ with relatively small value is more likely to represent interclass relations than the $\mathbf{A}_{ij}^{(l)}$ with large value. Therefore, we employ a threshold $\beta^{(l)}$ for each graph convolutional layer to filter out the interclass relations and reduce the adverse effect of interclass feature aggregation. The selection of $\beta^{(l)}$ will be discussed in Section III. Specifically, in the lth layer, the edge filter $\mathcal{F}(\cdot)$ used can be simply expressed as

$$\mathcal{F}(\mathbf{A}_{ij}^{(l)}) = \begin{cases} \mathbf{A}_{ij}^{(l)}, & \text{if } \mathcal{F}(\mathbf{A}_{ij}^{(l)}) > \beta^{(l)} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

In practice, constraining the number of parameters can be beneficial to address the problem of overfitting [18], and thus we set $\beta^{(l)} = \beta$ for all the layers. With the edge filter, the graph convolutional layer can then be reformulated as

$$\mathbf{H}^{(l)} = \sigma(\mathcal{F}(\mathbf{A}^{(l)})\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}) \tag{14}$$

with $\mathbf{H}^{(0)} = \mathbf{X}$.

D. Region-to-Pixel Assignment

After conducting the dynamic graph convolution on the region level, we need to reproject the new region features (i.e., the learned graph representation) $\mathbf{H}^{(L)}$ back into 2-D image with grids of pixels, and this process is called graph reprojection. Specifically, the region-to-pixel assignment is accomplished by linearly interpolating pixel features based on the soft assignment matrix $\mathbf{P}$, namely $\mathbf{P}\mathbf{H}^{(L)}$, where L denotes the number of graph convolutional layers. It is noted that all the reprojected pixels will have diverse feature representations, even if some of them are assigned to the same region. Therefore, the contextual details of the HSI can be well preserved.

With the region-to-pixel assignment, the output of our proposed CAD-GCN can be obtained as

$$\mathbf{O} = \mathbf{P}\mathbf{H}^{(L)}. \tag{15}$$

In our CAD-GCN model, the cross-entropy error is employed to penalize the differences between the network output and the labels of labeled pixels, namely

$$\mathcal{L} = -\sum_{g \in \mathbf{y}_G}\sum_{f=1}^{C} \mathbf{Y}_{gf} \ln \mathbf{O}_{gf} \tag{16}$$

where C is the number of classes, $\mathbf{y}_G$ denotes the set of indices corresponding to the labeled pixels, and $\mathbf{Y}$ represents the label matrix. Herein, we let $\mathbf{Y}_{ij}$ be 1 if the pixel $\mathbf{z}_i$ belongs to the jth class, and 0 otherwise. It is noticeable that our model can be trained via an end-to-end way. Similar to [18], full-batch gradient descent is utilized to update the network parameters for CAD-GCN. Algorithm 1 shows the summarization of our proposed CAD-GCN classification method.

III. Experimental Results

To test the effectiveness of the proposed CAD-GCN model, in this section, we conduct exhaustive experiments on four real-world benchmark data sets, namely Indian Pines, University of Pavia, and Salinas. We first compare CAD-GCN with

Algorithm 1 Proposed CAD-GCN for HSI Classification

```
Input: Input image; number of iterations $\mathcal{T}$; learning rate $\eta$;
    number of graph convolutional layers $L$;
Initialize the anchor point matrix $\mathbf{V}$ with SLIC algorithm;
// Train the CAD-GCN model
for $t = 1$ to $\mathcal{T}$ do
    Learn the region features $\mathbf{X}$ through Eq. (8) and Eq. (9);
    Dynamically refine the graph $\mathbf{A}^{(l)}$ using Eq. (12) and
        Eq. (13) along with the graph convolution operation
        of Eq. (14);
    Interpolate the region features back into the original 2D
        grids by Eq. (15);
    Calculate the error term according to Eq. (16), and
        update the weight matrices $\mathbf{W}^{(l)}$ $(1 \le l \le L)$ using
        full-batch gradient descent;
end for
Conduct label prediction via Eq. (14) and Eq. (15);
Output: Predicted label for each pixel.
```

other state-of-the-art methods, where four metrics including per-class accuracy, overall accuracy (OA), average accuracy (AA), and kappa coefficient, are used to evaluate the model performance. Then we study the influence of the number of labeled pixels on the classification performance. After that, we investigate the impact of hyperparameters incorporated by the proposed CAD-GCN. Finally, we present the ablation study and also investigate the running time of our model.

A. Data Sets

The performance of our proposed CAD-GCN is evaluated on four real-world benchmark data sets, i.e., the Indian Pines,[1] the University of Pavia,[2] the Salinas,[3] and the Houston University,[4] which will be introduced in the following.

1) Indian Pines: The Indian Pines data set was gathered by Airborne Visible/Infrared Imaging Spectrometer sensor in 1992, which records north-western India. This data set consists of 145 × 145 pixels with a spatial resolution of 20 m × 20 m, and there are 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 the remaining 200 bands are retained. The original ground truth of the Indian Pines data set includes 16 land-cover classes, such as “Alfalfa," “Corn-notill," “Corn-mintill," and so on. Fig. 3 exhibits 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 captures the Pavia University of Italy with the ROSIS sensor. This data set consists of 610 × 340 pixels with a spatial resolution of 1.3 m × 1.3 m and has 103 spectral channels in the wavelength ranging from 0.43 to 0.86 μm after removing noisy bands. The University of Pavia data set includes nine land-cover classes, such as “Asphalt," “Meadows," “Gravel," and so on, which are displayed in Fig. 4. Table II shows the amounts of labeled and unlabeled pixels of each class.

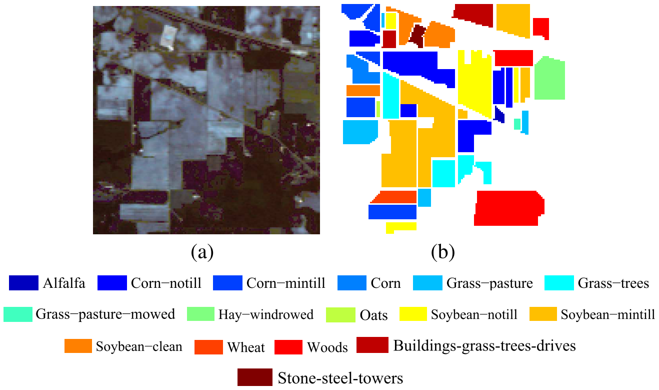


Fig. 3. Indian Pines. (a) False color image. (b) Ground truth map.

TABLE I

NUMBERS OF LABELED AND UNLABELED PIXELS OF ALL CLASSES IN 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

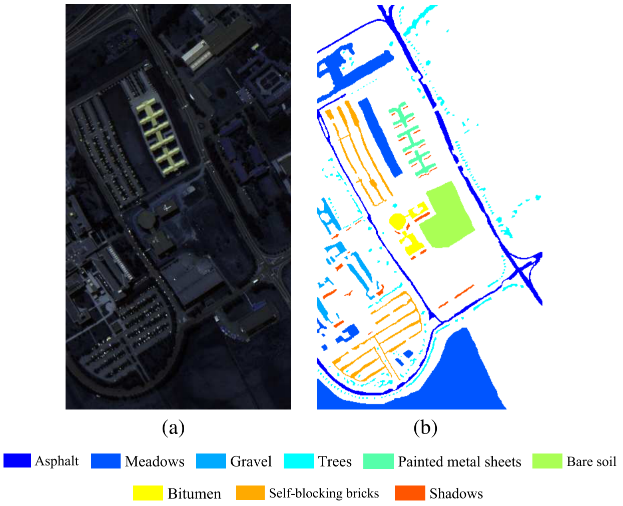


Fig. 4. University of Pavia. (a) False color image. (b) Ground truth map.

[1] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Indian_Pines

[2] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes# Pavia_University_scene

[3] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Salinas_scene

[4] http://www.grss-ieee.org/community/technical-committees/data-fusion/

TABLE II

NUMBERS OF LABELED AND UNLABELED PIXELS OF ALL CLASSES IN 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

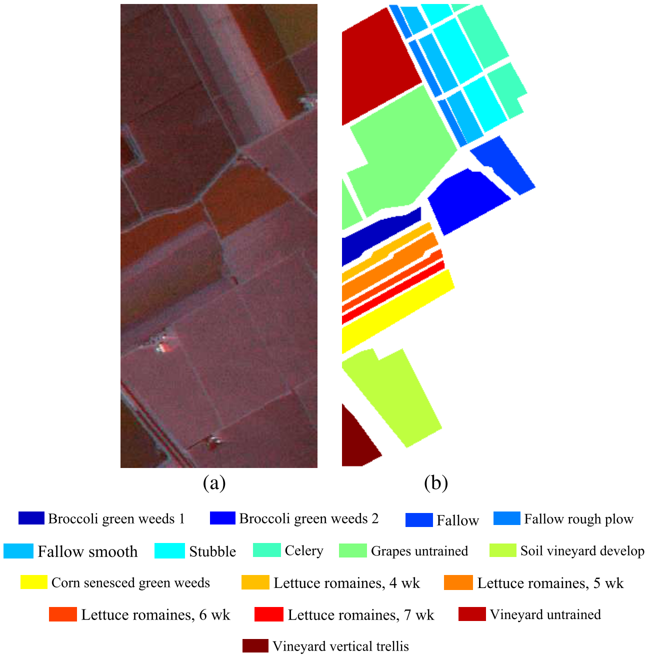


Fig. 5. Salinas. (a) False color image. (b) Ground truth map.

3) Salinas: The Salinas data set is another classic HSI which is collected by the AVIRIS sensor over Salinas Valley, CA, USA. This data set comprises 204 spectral bands (20 water absorption bands are removed) and 512 × 217 pixels with a spatial resolution of 3.7 m. The Salinas data set contains 16 land-cover classes, such as "Fallow," "Stubble," "Celery," and so on. Fig. 5 exhibits the false color image and ground truth map of the Salinas data set. The numbers of labeled and unlabeled pixels of different classes are listed in Table III.

4) Houston University: The Houston University data set, which has been used in the 2013 GRSS Data Fusion Contest, was collected by the NSF-funded Center for Airborne Laser Mapping over the Houston University campus and its neighboring areas. This data set contains 349 × 1905 pixels with a spatial resolution of 2.5 m and 144 spectral bands in the range of 380–1050 nm. There are 15 land-cover classes in the image, including "Healthy grass," "Water," "Running Track," and so on, which are displayed in Fig. 6. Table IV exhibits the numbers of labeled and unlabeled pixels of each class.

TABLE III

NUMBERS OF LABELED AND UNLABELED PIXELS OF ALL CLASSES IN SALINAS DATA SET

ID	Class	#Labeled	#Unlabeled
1	Broccoli green weeds 1	30	1979
2	Broccoli green weeds 2	30	3696
3	Fallow	30	1946
4	Fallow rough plow	30	1364
5	Fallow smooth	30	2648
6	Stubble	30	3929
7	Celery	30	3549
8	Grapes untrained	30	11241
9	Soil vineyard develop	30	6173
10	Corn senesced green weeds	30	3248
11	Lettuce romaines, 4 wk	30	1038
12	Lettuce romaines, 5 wk	30	1897
13	Lettuce romaines, 6 wk	30	886
14	Lettuce romaines, 7 wk	30	1040
15	Vineyard untrained	30	7238
16	Vineyard vertical trellis	30	1777

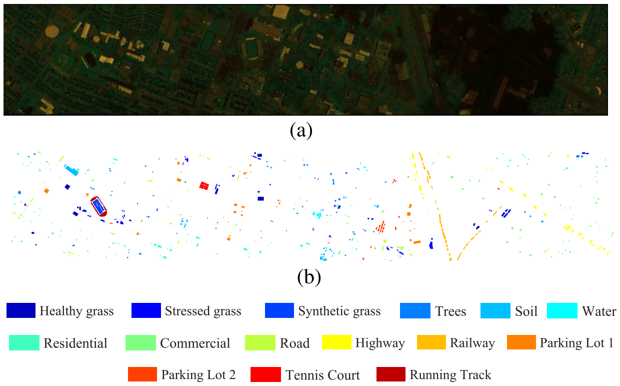


Fig. 6. Houston University. (a) False color image. (b) Ground truth map.

TABLE IV

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

No.	Class	#Labeled	#Unlabeled
1	Healthy grass	30	1344
2	Stressed grass	30	1424
3	Synthetic grass	30	730
4	Trees	30	1234
5	Soil	30	1268
6	Water	30	295
7	Residential	30	1446
8	Commercial	30	1324
9	Road	30	1524
10	Highway	30	1394
11	Railway	30	1483
12	Parking Lot 1	30	1399
13	Parking Lot 2	30	540
14	Tennis Court	30	451
15	Running Track	30	728

B. Experimental Settings

In our experiments, the proposed CAD-GCN algorithm is implemented by TensorFlow with Adam optimizer. For all the adopted four data sets mentioned in Section III-A, usually 30 labeled pixels (i.e., examples) per class are randomly chosen for training, and 15 labeled pixels are chosen if the

TABLE V

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

ID	GCN [18]	S^{2}GCN [26]	MDGCN [20]	CNN-PPF [27]	MBCTU [15]	JSDF [28]	CAD-GCN
1	95.00±2.80	**100.00±0.00**	**100.00±0.00**	95.00±2.64	93.06±7.99	**100.00±0.00**	99.46±1.54
2	56.71±4.42	84.43±2.50	80.18±0.84	73.53±5.61	72.79±6.71	**90.75±3.19**	88.08±4.57
3	51.50±2.56	82.87±5.53	**98.26±0.00**	81.34±3.76	84.32±5.38	77.84±3.81	95.62±1.89
4	84.64±3.16	93.08±1.95	98.57±0.00	91.84±3.53	93.18±3.58	**99.86±0.33**	97.85±2.75
5	83.71±3.20	**97.13±1.34**	95.14±0.33	93.69±0.84	90.65±3.16	87.20±2.73	93.79±3.71
6	94.03±2.11	97.29±1.27	97.16±0.57	97.46±1.01	95.04±3.23	**98.54±0.28**	96.41±1.78
7	92.31±0.00	92.31±0.00	**100.00±0.00**	75.38±8.73	91.45±9.83	**100.00±0.00**	97.95±2.84
8	96.61±1.86	99.03±0.93	98.89±0.00	98.01±0.69	96.99±1.70	99.80±0.31	**99.81±0.25**
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**
10	77.47±1.24	93.77±3.72	**90.02±1.02**	82.30±1.55	81.66±5.97	89.99±4.24	89.34±4.03
11	56.56±1.53	84.98±2.82	**93.35±1.47**	62.64±3.32	70.33±4.10	76.75±5.12	93.09±3.14
12	58.29±6.58	80.05±5.17	93.05±2.30	88.92±2.50	83.31±5.97	87.10±2.82	**93.41±2.54**
13	**100.00±0.00**	99.43±0.00	**100.00±0.00**	98.80±0.57	96.89±2.70	99.89±0.36	99.79±0.29
14	80.03±3.93	96.73±0.92	**99.72±0.05**	86.49±2.23	92.85±2.39	97.21±2.78	99.36±0.54
15	69.55±6.66	86.80±3.42	**99.72±0.00**	86.71±4.36	90.11±5.01	99.58±0.68	99.20±0.87
16	98.41±0.00	**100.00±0.00**	95.71±0.00	92.70±3.45	95.77±2.93	**100.00±0.00**	98.91±1.01
OA	69.24±1.56	89.49±1.08	93.47±0.38	80.09±1.56	82.34±1.06	88.34±1.39	**94.13±0.78**
AA	80.93±1.71	92.99±1.04	96.24±0.21	87.80±1.53	89.27±0.78	94.03±0.55	**96.38±0.35**
Kappa	65.27±1.80	88.00±1.23	92.55±0.43	77.52±1.74	79.98±1.18	86.80±1.55	**93.29±0.88**

corresponding class contains less than 30 pixels. The remaining pixels in each class are regarded as unlabeled examples during the training process and will be used as the test set to evaluate the classification performance afterward. 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.

To evaluate the classification ability of our proposed CAD-GCN, other recent state-of-the-art HSI classification methods are also utilized for comparison. Specifically, we employ three GCN-based methods, i.e., GCN [18], spectral–spatial GCN (S^{2}GCN) [26], and MDGCN [20], together with one CNN-based methods such as CNN-pixel-pair features (CNN-PPFs) [27]. Meanwhile, we also compare the proposed CAD-GCN with two traditional HSI classification methods, namely multiband compact texture unit (MBCTU) [15] and multiple feature learning (MFL) [29], respectively. All these methods are implemented ten times with different labeled pixels on each hyperspectral data set, and the mean accuracies together with the standard deviations over these ten independent implementations are reported.

C. Classification Results

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

1) Results on the Indian Pines Data Set: The quantitative results acquired by different methods on the Indian Pines data set are presented in Table V, and the highest record regarding each class (i.e., each row) has been highlighted in bold. As shown in Table V, the classical HSI classification methods (i.e., MBCTU and JSDF) outperform GCN by a substantial margin, which confirms the effectiveness of spatial context. We also see that our proposed CAD-GCN achieves the top-level performance among all the methods in terms of OA, AA, and Kappa coefficient, and the standard deviations are very small as well. Meanwhile, the proposed CAD-GCN acquires stable and very high classification accuracies on most of the land-cover classes. All these statistics demonstrate the effectiveness of our CAD-GCN in HSI classification.

The classification maps generated by different methods on the Indian Pines data set are exhibited in Fig. 7. To facilitate the comparison among the investigated methods, the ground truth map is also provided in Fig. 7(a). A visual inspection reveals that the proposed CAD-GCN method produces a much more compact classification map and shows fewer misclassifications than other methods. More concretely, in the classification maps of GCN, S^{2}GCN, and CNN-PPF, the errors are almost uniformly distributed (the salt-and-pepper effect in the homogeneous regions), while in the classification maps of MDGCN and our proposed CAD-GCN, the errors only appear in some highly heterogeneous areas, where the spatial separability between classes is quite low. For instance, in the classification maps obtained by GCN, S^{2}GCN, and CNN-PPF, the middle and the bottom left parts of the classification maps which correspond to “Soybean-mintill" are highly confusing. Moreover, by comparing CAD-GCN with JSDF, we can also find that JSDF produces more errors around class boundaries than our CAD-GCN method, which reveals the good discriminability of the proposed CAD-GCN in boundary regions.

2) Results on the University of Pavia Data Set: In Table VI, different methods are compared on the aforementioned four data sets, where per-class accuracy, OA, AA, and Kappa coefficient are reported, and the best result in each row is highlighted in bold. From Table VI, we can conclude that the classification performance of our proposed CAD-GCN method is superior to the competitors in terms of OA, AA, and Kappa coefficient except MDGCN. In can be inferred that the incorporation of multiscale cues enables MDGCN to flexibly capture the variations of contextual distribution around objects. Therefore, MDGCN is able to effectively perceive the

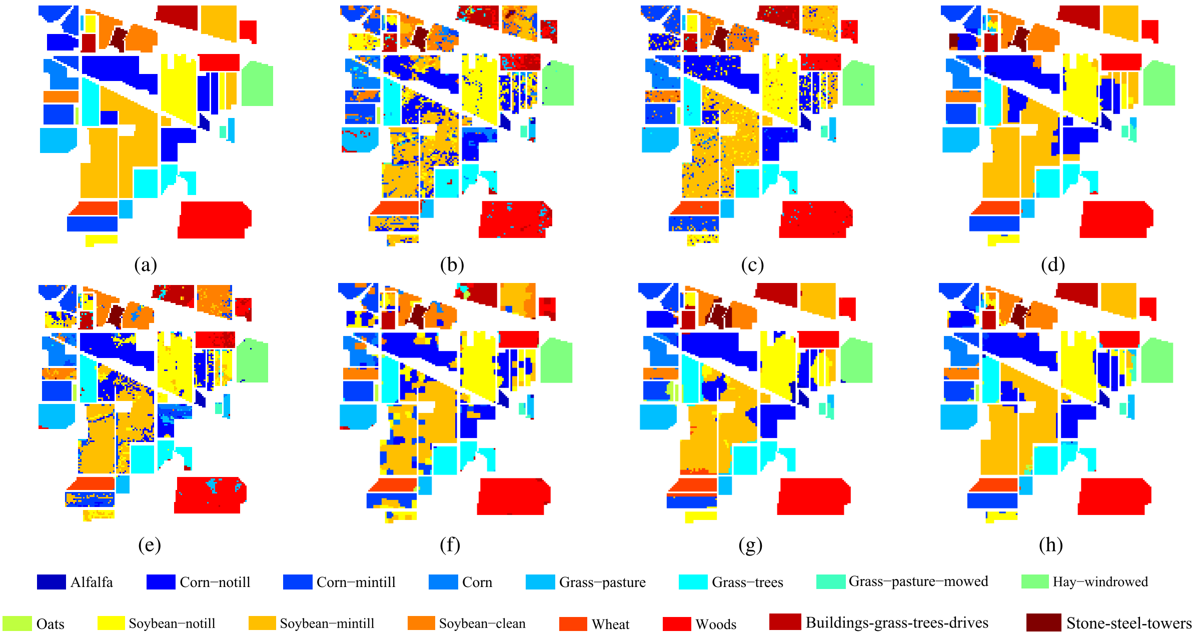

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 7. Classification maps obtained by different methods on Indian Pines data set. (a) Ground truth map. (b) GCN. (c) S^{2}GCN. (d) MDGCN. (e) CNN-PPF. (f) MBCTU. (g) JSDF. (h) CAD-GCN.

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

ID	GCN [18]	S^{2}GCN [26]	MDGCN [20]	CNN-PPF [27]	MBCTU [15]	JSDF [28]	CAD-GCN
1	69.78±4.71	92.87±3.79	93.55±0.37	**95.73±0.80**	87.49±3.99	82.40±4.07	83.85±4.80
2	54.10±10.54	87.06±4.47	**99.25±0.23**	84.01±1.99	89.11±5.58	90.76±3.74	95.56±1.97
3	69.69±4.48	87.97±4.77	92.03±0.24	86.45±1.94	86.24±4.23	86.71±4.14	**95.50±2.41**
4	91.23±7.02	90.85±0.94	83.78±1.55	91.70±2.06	90.61±3.39	**92.88±2.16**	81.90±3.51
5	98.74±0.11	**100.00±0.00**	99.47±0.09	99.93±0.04	97.18±1.18	**100.00±0.00**	98.91±1.06
6	65.34±10.53	88.69±2.64	95.26±0.50	93.57±1.28	93.25±2.93	94.30±4.55	**97.81±2.89**
7	86.64±4.68	98.88±1.08	**98.92±1.04**	93.53±0.72	93.49±2.47	96.62±1.37	96.79±2.42
8	72.26±2.63	89.97±3.28	94.99±1.33	83.83±1.60	84.14±4.78	94.69±3.74	**95.35±2.76**
9	**99.93±0.06**	98.89±0.53	81.03±0.49	99.47±0.34	96.57±1.22	99.56±0.36	84.80±5.33
OA	66.19±3.43	89.74±1.70	**95.68±0.22**	88.72±0.95	89.43±2.14	90.82±1.30	92.91±1.01
AA	78.63±1.23	92.80±0.47	**93.15±0.28**	92.02±0.37	90.90±0.89	93.10±0.65	92.27±0.94
Kappa	58.39±3.28	86.65±2.06	**94.25±0.29**	85.43±1.18	86.24±2.62	88.02±1.62	90.69±1.29

irregular regions in the University of Pavia data set. However, the employment of multiscale contextual information also makes MDGCN time consuming, which will be presented in Section III-G. Differently, our CAD-GCN can achieve faithful classification ability without utilizing information at different scales. Specifically, compared with CNN-based method (i.e., CNN-PPF), the proposed CAD-GCN increases the OA by 4.19%, which suggests that the refined contextual relations captured by our CAD-GCN is superior to the contextual information characterized by the fixed convolutional kernels of CNN.

Fig. 8 visualizes the classification results generated by the seven different methods on the University of Pavia data set. As depicted in Fig. 8(h), the classification map of our proposed CAD-GCN are noticeably closer to the ground truth map [see Fig. 8(a)] than other methods except MDGCN, which is consistent with previous results in Table VI. Although GCN and S^{2}GCN are able to capture the relations among graph nodes, they are not originally designed for accurately encoding the contextual relations of HSI. Different from these two methods, our CAD-GCN employs graph projection and dynamic graph refinement operations to effectively exploit the improved contextual relations. As a result, GCN and S^{2}GCN which use the fixed coarse graph convolution produce more errors than CAD-GCN.

3) Results on the Salinas Data Set: Table VII presents the experimental results of different methods on the Salinas data set. The proposed CAD-GCN is obviously superior to the CNN-based method (i.e., CNN-PPF) and all the other competitors. For instance, in Table VII, CAD-GCN yields approximately 8% higher OA than CNN-PPF. Especially in some classes such as “Grapes untrained" (ID = 8) and “Vineyard untrained" (ID = 15), the class-specific accuracies of our proposed CAD-GCN are even approximately 20% higher than those of the CNN-PPF. We also note that MBCTU outperforms S^{2}GCN, which can be inferred that the texture information extracted by MBCTU is more powerful than the contextual relations explored by S^{2}GCN in distinguishing the

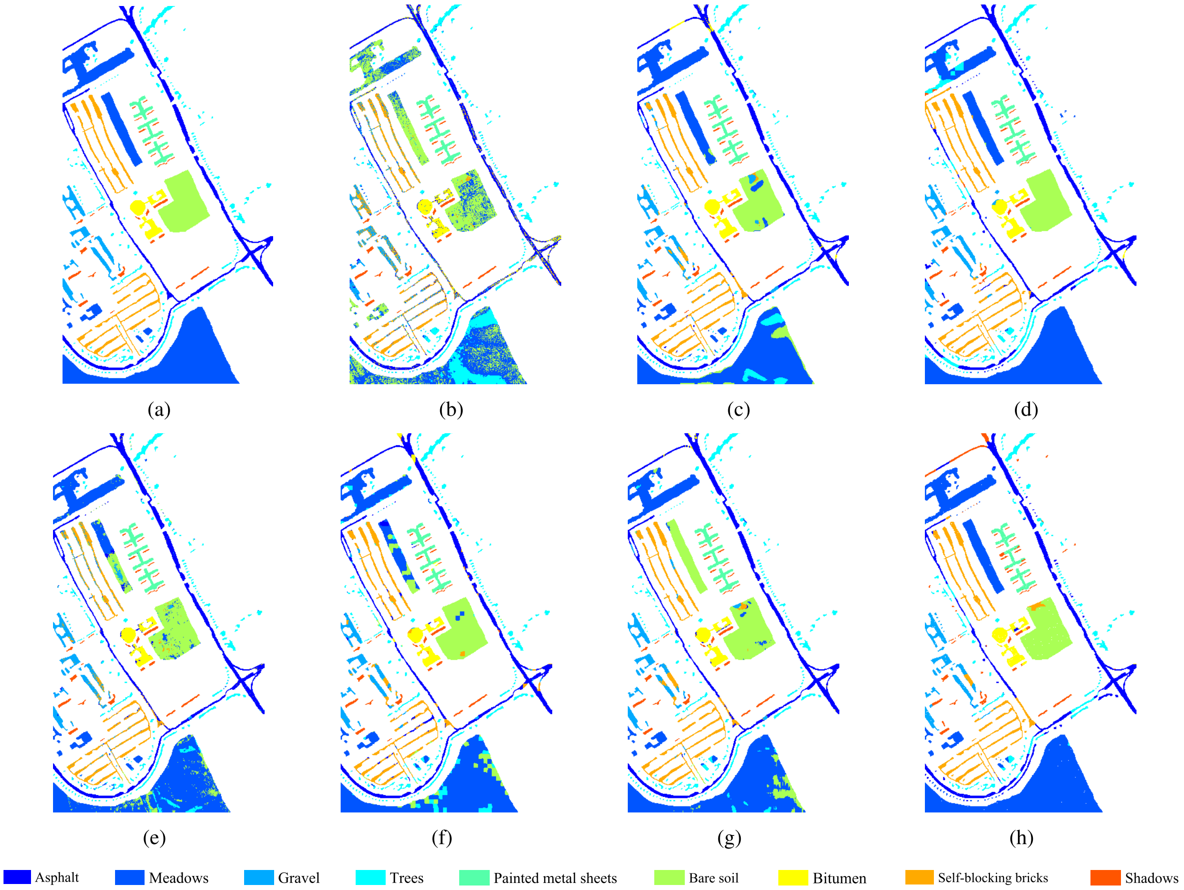


Fig. 8. Classification maps obtained by different methods on University of Pavia data set. (a) Ground truth map. (b) GCN. (c) S^{2}GCN. (d) MDGCN. (e) CNN-PPF. (f) MBCTU. (g) JSDF. (h) CAD-GCN.

TABLE VII

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

ID	GCN [18]	S^{2}GCN [26]	MDGCN [20]	CNN-PPF [27]	MBCTU [15]	JSDF [28]	CAD-GCN
1	98.62±0.86	99.01±0.44	99.98±0.03	99.77±0.21	99.18±0.80	**100.00±0.00**	**100.00±0.00**
2	99.07±1.21	99.18±0.59	99.90±0.28	98.69±0.89	99.76±0.33	**100.00±0.00**	**100.00±0.00**
3	97.03±1.10	97.15±2.76	99.80±0.21	99.50±0.49	99.13±1.04	**100.00±0.00**	99.97±0.07
4	99.28±0.49	99.11±0.55	97.49±2.16	99.81±0.04	97.61±0.82	**99.93±0.09**	98.46±1.48
5	98.58±0.79	97.55±2.35	97.96±0.77	96.64±1.26	96.54±1.01	**99.77±0.31**	97.77±1.05
6	99.58±0.30	99.32±0.35	99.10±1.67	99.32±0.86	99.74±0.32	**100.00±0.00**	99.61±0.33
7	99.13±0.25	99.06±0.27	98.18±1.49	99.59±0.13	98.26±1.64	**99.99±0.01**	99.10±1.56
8	67.94±8.33	70.68±5.20	92.78±4.61	74.77±4.01	81.98±4.32	87.79±4.89	**95.80±3.17**
9	98.50±0.85	98.32±1.79	**100.00±0.00**	98.99±0.18	99.47±0.51	99.67±0.33	**100.00±0.00**
10	89.64±1.57	90.97±2.59	**98.31±1.29**	89.32±3.04	92.21±2.75	96.53±2.55	98.01±1.40
11	94.80±2.98	98.00±1.65	99.39±0.55	97.65±1.49	96.24±2.68	99.76±0.21	**99.82±0.27**
12	99.71±0.08	99.56±0.59	99.01±0.78	99.82±0.30	98.98±0.45	**100.00±0.00**	97.99±0.78
13	97.99±0.61	97.83±0.72	97.59±1.32	97.70±0.50	96.73±1.66	**100.00±0.00**	97.90±0.98
14	93.58±2.60	95.75±1.65	97.92±1.72	94.14±1.22	96.50±3.05	98.71±0.72	**98.89±1.53**
15	66.18±9.08	70.36±3.62	95.71±4.57	79.12±1.99	79.41±5.67	81.86±5.26	**97.47±1.65**
16	97.24±1.21	96.90±1.97	98.18±2.92	98.65±0.31	96.89±2.19	98.99±0.63	**99.76±0.57**
OA	87.16±0.85	88.39±1.01	97.25±0.87	90.52±0.77	92.14±0.86	94.67±0.77	**98.28±0.54**
AA	93.55±0.39	94.30±0.47	98.21±0.30	95.22±0.34	95.54±0.56	97.69±0.34	**98.78±0.22**
Kappa	85.74±0.92	87.10±1.12	96.94±0.96	89.46±0.85	91.25±0.95	94.06±0.85	**98.09±0.60**

land covers of the Salinas data set. By contrast, the improved contextual relations captured by our CAD-GCN still show their advantage on this data set.

Fig. 9 provides a visual comparison of the classification results obtained by different methods. It is observable that some areas in the classification map of our proposed CAD-GCN are less noisy than those of other methods, e.g., the regions of "Grapes untrained" and "Vineyard untrained," which is in consistence with the results listed in Table VII.

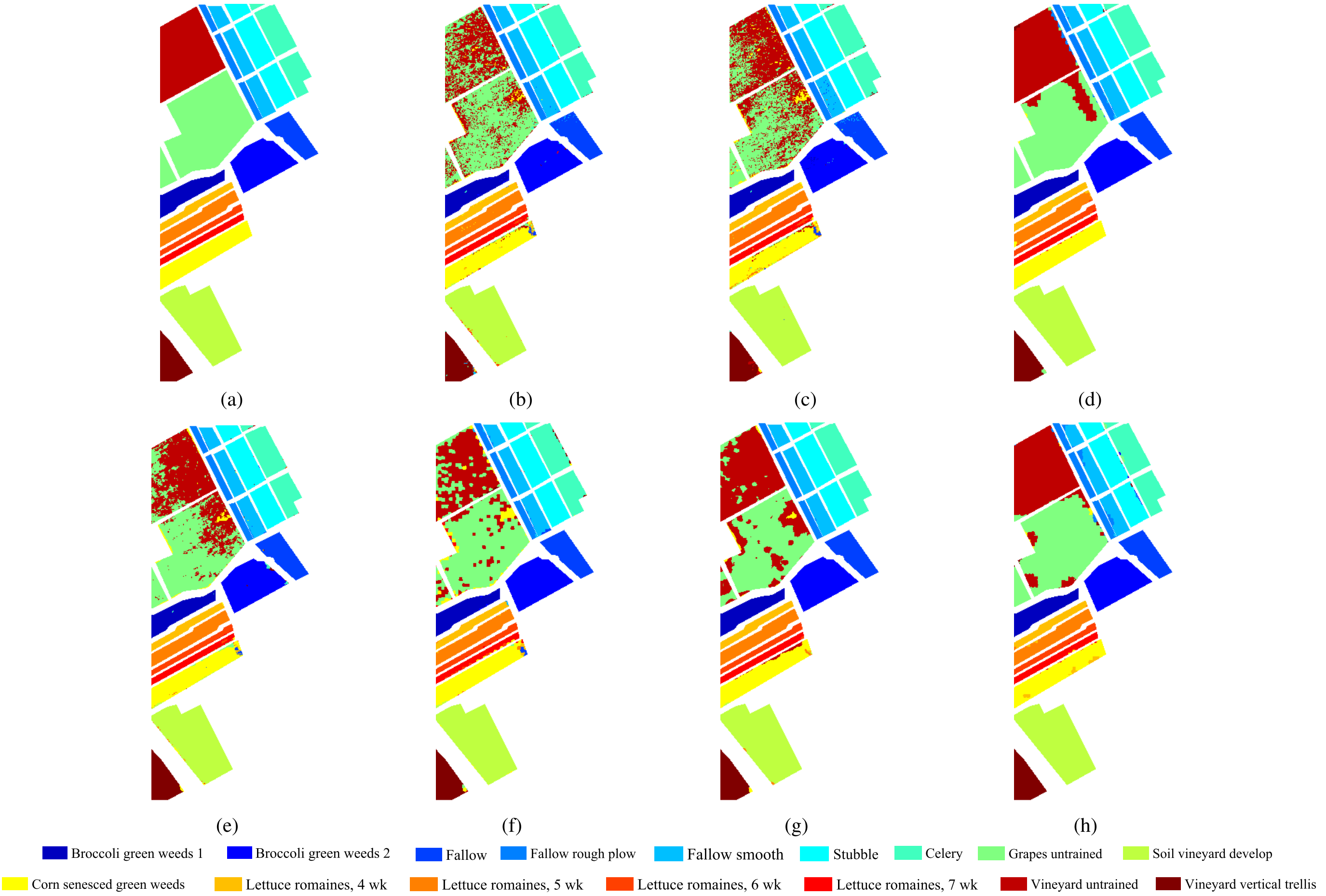

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 9. Classification maps obtained by different methods on Salinas data set. (a) Ground truth map. (b) GCN. (c) S^{2}GCN. (d) MDGCN. (e) CNN-PPF. (f) MBCTU. (g) JSDF. (h) CAD-GCN.

4) Results on the Houston University Data Set: Table VIII summarizes the classification results of different methods on the Houston University data set. As can be observed, the proposed CAD-GCN achieves the best classification result in terms of three quantitative criteria, namely OA, AA, and Kappa coefficient. Compared with the proposed CAD-GCN, we can see that CNN-PPF achieves slightly higher accuracies in five land cover classes, but it has very poor performance in several other classes, such as “Railway" (ID = 11), “Parking Lot 1" (ID = 12), and “Parking Lot 2" (ID = 13). It can be inferred that the CNN-based methods perform well in homogeneous regions, but they cannot precisely perceive the region boundaries due to the fixed convolutional kernels. Another notable fact is that our CAD-GCN outperforms other GCN-based method (i.e., GCN, S^{2}GCN, and MDGCN) in ten land-cover classes, which reveals the effectiveness of graph projection and graph refinement.

The visual performance comparison of the seven different methods on the Houston University data set is presented in Fig. 10. As can be seen, there often exist noticeable errors in the classification maps of the competitors (see the zoomed-in regions of Fig. 10). By contrast, our proposed CAD-GCN achieves the best visual classification result among all the methods, which confirms the advantage of our CAD-GCN.

D. Impact of the Number of Labeled Examples

In this experiment, the classification performances of the aforementioned seven methods with different numbers of labeled examples (i.e., pixels) for training are investigated. To be specific, we vary the number of labeled examples per class from 5 to 30 with an interval of 5, and report the OA acquired by all the methods on the Indian Pines, the University of Pavia, the Salinas, and the Houston University data sets (see Fig. 11). From the results, we can find that the proposed CAD-GCN generally outperforms the GCN, S^{2}GCN, MDGCN and all the other competitors on three data sets (namely the Indian Pines, the Salinas, and the Houston University data sets), which verifies the effectiveness of contextual relations captured by CAD-GCN. Meanwhile, the performance of our CAD-GCN is also comparable to MDGCN on the University of Pavia data set. Besides, we see that the performance of S^{2}GCN is unstable, since even the classical HSI classification method (namely MBCTU) can obtain better results than S^{2}GCN on the Salinas data set. Another interesting observation is that even if the labeled examples are quite limited (i.e., five or ten labeled examples per class), our CAD-GCN still achieves relatively high OA, which suggests good stability of CAD-GCN in HSI classification tasks.

TABLE VIII

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

ID	GCN [18]	S^2GCN [26]	MDGCN [20]	CNN-PPF [27]	MBCTU [15]	JSDF [28]	CAD-GCN
1	88.16±1.90	96.30±3.07	93.42±4.25	**98.62±0.71**	92.86±3.83	97.41±1.21	94.45±3.49
2	97.20±0.48	98.57±1.47	93.67±3.60	98.15±0.53	92.18±2.79	**99.48±0.25**	96.43±2.83
3	97.91±0.13	98.88±0.43	98.12±1.09	99.01±0.33	97.42±1.19	**99.88±0.22**	95.17±4.11
4	96.55±0.41	97.68±2.89	95.58±1.85	93.21±0.48	90.96±1.98	**98.22±2.80**	94.82±2.38
5	89.79±0.71	97.66±1.12	99.00±1.30	99.13±0.73	97.17±1.29	**100.00±0.00**	98.91±1.51
6	98.21±1.15	96.84±1.17	93.28±6.08	91.26±5.25	91.78±3.22	**99.32±1.09**	97.48±3.48
7	73.67±1.94	83.48±5.89	87.68±4.41	81.54±5.84	82.88±3.81	**91.93±4.91**	91.58±3.16
8	65.71±4.64	76.15±4.37	**80.45±6.12**	68.15±3.97	71.85±5.64	68.82±6.16	74.63±4.82
9	70.27±3.03	82.17±1.78	**89.64±2.26**	77.17±1.65	81.94±4.25	69.47±8.56	86.75±3.58
10	74.71±2.32	86.85±8.32	90.06±6.41	92.12±1.76	87.31±5.08	85.63±9.32	**94.23±3.34**
11	75.36±2.37	88.57±5.06	86.73±3.22	81.05±3.31	77.41±6.46	94.51±3.82	**94.65±2.73**
12	79.29±4.80	78.64±4.79	89.44±5.69	78.10±5.07	86.35±5.85	84.33±5.33	**89.55±1.93**
13	12.09±2.68	75.62±6.93	92.78±4.45	72.55±4.36	85.58±5.35	**98.10±1.28**	96.80±3.68
14	86.03±3.31	99.45±0.44	99.43±0.97	99.85±0.16	96.85±1.85	**100.00±0.00**	**100.00±0.00**
15	95.29±1.67	98.03±1.07	96.27±1.72	98.60±0.34	92.27±3.32	**99.86±0.36**	98.02±1.42
OA	80.35±0.61	89.31±1.00	91.40±0.92	87.54±1.03	87.07±1.12	90.51±0.95	**92.51±0.73**
AA	80.02±0.46	90.33±1.06	92.37±0.89	88.57±0.77	88.32±1.08	92.46±0.75	**93.57±0.60**
Kappa	78.72±0.66	88.44±1.08	90.70±1.00	86.53±1.12	86.01±1.21	89.74±1.03	**91.89±0.78**

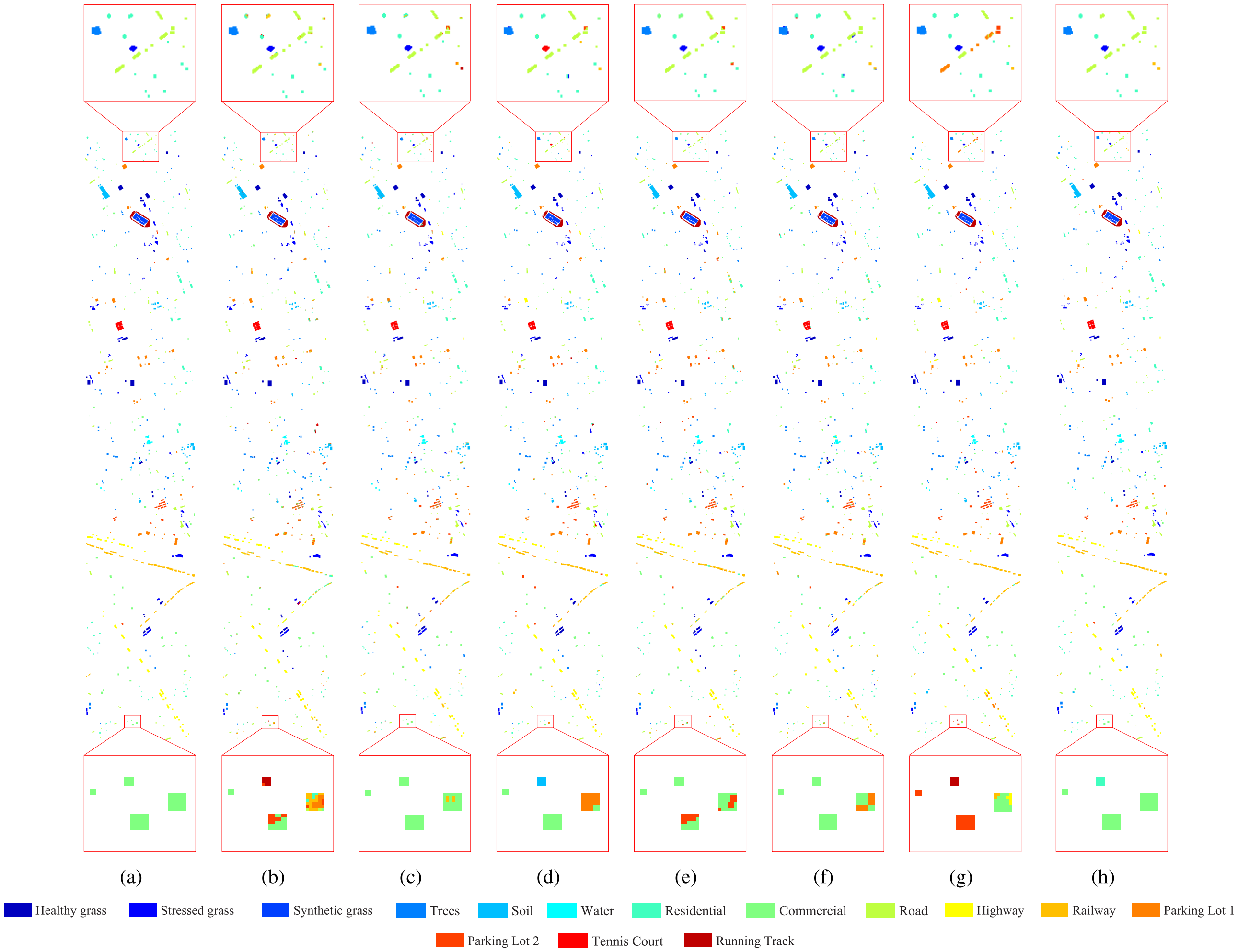


Fig. 10. Classification maps obtained by different methods on Houston University data set. (a) Ground truth map. (b) GCN. (c) S^2GCN. (d) MDGCN. (e) CNN-PPF. (f) MBCTU. (g) JSDF. (h) CAD-GCN. In (a)–(h), zoomed-in views of the regions are denoted by red boxes.

E. Impact of Hyperparameters

There are several important hyperparameters that should be manually tuned in the designed CAD-GCN architecture. Herein, we will evaluate in detail the sensitivity of the classification performance to different hyperparameter settings of the proposed CAD-GCN. Since GCN-based methods usually do not require deep structure to achieve excellent performance [26], [30], we empirically employ two convolutional

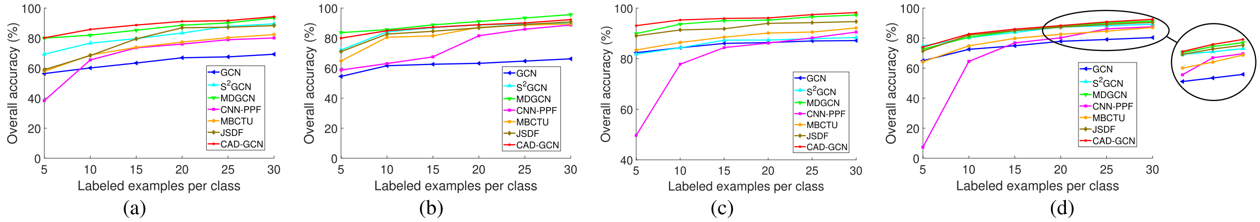


Fig. 11. Overall accuracies of various methods under different numbers of labeled examples per class. (a) Indian Pines data set. (b) University of Pavia data set. (c) Salinas data set. (d) Houston University data set.

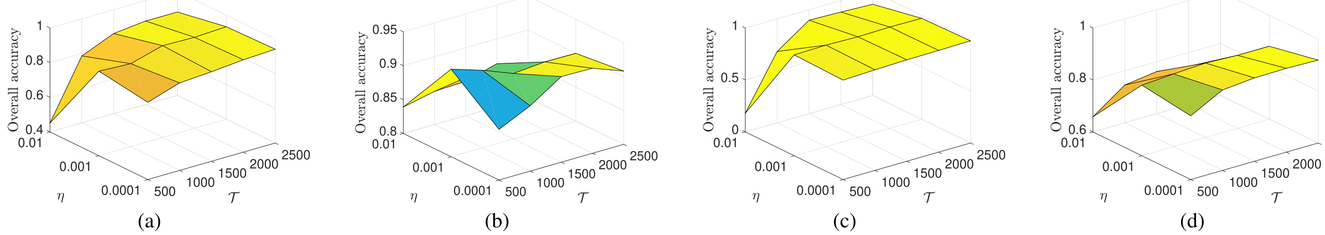


Fig. 12. Parametric sensitivity of η and $\mathcal{T}$. (a) Indian Pines data set. (b) University of Pavia data set. (c) Salinas data set. (d) Houston University data set.

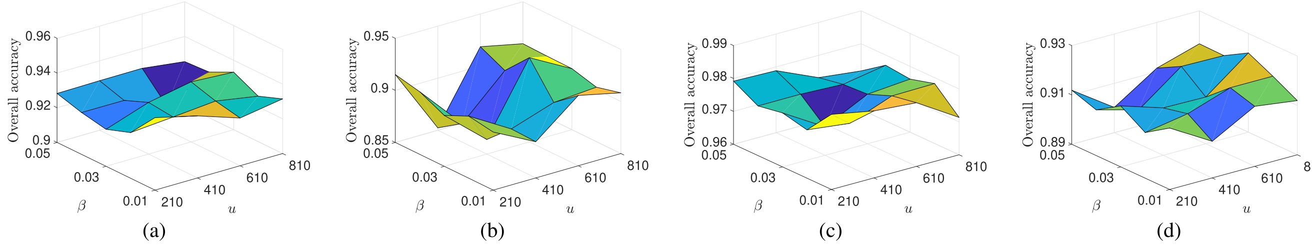


Fig. 13. Parametric sensitivity of β and u. (a) Indian Pines data set. (b) University of Pavia data set. (c) Salinas data set. (d) Houston University data set.

layers for all the four data sets. The hyperparameters to be pretuned manually mainly include the number of iterations $\mathcal{T}$, the learning rate η, the number of hidden units u, and the threshold β used in dynamic graph refinement. Herein, we adopt the grid search strategy to find the optimal setting. To facilitate the evaluation of the four hyperparameters, we divide them into two groups and report the OA with respect to the change of each group of hyperparameters, respectively. The parametric sensitivity of η and $\mathcal{T}$ is exhibited in Fig. 12, and Fig. 13 presents the parametric sensitivity of β and u.

In Fig. 12, we find that using a large learning rate η usually leads to unstable performances, while promising results can be obtained with a small η on all of the four data sets. Hence, it is reasonable to select a relatively small value for η on the four data sets. In addition, an appropriate number of iterations $\mathcal{T}$ is critical for achieving satisfactory performance. For the Indian Pines, the Salinas, and the Houston University data sets, the OA generally improves with an increase of $\mathcal{T}$. However, it is not the case on the University of Pavia data set, where the best result is reached when $\mathcal{T} = 500$.

The impact of β and u on the four data sets is revealed in Fig. 13. We observe that both β and u have an evident impact on the classification accuracies. Meanwhile, the best result is usually reached with a relatively small β on each data set, since useful information may be removed when adopting a large threshold value for edge filtering. For the number of hidden units u, although increasing the value of u will enhance the representation ability of the network, the risk of overfitting also gets higher. Therefore, we should carefully select a reasonable u for each data set, respectively.

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

ID	CAD-GCN-v1	CAD-GCN-v2	CAD-GCN-v3	CAD-GCN
1	99.04±1.91	98.88±2.13	**100.00±0.00**	99.46±1.54
2	86.02±5.20	86.17±5.10	77.22±1.38	**88.08±4.57**
3	93.01±4.16	93.14±3.88	92.43±0.61	**95.62±1.89**
4	98.08±2.11	98.47±1.18	**99.48±1.81**	97.85±2.75
5	93.73±3.35	**93.98±3.42**	93.60±1.53	93.79±3.71
6	96.86±2.08	**97.54±2.31**	96.05±0.82	96.41±1.78
7	96.50±4.49	97.03±4.86	**100.00±0.00**	97.95±2.84
8	99.36±1.45	98.55±1.90	95.87±1.16	**99.81±0.25**
9	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
10	90.43±1.82	89.65±4.01	**92.29±0.61**	89.34±4.03
11	90.24±2.62	89.21±2.88	88.20±2.15	**93.09±3.14**
12	92.33±3.29	**93.67±3.13**	88.19±0.92	93.41±2.54
13	99.76±0.36	99.75±0.36	99.76±0.82	**99.79±0.29**
14	97.72±2.28	98.71±2.96	96.13±0.33	**99.36±0.54**
15	98.79±1.01	99.04±1.15	95.27±0.16	**99.20±0.87**
16	98.86±1.30	98.86±1.38	93.92±0.92	**98.91±1.01**
OA	92.75±1.02	92.70±1.20	90.31±0.47	**94.13±0.78**
AA	95.67±0.57	95.79±0.53	94.27±0.29	**96.38±0.35**
Kappa	91.74±1.16	91.68±1.34	88.95±0.52	**93.29±0.88**

F. Ablation Study

As is mentioned in Section I, the proposed CAD-GCN contains three critical parts for improving the contextual relations that is the graph projection framework, the dynamic refinement

TABLE X

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

ID	CAD-GCN-v1	CAD-GCN-v2	CAD-GCN-v3	CAD-GCN
1	77.74±8.06	83.10±6.59	82.55±3.34	**83.85±4.80**
2	92.72±2.90	89.83±3.73	94.09±3.50	**95.56±1.97**
3	94.97±2.02	**95.69±3.62**	87.99±7.38	95.50±2.41
4	78.30±5.64	74.29±5.67	**89.91±6.60**	81.90±3.51
5	98.78±0.78	**98.95±1.10**	98.15±1.12	98.91±1.06
6	96.86±4.33	**98.44±1.25**	97.32±1.99	97.81±2.89
7	**97.57±3.00**	97.36±3.74	95.79±2.58	96.79±2.42
8	92.45±3.08	94.43±3.03	85.35±4.75	**95.35±2.76**
9	81.96±4.81	82.71±4.12	**94.23±3.49**	84.80±5.33
OA	90.04±1.69	89.72±1.43	91.51±1.51	**92.91±1.01**
AA	90.15±1.86	90.53±1.02	91.71±1.10	**92.27±0.94**
Kappa	86.99±2.19	86.65±1.79	88.88±1.89	**90.69±1.29**

TABLE XI

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT MODEL SETTINGS ON SALINAS DATA SET

ID	CAD-GCN-v1	CAD-GCN-v2	CAD-GCN-v3	CAD-GCN
1	**100.00±0.00**	**100.00±0.00**	99.93±0.03	**100.00±0.00**
2	**100.00±0.00**	**100.00±0.00**	99.95±0.06	**100.00±0.00**
3	99.96±0.09	**100.00±0.00**	99.86±0.12	99.97±0.07
4	98.24±1.47	**99.37±0.62**	96.99±0.89	98.46±1.48
5	97.06±1.79	96.29±1.86	**98.19±0.30**	97.77±1.05
6	99.60±0.36	99.51±0.35	98.77±0.93	**99.61±0.33**
7	**99.89±0.28**	98.86±1.75	99.73±0.17	99.10±1.56
8	94.59±2.23	95.11±3.48	88.76±1.04	**95.80±3.17**
9	**100.00±0.00**	**100.00±0.00**	99.65±0.55	**100.00±0.00**
10	96.46±1.82	97.03±2.46	94.38±2.27	**98.01±1.40**
11	99.68±0.56	99.24±0.89	96.72±1.78	**99.82±0.27**
12	97.74±0.81	97.65±1.18	**99.31±0.00**	97.99±0.78
13	97.26±1.31	97.88±0.60	**98.38±1.34**	97.90±0.98
14	**99.43±0.59**	99.36±0.26	95.61±2.91	98.89±1.53
15	97.44±1.42	97.39±1.82	90.52±0.86	**97.47±1.65**
16	99.63±1.06	99.74±0.85	98.39±1.98	**99.76±0.57**
OA	97.93±0.42	97.98±0.68	95.45±0.45	**98.28±0.54**
AA	98.56±0.19	98.59±0.31	97.20±0.30	**98.78±0.22**
Kappa	97.69±0.46	97.75±0.76	94.93±0.49	**98.09±0.60**

TABLE XII

PER-CLASS ACCURACY, OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT MODEL SETTINGS ON HOUSTON UNIVERSITY DATA SET

ID	CAD-GCN-v1	CAD-GCN-v2	CAD-GCN-v3	CAD-GCN
1	**95.04±2.02**	94.78±3.92	94.22±4.09	94.45±3.49
2	94.95±2.98	95.05±3.77	93.70±4.06	**96.43±2.83**
3	95.27±2.55	96.63±2.89	**97.24±1.59**	95.17±4.11
4	**96.47±2.54**	96.35±2.70	94.12±3.02	94.82±2.38
5	**99.29±1.08**	**99.29±1.11**	98.16±0.82	98.91±1.51
6	98.13±2.93	**98.34±2.46**	95.02±3.07	97.48±3.48
7	91.49±6.94	**91.90±3.96**	81.55±5.24	91.58±3.16
8	71.98±6.22	73.04±4.83	**79.32±6.18**	74.63±4.82
9	83.31±5.50	82.82±6.37	83.99±3.77	**86.75±3.58**
10	92.95±5.12	**94.89±4.23**	85.60±4.36	94.23±3.34
11	91.34±4.41	90.39±4.34	80.08±4.37	**94.65±2.73**
12	85.32±4.99	84.10±7.23	87.44±5.05	**89.55±1.93**
13	95.52±4.00	96.25±3.81	90.48±2.54	**96.80±3.68**
14	**100.00±0.00**	**100.00±0.00**	96.95±1.97	**100.00±0.00**
15	97.13±3.16	96.09±2.72	94.01±2.73	**98.02±1.42**
OA	91.22±1.05	91.29±0.75	88.77±0.78	**92.51±0.73**
AA	92.55±0.92	92.66±0.61	90.13±0.54	**93.57±0.60**
Kappa	90.50±1.14	90.58±0.81	87.85±0.85	**91.89±0.78**

TABLE XIII

RUNNING TIME COMPARISON (IN SECONDS) OF DIFFERENT METHODS. “IP" DENOTES INDIAN PINES DATA SET, “UH" DENOTES HOUSTON UNIVERSITY DATA SET, AND “PAVIAU" DENOTES UNIVERSITY OF PAVIA DATA SET

Dataset	GCN [18]	S^2GCN [26]	MDGCN [20]	CNN-PPF [27]	CAD-GCN
IP	**58**	71	95	1495	62
PaviaU	1783	1803	244	1545	**73**
Salinas	3497	3528	1108	1769	**826**
UH	901	971	360	1223	**159**

of node similarities, and the edge filter. To shed light on the contributions of these three components, every time we report the classification results of CAD-GCN without one of the three components on the four adopted data sets (namely the Indian Pines, the University of Pavia, the Salinas, and the Houston University). For simplicity, we adopt “CAD-GCN-v1," “CAD-GCN-v2," and “CAD-GCN-v3" to represent the reduced model by removing dynamic refinement of node similarities, the edge filter, and the graph projection framework, respectively. Tables IX–XII exhibit the comparative results on the aforementioned data sets. It can be obviously observed that lacking any one of the components will inevitably hurt the OA. Therefore, the graph projection framework, the dynamic refinement of node similarities, and the edge filter work collaboratively to render satisfactory classification performance.

G. Running Time

To reveal the advantage of our proposed CAD-GCN to the baselines in terms of efficiency, in Table XIII, we report the running time of different deep models, including GCN, S^2GCN, MDGCN, CNN-PPF, and the proposed CAD-GCN on four data sets (i.e., the Indian Pines, the University of Pavia, the Salinas, and the Houston University), where the number of labeled pixels per class is kept identical to the experiments presented in Section III-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. In Table XIII, we see that our proposed CAD-GCN shows the comparable efficiency to GCN on the Indian Pines data set and shows remarkably higher efficiency than other methods in large-scale data sets (i.e., the University of Pavia, the Salinas, and the Houston University data set), which is owing much to the employment of graph projection operation. Since the graph size can be significantly reduced by graph projection, our proposed CAD-GCN exhibits high efficiency on all the four data sets. The comparison results demonstrate that our proposed method is effective and efficient for HSI classification.

IV. CONCLUSION

In this article, we have developed a novel CAD-GCN for HSI classification. To capture long-range contextual relations, we move beyond regular image grids by learning the pixel-to-region assignment, and further encode the contextual relations among regions, so that the regions which are originally far away in the 2-D space can be connected by successive graph convolutions. Moreover, we enable the node similarities and connective relationships to be dynamically updated via learning the improved distance metric and the edge filter. Therefore, the contextual relations among pixels can be gradually refined along with graph convolution, which significantly improves the performance of CAD-GCN on representation and classification of HSI. The experimental results on four real-world HSI data sets indicate the effectiveness of the proposed CAD-GCN.

References

[1] J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.

[2] T. Matsuki, N. Yokoya, and A. Iwasaki, "Hyperspectral tree species classification of Japanese complex mixed forest with the aid of Lidar data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2177–2187, May 2015.

[3] 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.

[4] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[5] 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.

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

[7] 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.

[8] R. Kettig and D. Landgrebe, "Classification of multispectral image data by extraction and classification of homogeneous objects," *IEEE Trans. Geosci. Electron.*, vol. 14, no. 1, pp. 19–26, Jan. 1976.

[9] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image classification with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.

[10] B. Zhang, S. Li, X. Jia, L. Gao, and M. Peng, "Adaptive Markov random field approach for classification of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 973–977, Sep. 2011.

[11] G. Moser and S. B. Serpico, "Combining support vector machines and Markov random fields in an integrated framework for contextual image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2734–2752, May 2013.

[12] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "Classification of hyperspectral images by exploiting spectral–spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.

[13] Y. Xu, Z. Wu, and Z. Wei, "Spectral–spatial classification of hyperspectral image based on low-rank decomposition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2370–2380, Jun. 2015.

[14] A. Safia and D.-C. He, "Multiband compact texture unit descriptor for intra-band and inter-band texture analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 169–185, Jul. 2015.

[15] K. Djerriri, A. Safia, R. Adjoudj, and M. S. Karoui, "Improving hyperspectral image classification by combining spectral and multiband compact texture features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 465–468.

[16] C. Proctor, Y. He, and V. Robinson, "Texture augmented detection of macrophyte species using decision trees," *ISPRS J. Photogramm. Remote Sens.*, vol. 80, pp. 10–20, Jun. 2013.

[17] X. Xie and M. Mirmehdi, "TEXEMS: Texture exemplars for defect detection on random textured surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1454–1464, Aug. 2007.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

[19] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 3546–3553.

[20] 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.

[21] CR Formatted H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–4.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] J. Li *et al.*, "Multiple feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1592–1606, Mar. 2015.

[30] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, Jul. 2018, pp. 1416–1424.

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 Laboratory, 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 (Member, IEEE) received the B.E. degree from the East China University of Science and Technology (ECUST), Shanghai, China, in 2010, and the dual Ph.D. degree from Shanghai Jiao Tong University (SJTU), Shanghai, and the University of Technology Sydney (UTS), Ultimo, NSW, Australia, in 2016 and 2017, respectively.

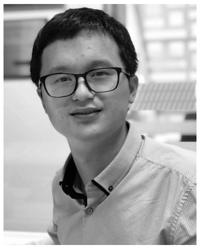
He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing. He has published more than 70 technical articles at prominent journals and conferences, such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Image Processing, the IEEE Transactions on Cybernetics, the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Multimedia, the IEEE Transactions on Intelligent Transportation Systems, the *ACM Transactions on Intelligent Systems and Technology*, the Conference and Workshop on Neural Information Processing Systems (NeurIPS), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the Association for the Advance of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), and International Conference on Data Mining (ICDM). His research interests mainly include machine learning, data mining, and learning-based vision problems.

Dr. Gong received the "Excellent Doctoral Dissertation" awarded by Shanghai Jiao Tong University (SJTU) and the Chinese Association for Artificial Intelligence (CAAI). He serves as a Reviewer for more than 20 international journals, such as Artificial Intelligence Journal, the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Image Processing, and also a SPC/PC Member for several top-tier conferences, such as the International Conference on Machine Learning (ICML), NeurIPS, AAAI, IJCAI, ICDM, and International Conference on Artificial Intelligence and Statistics (AISTATS). He was also enrolled by the "Young Elite Scientists Sponsorship Program" of Jiangsu Province and the China Association for Science and Technology.

Ping Zhong (Senior Member, IEEE) 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 articles in international journals, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. 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 the University of China in 2013. He is a Referee of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.



Shirui Pan received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia, in 2015.

He is currently a Lecturer with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia. Prior to this, he was a Lecturer with the School of Software, University of Technology Sydney. To date, he has published over 80 research articles in top-tier journals and conferences, including the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, International World Wide Web Conferences (WWW), the Association for the Advance of Artificial Intelligence (AAAI), and International Conference on Data Mining (ICDM). His research interests include data mining and machine learning.

Guangyu Li received the B.S. degree from the China University of Mining and Technology, Xuzhou, China, in 2008, the M.S. degree from Tongji University, Shanghai, China, in 2011, and the Ph.D. degree from the University of Paris-Sud, Paris, France, in 2015.

He is currently an Assistant Professor with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing, China. His research interests include information dissemination in vehicle networks, big data mining, electric vehicles charging/discharging scheduling strategy, and traffic control.

Jian Yang (Member, IEEE) 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 Centre, The 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, NJ, USA. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He has authored more than 200 scientific articles in pattern recognition and computer vision. His articles have been cited more than 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 the International Association for Pattern Recognition. He is/was currently an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*.