Graph Convolutional Networks for Hyperspectral Image Classification

Danfeng Hong, *Member, IEEE*, Lianru Gao, *Senior Member, IEEE*, Jing Yao, Bing Zhang, *Fellow, IEEE*, Antonio Plaza, *Fellow, IEEE*, and Jocelyn Chanussot, *Fellow, IEEE*

***Abstract*— Convolutional neural networks (CNNs) have been attracting increasing attention in hyperspectral (HS) image classification due to their ability to capture spatial–spectral feature representations. Nevertheless, their ability in modeling relations between the samples remains limited. Beyond the limitations of grid sampling, graph convolutional networks (GCNs) have been recently proposed and successfully applied in irregular (or nongrid) data representation and analysis. In this article, we thoroughly investigate CNNs and GCNs (qualitatively and quantitatively) in terms of HS image classification. Due to the construction of the adjacency matrix on all the data, traditional GCNs usually suffer from a huge computational cost, particularly in large-scale remote sensing (RS) problems. To this end, we develop a new minibatch GCN (called miniGCN hereinafter), which allows to train large-scale GCNs in a minibatch fashion. More significantly, our miniGCN is capable of inferring out-of-sample data without retraining networks and improving classification performance. Furthermore, as CNNs and GCNs can extract different types of HS features, an intuitive solution to break the performance bottleneck of a single model is to fuse them. Since miniGCNs can perform batchwise network training (enabling the combination of CNNs and GCNs), we explore three fusion strategies: additive fusion, elementwise multiplicative fusion, and concatenation fusion to measure the obtained performance gain. Extensive experiments, conducted on three HS data sets, demonstrate the advantages of miniGCNs over GCNs and the superiority of the tested fusion strategies with regard to the single CNN or GCN models. The codes of this work will be available at https://github.com/danfenghong/IEEE_TGRS_GCN for the sake of reproducibility.**

***Index Terms*— Hyperspectral (HS) classification, convolutional neural networks (CNNs), graph convolutional networks (GCNs), deep learning (DL), fusion.**

I. Introduction

LAND use and land cover (LULC) classification [1] using earth observation (EO) data, e.g., hyperspectral (HS) [2], synthetic aperture radar (SAR) [3], light detection and ranging (LiDAR) [4], and so on is a challenging topic in geoscience and remote sensing (RS). Characterized by their rich and detailed spectral information, HS images allow discriminating the objects of interest more effectively (particularly those in spectrally similar classes) by capturing more subtle discrepancies from the contiguous shape of the spectral signatures associated with their pixels. HS imagery enables the detection and recognition of the materials on the earth's surface at a more fine and accurate level compared with RGB and multispectral (MS) data. However, the high spectral mixing between materials [5] and spectral variability and complex noise effects [6] bring difficulties in extracting discriminative information from such data.

Over the past decades, a variety of handcrafted and learning-based feature extraction (FE) algorithms [7]–[15] (either unsupervised or supervised) have been successfully designed for HS image classification. Among them, morphological profiles (MPs) [16] are an effective tool that allows us to manually extract spatial–spectral features from HS images. Fauvel *et al.* [17] used MPs as input vectors for a support vector machine (SVM) classifier, achieving satisfactory classification results. Samat *et al.* [18] designed new maximally stable extremal region-guided MPs, yielding a high classification performance on MS images. Other works based on morphological operations have been developed to further enhance feature representations, including attribute profiles (APs) [19] and invariant APs [20], [21]. Another typical FE strategy is subspace-based learning, e.g., sparse representation [22], [23] and manifold learning [11], [24]. These methods learn transformations or projections by managing the high-dimensional original space using a new, latent, low-dimensional subspace representation. Although the aforementioned approaches have been proven to be effective in HS classification tasks, feature discrimination still remains limited due to the lack of powerful data fitting and representation ability.

Manuscript received June 19, 2020; revised July 19, 2020 and July 29, 2020; accepted August 5, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41722108 and Grant 91638201, and in part by the AXA Research Fund. *(Corresponding author: Lianru Gao.)*

Danfeng Hong is with the Universitė. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-laboratory, 38000 Grenoble, France (e-mail: hongdanfeng1989@gmail.com).

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

Jing Yao is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: jasonyao@stu.xjtu.edu.cn).

Bing Zhang is with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China, and also with the College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zb@radi.ac.cn).

Antonio Plaza is with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10003 Cáceres, Spain (e-mail: aplaza@unex.es).

Jocelyn Chanussot is with the Université Grenoble Alpes, INRIA, CNRS, Grenoble INP, LJK, F-38000 Grenoble, France, and also with Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: jocelyn@hi.is).

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.2020.3015157

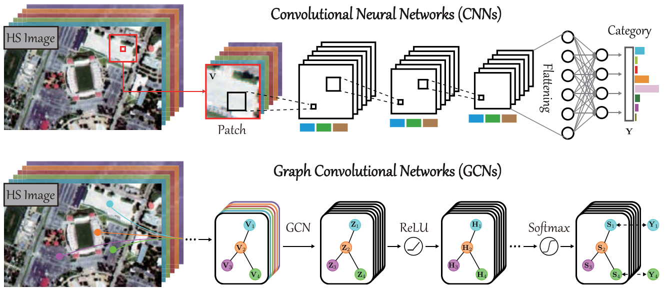

Convolutional Neural Networks (CNNs)
HS Image
Patch
Flattening
Category
Y
Graph Convolutional Networks (GCNs)
HS Image
GCN
ReLU
Softmax

Fig. 1. Comparison of CNN and GCN architectures in HS image classification tasks. The variables of **V**, **Z**, **H**, **S**, and **Y** in GCNs denote vertexes, hidden representations via GCN layer, hidden representations via ReLU layer, hidden representations via softmax layer, and labels, respectively.

Inspired by the success of deep learning (DL) techniques, significant progress has been made in the area of HS image classification by using various advanced deep networks [25]. Chen *et al.* [26] applied stacked autoencoder networks to dimensionally reduced HS images—obtained by principal component analysis (PCA)—for HS image classification. Furthermore, Chen *et al.* [27] adopted convolutional neural networks (CNNs) to extract spatial–spectral features more effectively from HS images, thereby yielding higher classification performance. Recurrent neural networks (RNNs) [28], [29] can process the spectral signatures as sequential data. In [30], a cascaded RNN was proposed to make full use of spectral information for high-accuracy HS image classification. Recently, Hang *et al.* [31] developed multitask generative adversarial networks and provided new insight into HS image classification, yielding state-of-the-art performance.

Comparatively, graph convolutional networks (GCNs) [32] are a hot topic and emerging network architecture, which is able to effectively handle graph structure data by modeling relations between samples (or vertexes). Therefore, GCNs can be naturally used to model long-range spatial relations in the HS image (see Fig. 1), which fails to be considered in CNNs. Currently, GCNs are less popular than CNNs in HS image classification. There are a few works related to the use of GCNs in HSI classification, though. Shahraki and Prasad [33] proposed to cascade 1-D CNNs and GCNs for HS image classification. Qin *et al.* [34] extended the original GCNs to a second-order version by simultaneously considering spatial and spectral neighborhoods. Wan *et al.* [35] performed superpixel segmentation on the HS image and fed it into GCN to reduce the computational cost and improve the classification accuracy. However, there are some potential limitations of GCNs regarding the following aspects.

1) The high computational cost (resulting from the construction of the adjacency matrix) is a significant bottleneck of GCNs in the HS image classification task, particularly when using large-scale HS image data.
2) GCNs only allow for full-batch network learning, that is, feeding all samples at once into the network. This might lead to large memory costs and slow gradient descent, as well as the negative effects of variable updating.
3) Last but not least, a trained GCN-based model fails to predict the new input samples (i.e., out of samples) without retraining the network, which has a major influence on the use of GCNs in practice.

To overcome these difficulties, in this work, we introduce a simple but effective minibatch GCN (called miniGCN). Similar to CNNs, miniGCNs can effectively train the network for classification on a downsampled graph (or topological structure) in minibatch fashion, and meanwhile, the learned model can be directly used for prediction purposes on new data. In addition, with our newly proposed miniGCNs, we aim to make a side-by-side comparison between CNNs and GCNs (both qualitatively and quantitatively) and raise an interesting question: which one between CNNs and GCNs can assist more in the HS image classification task? It is well known that CNNs and GCNs can extract and represent spectral information from HS images using different perspectives, i.e., spatial–spectral features of CNNs, graph (or relation) representations of GCNs, and so on. This naturally motivates us to jointly use them by investigating different fusion strategies, making them even more suitable for HS image classification. More specifically, the main contributions of this article are threefold.

1) We systematically analyze CNNs and GCNs with a focus on HS image classification. To the best of our knowledge, this is the first time that the potentials and drawbacks of GCNs (in comparison with CNNs) are investigated in the community.
2) We propose a novel supervised version of GCNs: miniGCNs, for short. As the name suggests, miniGCNs can be trained in minibatch fashion, trying to find a better and more robust local optimum. Unlike traditional GCNs, our miniGCNs are not only capable of training the networks using training set but also allow for a straightforward inference of large-scale, out-of-samples using the trained model.
3) We develop three fusion schemes, including additive fusion, elementwise multiplicative fusion, and concatenation fusion, to achieve better classification results in HS images by integrating features extracted from CNNs and our miniGCNs, in an end-to-end trainable network.

The remaining of this article is organized as follows. Section II deeply reviews GCN-related knowledge. Section III elaborates on the proposed miniGCNs and introduces three different fusion strategies in the context of a general end-to-end fusion network. Extensive experiments and analyses are given in Section IV. Section V concludes this article with some remarks and hints at plausible future research work.

II. Review of GCNs

In this section, we provide some preliminaries of GCNs by reviewing the basic definitions and notations, including graph construction and several important theorems and proofs for graph convolution in the spectral domain.

A. Definition of Graph

A graph is a complex nonlinear data structure, which is used to describe a one-to-many relationship in a non-Euclidean space. In our case, the relations between spectral signatures can be represented as an undirected graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ and $\mathcal{E}$ denote the vertex and edge sets, respectively. In our context, the vertex set consists of HS pixels, whereas the edge set is composed of the similarities between any two vertexes, i.e., $\mathcal{V}_i$ and $\mathcal{V}_j$.

B. Construction of the Adjacency Matrix

The adjacency matrix, denoted as $\mathbf{A}$, defines the relationships (or edges) between vertexes. Each element in $\mathbf{A}$ can be generally computed by using the following radial basis function (RBF):

$$\mathbf{A}_{i,j} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \tag{1}$$

where σ is a parameter to control the width of the RBF. The vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the spectral signatures associated with the vertexes v_i and v_j. Once $\mathbf{A}$ is given, we create the corresponding graph Laplacian matrix $\mathbf{L}$ as follows:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{2}$$

where $\mathbf{D}$ is a diagonal matrix representing the degrees of $\mathbf{A}$, i.e., $\mathbf{D}_{i,i} = \sum_j \mathbf{A}_{i,j}$ [36], [37]. To enhance the generalization ability of the graph [38], the symmetric normalized Laplacian matrix ($\mathbf{L}_{\text{sym}}$) can be used as follows:

$$\begin{aligned}\mathbf{L}_{\text{sym}} &= \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\end{aligned} \tag{3}$$

where $\mathbf{I}$ is the identity matrix.

C. Graph Convolutions in the Spectral Domain

Given two functions f and g, their convolution can be then written as

$$f(t) \star g(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{4}$$

where τ is the shifting distance and $\star$ denotes the convolution operator.

Theorem 1: The Fourier transform of the convolution of two functions f and g is the product of their corresponding Fourier transforms. This can be formulated as

$$\mathcal{F}[f(t) \star g(t)] = \mathcal{F}[f(t)] \cdot \mathcal{F}[g(t)] \tag{5}$$

where $\mathcal{F}$ and $\cdot$ denote the Fourier transform and pointwise multiplication, respectively.

Theorem 2: The inverse Fourier transform ($\mathcal{F}^{-1}$) of the convolution of two functions f and g is equal to 2π the product of their corresponding inverse Fourier transforms

$$\mathcal{F}^{-1}[f(t) \star g(t)] = 2\pi\mathcal{F}^{-1}[f(t)] \cdot \mathcal{F}^{-1}[g(t)]. \tag{6}$$

By means of the abovementioned two well-known theorems [39], i.e., (5) and (6), the convolution can be generalized to the graph signal as

$$f(t) \star g(t) = \mathcal{F}^{-1}\{\mathcal{F}[f(t)] \cdot \mathcal{F}[g(t)]\}]. \tag{7}$$

Hence, the convolution operation on a graph can be converted to define the Fourier transform ($\mathcal{F}$) or to find a set of basis functions.

Lemma 1: The basis functions of $\mathcal{F}$ can be equivalently represented by a set of eigenvectors of $\mathbf{L}$.

Proof: By referring to [39], we have the following proof. For many functions that do not converge in domain, e.g., $y(t) = t^2$, we can always find a real-valued exponential function $e^{-\sigma t}$ to make $y(t)e^{-\sigma t}$ converge, thereby satisfying the Dirichlet condition of $\mathcal{F}$, that is

$$\int_{-\infty}^{\infty} |y(t)e^{-\sigma t}|dt < \infty. \tag{8}$$

Plugging $y(t)e^{-\sigma t}$ into $\mathcal{F}$, we have

$$\int_{-\infty}^{\infty} y(t)e^{-\sigma t}e^{-2\pi i x \xi}dt \tag{9}$$

and we can rewrite (9) as

$$\int_{-\infty}^{\infty} y(t)e^{-st}dt \tag{10}$$

where $s = \sigma + 2\pi i x \xi$. Note that (10) is the Laplace transform. In other words, the eigenvectors of $\mathbf{L}$ are identical to the basis functions of $\mathcal{F}$. □

Given Lemma 1, we can perform spectral decomposition on $\mathbf{L}$. We then have

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \tag{11}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)$ is the set of eigenvectors of $\mathbf{L}$, that is, the basis of $\mathcal{F}$. As $\mathbf{U}$ is the orthogonal matrix, i.e., $\mathbf{U}\mathbf{U}^\top = \mathbf{E}$, (11) can also be written as

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top. \tag{12}$$

According to (12), $\mathcal{F}$ of f on a graph is $\mathcal{G}\mathcal{F}[f] = \mathbf{U}^\top f$, and the inverse transform becomes $f = \mathbf{U}\mathcal{G}\mathcal{F}[f]$. In analogy with (7), the convolution between f and g on a graph can be expressed as

$$\mathcal{G}[f \star g] = \mathbf{U}\{[\mathbf{U}^\top f] \cdot [\mathbf{U}^\top g]\}. \tag{13}$$

If we write $\mathbf{U}^\top g$ as g_θ, the convolution on a graph can be finally formulated as

$$\mathcal{G}[f \star g_\theta] = \mathbf{U}g_\theta\mathbf{U}^\top f \tag{14}$$

where g_θ can be regarded as the function of the eigenvalues ($\mathbf{\Lambda}$) of $\mathbf{L}$ with the respect to the variable θ, i.e., $g_\theta(\mathbf{\Lambda})$.

To reduce the computational complexity of (14), Hammond *et al.* [40] approximately fitted g_θ by applying the Kth order truncated expansion of Chebyshev polynomials. By doing so, (14) can be rewritten as

$$\mathcal{G}[f \star g_\theta] \approx \sum_{k=0}^{K} \theta'_k T_k(\widetilde{\mathbf{L}})f \tag{15}$$

where $T_k(\bullet)$ and θ'_k are the Chebyshev polynomials with respect to the variable $\bullet$ and the Chebyshev coefficients, respectively. $\widetilde{\mathbf{L}} = (2/\lambda_{\max})\mathbf{L}_{\text{sym}} - \mathbf{I}$ denotes the normalized $\mathbf{L}$.

By limiting $K = 1$ and assigning the largest eigenvalue $\lambda_{\max}$ of $\widetilde{\mathbf{L}}$ to 2 [32], (15) can be further simplified to

$$\mathcal{G}[f \star g_\theta] \approx \boldsymbol{\theta}\big(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\big)f. \tag{16}$$

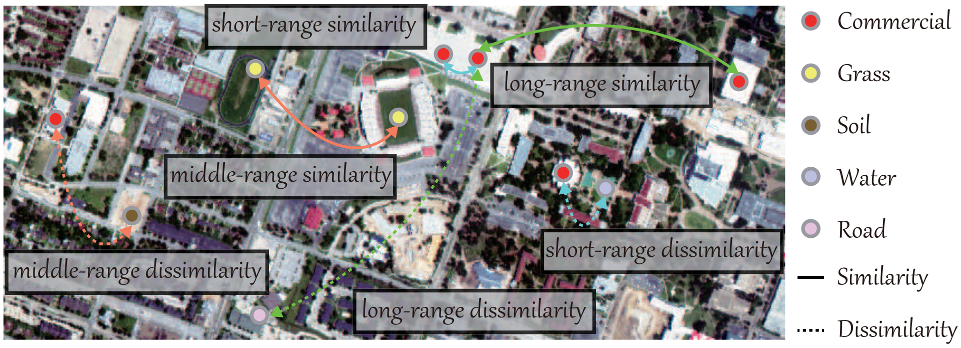


Fig. 2. Illustration of short-, middle-, and long-range spatial relations in an HS image. CNNs tend to extract locally spatial information, whereas GCNs are capable of capturing middle- or long-range spatial relationships (either similarities or dissimilarities) between samples.

Using (16), we have the following propagation rule for GCNs:

$$\mathbf{H}^{(\ell+1)} = h\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{b}^{(\ell)}\right) \tag{17}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\widetilde{\mathbf{D}}_{i,i} = \sum_j \widetilde{\mathbf{A}}_{i,j}$ are defined as the renormalization terms of $\mathbf{A}$ and $\mathbf{D}$, respectively, to enhance stability in the process of network training. Moreover, $\mathbf{H}^{(\ell)}$ denotes the output in the ℓ^{th} layer and $h(\bullet)$ is the activation function (e.g., ReLU, used in our case) with respect to the weights to-be-learned $\{\mathbf{W}^{(\ell)}\}_{\ell=1}^{p}$ and the biases $\{\mathbf{b}^{(\ell)}\}_{\ell=1}^{p}$ of all layers ($\ell = 1, 2, \ldots, p$).

III. Methodology

In this section, we systematically analyze CNNs and GCNs from four different perspectives and further develop an improvement of existing GCNs called miniGCNs, making them better applicable to the HS image classification task. Finally, we introduce three different fusion strategies, yielding a general end-to-end fusion network.

A. CNNs Versus GCNs: Qualitative Comparison

1) Data Preparation: It is well known that the input of CNNs is patchwise in HS image classification, and the output is the set of one-shot encoded labels. Unlike CNNs, GCNs feed pixelwise samples into the network with an adjacency matrix that models the relations between samples and needs to be computed before the training process starts.

2) Feature Representation: CNNs can extract rich spatial and spectral information from HS images in a short-range region, whereas GCNs are capable of modeling middle- and long-range spatial relations between samples by means of their graph structure. Fig. 2 shows such short-, middle-, and long-range relations in an HS scene.

3) Network Training: CNNs, as the main member of the DL family, are normally trained through the use of minibatch strategies. Conversely, GCNs only allow for full-batch network training since all samples need to be simultaneously fed into the network.

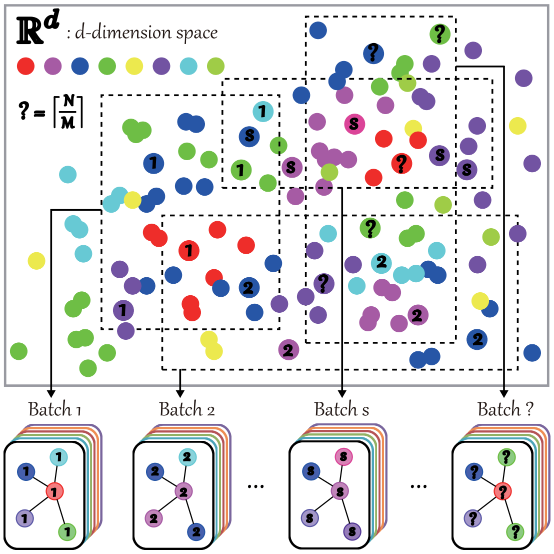


Fig. 3. Example illustrating how miniGCNs sample the subgraphs (or batches) from a full graph $\mathcal{G}$, aiming at training networks in a minibatch fashion. Solid circles: different colors denote spectral signatures of different classes in high-dimensional feature space. Dashed boxes: random sampling regions for each batch.

4) Computational Cost: The computational cost of CNNs and GCNs in one layer is mainly dominated by matrix products, yielding an overall $\mathcal{O}(NDP)$ and $\mathcal{O}(NDP + N^2D)$, respectively. N, D, and P denote the sample number, and the dimensions of the input and output features, respectively. Evidently, GCNs are computationally complex for large graphs compared with CNNs due to the large-sized matrix multiplication. To this end, a feasible solution might be the minibatch strategy performed in GCNs. If possible, the complexity of GCNs can be greatly reduced to $\mathcal{O}(NDP + NMD)$, where $M \ll N$ denotes the size of minibatches, thus having approximately the same order as CNNs with respect to N.

B. Proposed MiniGCNs

According to Sections III-A3 and III-A4, the computational cost of GCNs becomes high with an increase in the size of the graphs. To circumvent the computational burden on large graphs, a feasible solution (in analogy to CNNs) is to

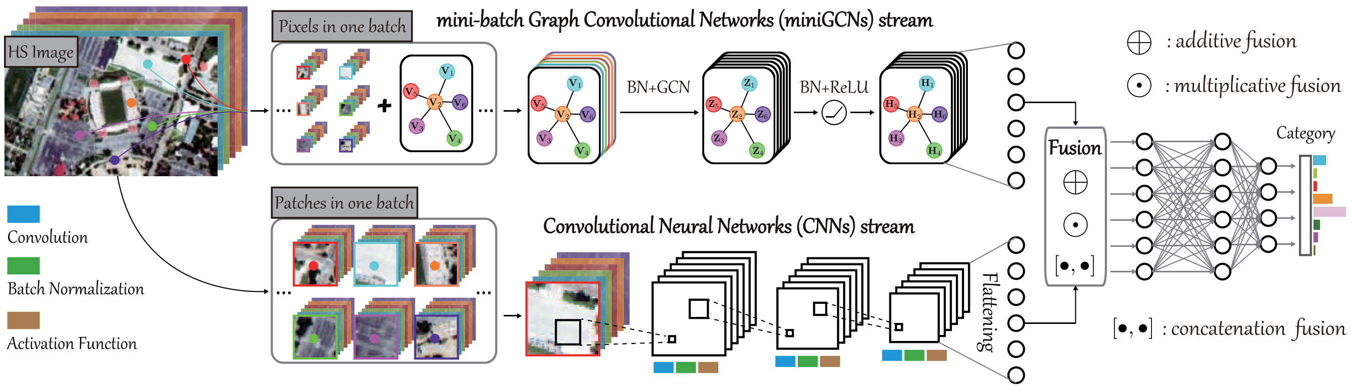

HS Image
Pixels in one batch
mini-batch Graph Convolutional Networks (miniGCNs) stream
BN+GCN
BN+ReLU
Patches in one batch
Convolutional Neural Networks (CNNs) stream
Flattening
Convolution
Batch Normalization
Activation Function
Fusion
⊕ : additive fusion
⊙ : multiplicative fusion
[•, •] : concatenation fusion
Category

Fig. 4. Overview of our end-to-end fusion network (FuNet), illustrating one batch training iteration. It comprises FE and fusion modules, where the former can extract different kinds of features (using both CNNs and miniGCNs) and the latter combines the resulting features using different fusion strategies before the final classification.

use minibatch processing. Inspired by inductive learning [41], we propose miniGCNs, making GCNs trainable in a minibatch fashion. Note that our inductive setting neither exploits features nor graph information of testing nodes in the training process.

Before presenting the new update rule of graph convolution in the proposed miniGCNs, we first cast a proposition—proved in [42]—to theoretically guarantee the applicability of the minibatch training strategy used in our miniGCNs. Given a full graph $\mathcal{G}$ with $|\mathcal{V}| = N$ on the labeled set, we construct a random node sampler with a budget M ($M \ll N$). Before training each epoch, we repeatedly apply the sampler to $\mathcal{G}$ until each node is sampled, yielding a set of subgraphs $\mathbb{G} = \{\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s) | s = 1, \ldots, \lceil (N/M) \rceil\}$, where $\lceil \bullet \rceil$ denotes the ceiling operation.

Proposition 1: Given a node v sampled from a certain subgraph $\mathcal{V}_s$, i.e., $v \in \mathcal{V}_s$, an unbiased estimator of the node v in the full-batch $(\ell + 1)$th GCN layer, denoted as $\mathbf{z}_v^{(\ell+1)}$, can be computed by aggregating features between v and all nodes $u \in \mathcal{V}_s$ in the ℓth layer

$$\mathbf{z}_v^{(\ell+1)} = \sum_{u\in\mathcal{V}_s} \frac{\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\right)_{uv}}{e_{uv}} \mathbf{z}_u^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{b}_u^{(\ell)} \tag{18}$$

i.e., $\mathbb{E}(\mathbf{z}_v^{(\ell+1)}) = \sum_{u\in\mathcal{V}}(\widetilde{\mathbf{D}}^{-(1/2)}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-(1/2)})_{uv}\mathbf{z}_u^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{b}^{(\ell)}$, if the constant of normalization e_{uv} is set to C_{uv}/C_v, where C_{uv} and C_v are defined as the number of times that node or edge occurs in all sampled subgraphs.

With Proposition 1 in mind, our miniGCNs can perform graph convolution in batches, just like CNNs. Using (17), the update rule in one batch can be directly given by

$$\widetilde{\mathbf{H}}_s^{(\ell+1)} = h\left(\widetilde{\mathbf{D}}_s^{-\frac{1}{2}}\widetilde{\mathbf{A}}_s\widetilde{\mathbf{D}}_s^{-\frac{1}{2}}\widetilde{\mathbf{H}}_s^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{b}_s^{(\ell)}\right) \tag{19}$$

where s is not only the sth subgraph but also the sth batch in the network training. Note that we consider a special case of Proposition 1: random node sampling without replacement, by simply setting $C_{uv} = C_v = 1$, i.e., $e_{uv} = 1$.

By collecting the outputs of all batches, the final output in the $(\ell + 1)$th layer can be reformulated as

$$\mathbf{H}^{(\ell+1)} = \left[\widetilde{\mathbf{H}}_1^{(\ell+1)}, \ldots, \widetilde{\mathbf{H}}_s^{(\ell+1)}, \ldots, \widetilde{\mathbf{H}}_{\lceil\frac{N}{M}\rceil}^{(\ell+1)}\right]. \tag{20}$$

Fig. 3 shows the process of batch generation in the proposed miniGCNs. This batch process is similar to the one adopted in CNNs, and the main difference lies in the fact that the graph or adjacency matrix in the obtained batch needs to be reassembled according to the connectivity of $\mathcal{G}$ after each sampling.

C. MiniGCNs Meet CNNs: End-to-End Fusion Networks

Different network architectures are capable of extracting distinctive representations of HS images, e.g., spatial–spectral features in CNNs or topological relations between samples in GCNs. Generally speaking, a single model may not provide optimal results in terms of performance due to the lack of feature diversity.

In this section, we naturally propose to fuse different models or features to enhance feature discrimination ability by jointly training CNNs and GCNs. Unlike traditional GCNs, the proposed miniGCNs can perform minibatch learning and can be combined with standard CNN models. This yields an end-to-end fusion network, called FuNet hereinafter. Three fusion strategies, additive (A), elementwise multiplicative (M), and concatenation (C), are considered. The three fusion models (A, M, and C) can be, respectively, formulated as follows:

$$\mathbf{H}_{\mathrm{FuNet-A}}^{(\ell+1)} = \mathbf{H}_{\mathrm{CNNs}}^{(\ell)} \oplus \mathbf{H}_{\mathrm{miniGCNs}}^{(\ell)} \tag{21}$$

$$\mathbf{H}_{\mathrm{FuNet-M}}^{(\ell+1)} = \mathbf{H}_{\mathrm{CNNs}}^{(\ell)} \odot \mathbf{H}_{\mathrm{miniGCNs}}^{(\ell)} \tag{22}$$

$$\mathbf{H}_{\mathrm{FuNet-C}}^{(\ell+1)} = \left[\mathbf{H}_{\mathrm{CNNs}}^{(\ell)}, \mathbf{H}_{\mathrm{miniGCNs}}^{(\ell)}\right] \tag{23}$$

where the operators $\oplus$, $\odot$, and $[\cdot, \cdot]$, respectively, denote the elementwise addition, elementwise multiplication, and concatenation. Accordingly, $\mathbf{H}_{\mathrm{CNNs}}^{(\ell)}$ and $\mathbf{H}_{\mathrm{miniGCNs}}^{(\ell)}$ are represented as the ℓth layer features extracted from CNNs and miniGCNs, respectively.

Fig. 4 shows one batch training iteration of CNNs and miniGCNs in our newly proposed end-to-end fusion networks. As it can be seen, it comprises FE and fusion modules, where the former can extract different kinds of features (using both CNNs and miniGCNs) and the latter combines the resulting features using different fusion strategies before the final classification.

TABLE I

LAND-COVER CLASSES OF THE INDIAN PINES DATA SET, WITH THE NUMBER OF TRAINING AND TEST SAMPLES SHOWN FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Corn Notill	50	1384
2	Corn Mintill	50	784
3	Corn	50	184
4	Grass Pasture	50	447
5	Grass Trees	50	697
6	Hay Windrowed	50	439
7	Soybean Notill	50	918
8	Soybean Mintill	50	2418
9	Soybean Clean	50	564
10	Wheat	50	162
11	Woods	50	1244
12	Buildings Grass Trees Drives	50	330
13	Stone Steel Towers	50	45
14	Alfalfa	15	39
15	Grass Pasture Mowed	15	11
16	Oats	15	5
	Total	695	9671

TABLE II

LAND-COVER CLASSES OF THE PAVIA UNIVERSITY DATA SET, WITH THE NUMBER OF TRAINING AND TEST SAMPLES SHOWN FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Asphalt	548	6304
2	Meadows	540	18146
3	Gravel	392	1815
4	Trees	524	2912
5	Metal Sheets	265	1113
6	Bare Soil	532	4572
7	Bitumen	375	981
8	Bricks	514	3364
9	Shadows	231	795
	Total	3921	40002

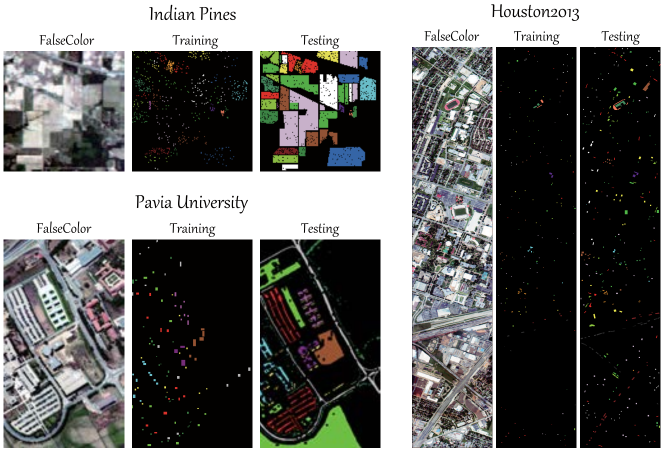


Fig. 5. False-color images and the distribution of training and test sets on the three considered data sets, i.e., Indian Pines, Pavia University, and Houston2013.

IV. EXPERIMENTS

A. Data Description

Three widely used HS data sets are adopted to assess the classification performance of our proposed algorithms, both quantitatively and qualitatively.

1) Indian Pines Data Set: The first HS data set was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over northwestern Indiana, USA. The scene comprises of 145 × 145 pixels with a ground sampling distance (GSD) of 20 m and 220 spectral bands in the wavelength range from 400 to 2500 nm, at 10-nm spectral resolution. We retain 200 channels by removing 20 noisy and water absorption bands, i.e., 104–108, 150–163, and 220. Table I lists 16 main land-cover categories involved in this studied scene, as well as the number of training and testing samples used for the classification task. Correspondingly, Fig. 5 shows a false-color image of this scene and the spatial distribution of training and test samples.

2) Pavia University Data Set: The second HS scene is the well-known Pavia University, which was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor. The ROSIS sensor acquired 103 bands covering the spectral range from 430 to 860 nm, and the scene consists of 610 × 340 pixels at GSD of 1.3 m. Moreover, there are nine land cover classes in the scene. The class name and the number of training and test sets are detailed in Table II, while the distribution of these samples is shown in Fig. 5.

3) Houston2013 Data Set: This data set was used for the 2013 IEEE GRSS data fusion contest,[1] and was collected using the ITRES CASI-1500 sensor over the campus of University of Houston and its surrounding rural areas in TX, USA. The image size is 349×1905 pixels with 144 spectral bands ranging from 364 to 1046 nm, at 10-nm spectral resolution. It should be noted that the used data set is a cloud-free HS product, processed by removing some small structures according to the illumination-related threshold maps computed based on the spectral signatures.[2] Table III lists 15 challenging land-cover categories and the training and test sets. In Fig. 5, we show a false-color image of the HS scene and the corresponding distribution of the training and test samples.

B. Experimental Setup

1) Implementation Details: All networks considered in this article are implemented using the Tensorflow platform, and Adam [43] is used to optimize the networks. By following the "exponential" learning rate policy, the current learning rate can be dynamically updated by multiplying a base learning rate (e.g., 0.001) by $(1 - (\text{iter}/\text{maxIter}))^{0.5}$ at intervals of 50 epochs. In the process of network training, the maximum number of epochs is set to 200. Batch normalization (BN) [44] is adopted with the 0.9 momentum, and the batch size in the training phase is set to 32. Moreover, the ℓ_2-norm regularization, set to 0.001, is employed on weights to stabilize the network training and reduce overfitting.

Note that the size for each layer and the hyperparameters in networks, such as learning rate and regularization, can be determined by tenfold cross validation,

[1] http://www.grss-ieee.org/community/technical-committees/data-fusion/2013-ieee-grss-data-fusion-contest/

[2] The data were provided by Prof. N. Yokoya from The University of Tokyo.

TABLE III

LAND-COVER CLASSES OF THE HOUSTON2013 DATA SET, WITH THE NUMBER OF TRAINING AND TEST SAMPLES SHOWN FOR EACH CLASS

Class No.	Class Name	Training	Testing
1	Healthy Grass	198	1053
2	Stressed Grass	190	1064
3	Synthetic Grass	192	505
4	Tree	188	1056
5	Soil	186	1056
6	Water	182	143
7	Residential	196	1072
8	Commercial	191	1053
9	Road	193	1059
10	Highway	191	1036
11	Railway	181	1054
12	Parking Lot1	192	1041
13	Parking Lot2	184	285
14	Tennis Court	181	247
15	Running Track	187	473
	Total	2832	12197

TABLE IV

GENERAL NETWORK CONFIGURATION IN EACH LAYER OF OUR FUNET. FC, CONV, AND MAXPOOL STAND FOR FC, CONVOLUTION, AND MAX POOLING, RESPECTIVELY, WHEREAS D AND P DENOTE THE INPUT AND OUTPUT DIMENSION IN THE NETWORKS, RESPECTIVELY. FURTHERMORE, THE LAST COMPONENT IN EACH BLOCK REPRESENTS THE OUTPUT SIZE

End-to-end Fusion Networks (FuNet)		CNNs	miniGCNs
Input Dimension		$7 \times 7 \times D$	D
Feature Extraction	Block1	3×3 Conv BN 2×2 MaxPool ReLU $4 \times 4 \times 32$	BN Graph Conv BN ReLU 128
	Block2	3×3 Conv BN 2×2 MaxPool ReLU $2 \times 2 \times 64$	– – – – –
	Block3	1×1 Conv BN 2×2 MaxPool ReLU $1 \times 1 \times 128$	– – – – –
Feature Fusion	Block4	FC Encoder BN ReLU 128	
	Block5	FC Encoder Softmax P	
Ouput Dimension		P	

e.g., using a grid search on the validation set. Ten replications are performed to randomly separate the original training set into the new training set and validation set, with a percentage of 80%–20%. More specifically, we perform cross validation to select the size of each layer and hyperparameters in the range of $\{16, 32, 64, 128, 256\}$ and $\{0.0001, 0.001, 0.01, 0.1, 1\}$, respectively. More details regarding the parameter settings can refer to our toolbox (or codes) that will be released after publication.

Furthermore, three commonly used indices, i.e., overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ), are used to evaluate the classification performance quantitatively.

2) Comparison With State-of-the-Art Baseline Methods: Several state-of-the-art baseline methods have been selected for comparison, including K-nearest neighbor (KNN) classifier, random forest (RF), 1-D CNN, 2-D CNN, GCN, and our proposed miniGCN, as well as three different fusion networks with different strategies: FuNet-A, FuNet-M, and FuNet-C. The parameter settings are described in the following.

1) For the KNN, we set the number of nearest neighbors (K) to 10, to be consistent with that of K in GCN-related methods, e.g., GCN, miniGCN, and FuNet.
2) For the RF, 200 decision trees are used in the classifier.
3) For the SVM, the well-known libsvm toolbox[3] is used for implementation in our case. The considered SVM uses the RBF kernel, whose two optimal hyperparameters σ and λ (the regularization parameter to balance the training and testing errors) can be determined by fivefold cross validation in the range $\sigma = [2^{-3}, 2^{-2}, \ldots, 2^{4}]$ and $\lambda = [10^{-2}, 10^{-1}, \ldots, 10^{4}]$.
4) For the 1-D CNN, we use one convolutional block, including a 1-D convolutional layer with a filter size of 128, a BN layer, a ReLU activation layer, and a softmax layer with the size of P, where P denotes the dimension of network output.
5) For the 2-D CNN (similar to 1-D CNN), the architecture is composed of three 2-D convolutional blocks and a softmax layer. Each convolutional block involves a 2-D conventional layer, a BN layer, a max-pooling layer, and a ReLU activation layer. Moreover, the receptive fields along the spatial and spectral domains for each convolutional layer are $3 \times 3 \times 32$, $3 \times 3 \times 64$, and $1 \times 1 \times 128$, respectively.
6) For the 3-D CNN, we adopt the same network architecture as the one in [27]. The only difference lies in that we remove the dropout layer in each block to make a fair comparison with other networks, e.g., 2-D CNN.
7) For the GCN, similar to [32], a graph convolutional hidden layer with 128 units is implemented in the GCN before feeding the features into the softmax layer, where the adjacency matrix $\widetilde{\mathbf{A}}$ can be computed using KNN-based graph ($K = 10$ in our case). The graph convolution, GCN, and 1-D CNN share the same network configuration for a fair comparison.
8) Our miniGCN has the same architecture as the GCN. The main difference between GCN and miniGCN lies in the fact that miniGCN is capable of training the networks in batchwise fashion and tends to reach a better local optimum of networks.
9) To better exploit diverse information of HS images, e.g., features extracted from CNNs or GCNs, our FuNets with A, M, and C different fusion strategies are developed by additionally adding a fully connected (FC) fusion layer behind CNNs and miniGCNs. Table IV details the

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

TABLE V

QUANTITATIVE COMPARISON OF DIFFERENT ALGORITHMS IN TERMS OF OA, AA, AND κ ON THE INDIAN PINES DATA SET. THE BEST ONE IS SHOWN IN BOLD

Class No.	KNN	RF	SVM	1-D CNN	2-D CNN	3-D CNN	GCN	miniGCN	FuNet-A	FuNet-M	FuNet-C
1	45.45	57.80	67.34	47.83	65.90	66.26	65.97	**72.54**	68.64	69.51	68.50
2	46.94	56.51	67.86	42.35	76.66	71.94	72.70	55.99	80.99	**82.40**	79.59
3	77.72	80.98	93.48	60.87	92.39	97.28	87.50	92.93	95.11	94.57	**99.46**
4	84.56	85.68	94.63	89.49	93.96	95.08	93.74	92.62	**96.64**	96.42	95.08
5	80.06	79.34	88.52	92.40	87.23	88.09	91.39	94.98	95.41	**96.99**	95.70
6	97.49	95.44	94.76	97.04	97.27	98.18	97.49	98.63	99.32	**99.54**	**99.54**
7	64.81	**77.56**	73.86	59.69	77.23	75.38	75.38	64.71	72.98	76.80	75.93
8	48.68	58.85	52.07	65.38	57.03	56.29	51.70	68.78	**70.31**	58.97	68.90
9	44.33	62.23	72.70	**93.44**	72.87	78.01	62.77	69.33	74.82	74.82	71.63
10	96.30	95.06	98.77	99.38	**100.00**	**100.00**	96.91	98.77	99.38	99.38	99.38
11	74.28	88.75	86.17	84.00	**92.85**	90.59	86.25	87.78	85.93	79.50	89.55
12	15.45	54.24	71.82	86.06	88.18	90.30	66.97	50.00	**93.03**	91.21	91.52
13	91.11	97.78	95.56	91.11	**100.00**	**100.00**	95.56	**100.00**	**100.00**	**100.00**	**100.00**
14	33.33	56.41	82.05	84.62	84.62	74.36	71.79	48.72	79.49	82.05	**94.87**
15	81.82	81.82	90.91	**100.00**	**100.00**	**100.00**	81.82	72.73	**100.00**	**100.00**	**100.00**
16	40.00	**100.00**	**100.00**	80.00	**100.00**	**100.00**	**100.00**	80.00	**100.00**	**100.00**	**100.00**
OA (%)	59.17	69.80	72.36	70.43	75.89	75.48	71.97	75.11	79.76	76.76	**79.89**
AA (%)	63.90	76.78	83.16	79.60	86.64	86.36	81.12	78.03	88.25	87.64	**89.35**
κ	0.5395	0.6591	0.6888	0.6642	0.7281	0.7240	0.6852	0.7164	0.7698	0.7382	**0.7716**

TABLE VI

QUANTITATIVE PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS IN TERMS OF OA, AA, AND κ ON THE PAVIA UNIVERSITY DATA SET. THE BEST ONE IS SHOWN IN BOLD

Class No.	KNN	RF	SVM	1-D CNN	2-D CNN	3-D CNN	GCN	miniGCN	FuNet-A	FuNet-M	FuNet-C
1	73.86	79.81	74.22	88.90	80.98	80.69	76.49	96.35	**96.99**	96.47	96.67
2	64.31	54.90	52.79	58.81	81.70	89.12	70.15	89.43	**97.74**	97.36	97.60
3	55.10	46.34	65.45	73.11	67.99	65.90	62.70	**87.01**	83.98	83.44	84.49
4	94.95	**98.73**	97.42	82.07	97.36	98.45	98.35	94.26	96.45	84.40	89.95
5	99.19	99.01	99.46	99.46	99.64	99.19	99.37	99.82	99.55	**100.00**	99.64
6	65.16	75.94	93.48	**97.92**	97.59	92.37	83.22	43.12	71.33	85.30	90.56
7	84.30	78.70	87.87	88.07	82.47	76.04	88.38	**90.96**	66.67	63.80	78.27
8	84.10	90.22	89.39	88.14	**97.62**	95.81	92.33	77.42	69.61	71.53	71.73
9	98.36	97.99	**99.87**	**99.87**	95.60	95.72	95.72	87.27	99.86	99.22	98.04
OA (%)	70.53	69.67	70.82	75.50	86.05	88.44	77.99	79.79	89.00	90.34	**92.20**
AA (%)	79.68	80.18	84.44	86.26	88.99	88.14	85.19	85.07	86.91	86.84	**89.66**
κ	0.6268	0.6237	0.6423	0.6948	0.8187	0.8472	0.7196	0.7367	0.8540	0.8709	**0.8951**

configuration of our FuNet for the layerwise network architecture.

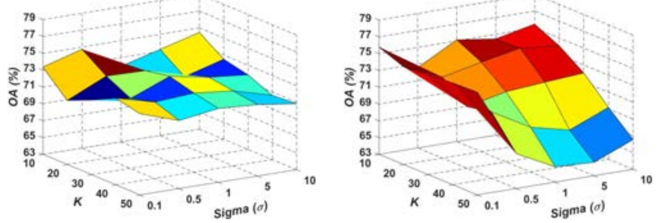
It should be noted, however, that the patch centered by a pixel is usually used as the input of CNNs in HS image classification. In this connection, the original HS image is extended by the "replicate" operation, i.e., copying the pixels within the image to that out of the original image boundary, to solve the problem of the boundaries in the CNN-related experiments.

C. Parameter Analysis on $\widetilde{\mathbf{A}}$ Generation

Since the performance of GCNs depends (to some extent) on the quality of adjacency matrix, i.e., $\widetilde{\mathbf{A}}$, it is important to investigate the performance gains that can be obtained by adjusting the two parameters: number of neighbors (K) and width of RBF function (σ) of $\widetilde{\mathbf{A}}$ [see (1)]. For this purpose, we show the changing trend (in terms of OA) for different combinations of the two parameters in the Indian Pines data. More specifically, GCNs and miniGCNs are selected to analyze the parameter sensitivity. As it can be seen from Fig. 6, the parameter K (to a large extent) dominates the performance gain. Nevertheless, the OAs of GCNs and miniGCNs remain stable with an increase of K value. On the other hand, varying the parameter σ only yields a slight performance fluctuation, indicating that this parameter might not be correctly fine-tuned. Most importantly, we observed that the performance gain or derogation in miniGCNs is relatively slow and gentle with the gradual change of the two parameters. In turn, with different parameter combinations, the GCNs lead to comparatively more perturbed results. Moreover, the whole classification performance of GCNs also seems to reach a bottleneck, because its full-batch training strategy usually fails to find a better local optimum. Comprehensively, the parameter combination of (K, σ) in our case is set to $(10, 1)$ since this parameter range is relatively stable and, hence, it is applied to the rest of the considered data sets for simplicity.

D. Quantitative Evaluation

Tables V–VII quantitatively report the classification scores obtained by different methods in terms of OA, AA, and κ,



(a) (b)

Fig. 6. Parameter sensitivity analysis (on the Indian Pines data) of the adjacency matrix $\widetilde{\mathbf{A}}$ [see (1)] in terms of K and σ for (a) GCNs and (b) miniGCNs.

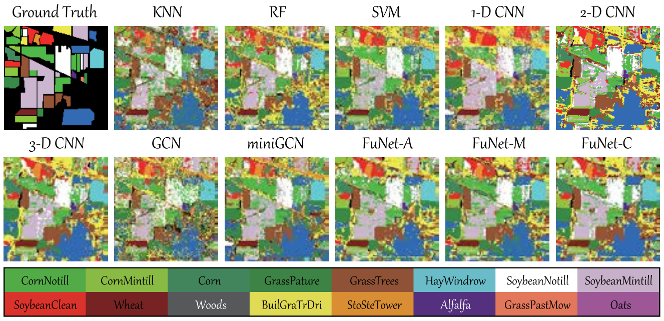


Fig. 7. Ground truth and classification maps obtained by different methods on the Indian Pines data set.

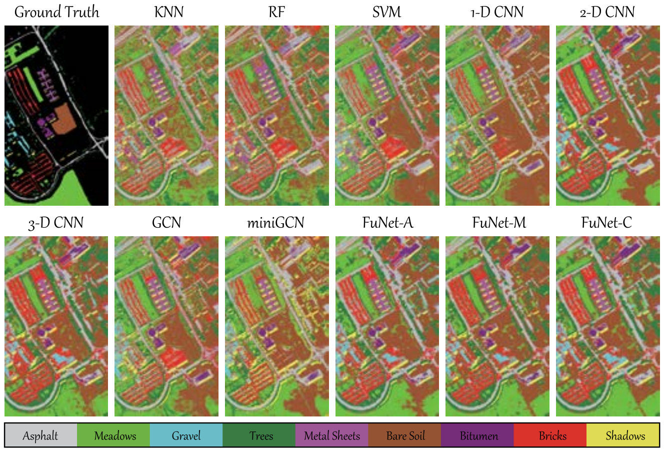


Fig. 8. Ground truth and classification maps obtained by different methods on the Pavia University data set.

as well as the individual class accuracies for the Indian Pines, Pavia University, and Houston2013 data sets, respectively.

Overall, KNN, RF, and SVM obtain similar classification results on the Pavia University and Houston2013 data sets, while the classification performance of the KNN classifier is inferior to that achieved using the RF on the Indian Pines data set. This might be explained by a few noisy training samples. Please note that there is a similar trend between RF and SVM in classification performance. By means of the powerful learning ability of DL techniques, 1-D CNN, 2-D CNN, 3-D CNN, and GCN perform better than traditional classifiers (KNN, RF, and SVM). Unlike 1-D CNN and GCN that only consider pixelwise network input, 2-D CNN and 3-D CNN can extract the spatial–spectral information from HS images more effectively, yielding higher classification accuracies. Not surprisingly, the performance of 3-D CNN is generally superior to that of 2-D CNN, due to the additional local convolution on the spectral domain. We have to point out, however, that the 3-D CNN requires additional network parameters to be estimated and tends to suffer from overfitting problems (particularly with limited training samples). The resulting accuracies on the Indian Pines data set demonstrate these potential problems. Moreover, GCN brings moderate increments of at least 1% OA, AA, and κ over the 1-D CNN since the spatial relation between samples can be well-modeled in the form of a graph structure by GCNs.

Remarkably, our miniGCN achieves stable performance improvements when compared to either GCN or 1-D CNN, even making it comparable to 2-D CNN to some extent, e.g., on the Indian Pines and Houston2013 data sets. As expected, the FuNet (that combines the benefits of CNNs and GCNs) outperforms those single models, demonstrating its ability to fuse different spectral representations. More specifically, a comparison between the three commonly used fusion strategies reveals that FuNet-C tends to obtain better classification performance compared with FuNet-A and FuNet-M, particularly on the Indian Pines and Pavia University, where there is a dramatic performance improvement (see Tables V and VI).

Furthermore, for those classes that have very few samples, e.g., *Alfalfa*, *Grass Pasture Mowed*, *Oats* on Indian Pines, or unbalanced samples, e.g., *Road*, *Parking Lot2* on Houston2013, the 2-D CNN and 3-D CNN can obtain higher classification accuracies by considering the contextual information in both the spatial and spectral domains. On the contrary, the GCN-based models fail to accurately model those classes. However, it is worth noting that the fused networks are capable of better identifying these challenging classes, due to the joint use of spatial–spectral (2-D CNN) and relation-augmented (miniGCN) features.

E. Visual Comparison

We also make a visual comparison between different classification methods in the form of classification maps, as shown in Figs. 7–9. In general, pixelwise classification models (e.g., KNN, RF, SVM, and 1-D CNN) result in salt and pepper noise in the classification maps. Although the GCN considers the spatial relation modeling between samples, the use of large graphs constructed based on all samples (and full-batch network training) limits its performance to a great extent, thereby yielding relatively poor classification maps. Our proposed miniGCN extracts the HS features by locally preserving the graph (or manifold) structure in one batch, leading to results that are comparable to those obtained by the 2-D CNN and 3-D CNN. This means that we can achieve relatively robust representations compared to full graph preservation since the batchwise strategy can eliminate some errors resulting from the manually computed adjacency matrix and further reduce the error accumulation and propagation between layers. As expected, the FuNet-based methods obtain smoother and more detailed maps in comparison with other competitors,

TABLE VII
QUANTITATIVE PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS IN TERMS OF OA, AA, AND κ ON THE HOUSTON2013 DATA SET. THE BEST ONE IS SHOWN IN BOLD

Class No.	KNN	RF	SVM	1-D CNN	2-D CNN	3-D CNN	GCN	miniGCN	FuNet-A	FuNet-M	FuNet-C
1	83.19	83.38	83.00	87.27	85.09	84.71	90.14	**98.39**	84.33	83.86	85.75
2	95.68	98.40	98.40	98.21	99.91	99.34	99.08	92.11	**100.00**	98.59	99.44
3	99.41	98.02	99.60	**100.00**	77.23	84.55	79.94	99.60	82.57	83.37	80.79
4	97.92	97.54	98.48	92.99	97.73	98.01	96.69	96.78	98.48	**98.96**	98.58
5	96.12	96.40	97.82	97.35	99.53	97.82	86.18	97.73	98.86	**99.72**	99.24
6	92.31	**97.20**	90.91	95.10	92.31	93.01	33.33	95.10	95.80	96.50	95.10
7	80.88	82.09	90.39	77.33	92.16	86.29	**97.09**	57.28	88.43	89.55	91.60
8	48.62	40.65	40.46	51.38	79.39	76.26	71.63	68.09	85.94	**89.36**	74.83
9	72.05	69.78	41.93	27.95	**86.31**	84.23	70.93	53.92	85.08	83.29	85.27
10	53.19	57.63	62.64	**90.83**	43.73	74.32	72.17	77.41	72.30	79.25	79.25
11	86.24	76.09	75.43	79.32	**87.00**	82.35	85.22	84.91	81.69	79.89	82.35
12	44.48	49.38	60.04	76.56	66.28	77.71	63.41	77.23	79.06	**79.15**	78.87
13	28.42	61.40	49.47	69.47	**90.18**	81.05	62.34	50.88	**90.18**	87.72	89.12
14	97.57	**99.60**	98.79	99.19	90.69	88.66	89.73	98.38	90.69	93.93	88.26
15	98.10	97.67	97.46	98.10	77.80	84.57	**99.36**	98.52	93.66	98.94	86.68
OA (%)	77.30	77.48	76.91	80.04	83.72	86.04	81.19	81.71	87.73	**88.62**	87.39
AA (%)	78.28	80.35	78.99	82.74	84.35	86.19	79.82	83.09	88.47	**89.47**	87.68
κ	0.7538	0.7564	74.94	0.7835	0.8231	0.8483	0.7962	0.8018	0.8668	**0.8764**	0.8631

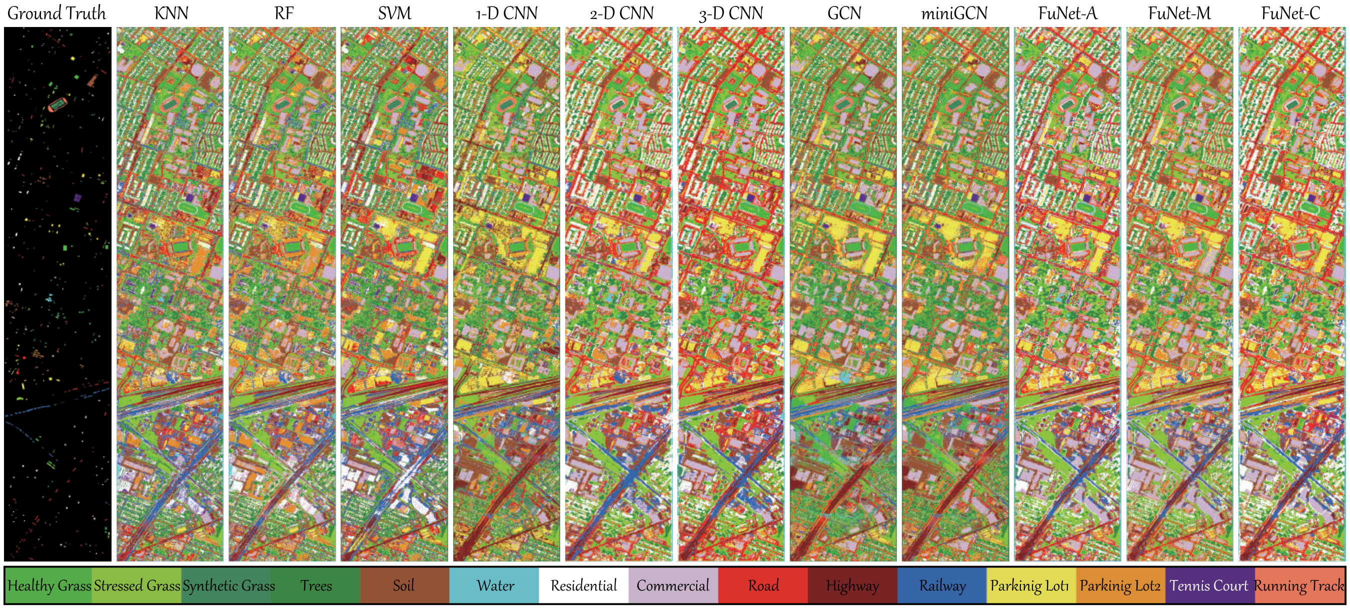


Fig. 9. Ground truth and classification maps obtained by different methods on the Houston2013 data set.

mainly due to the effective combination of different features that further enhance the HS representation ability. It should be noted, however, that the batchwise input in CNNs could lead to losing some edge details to some extent (e.g., 2-D CNN and 3-D CNN). This explains why the classification maps obtained by FuNets are not as sharp (in terms of edge delineation) as those obtained by only using miniGCNs.

V. CONCLUSION

Due to the embedding of graph (or topological) structure, GCNs can properly characterize the underlying data structure of HS images in high-dimensional space but inevitably introduce some drawbacks, e.g., high storage and computational cost when computing the adjacency matrix, gradient exploding or vanishing problems (due to full-batch network training) and the need to retrain these networks when new data are fed. In order to address these problems, in this article, we develop a new supervised version of GCNs, called miniGCNs, which allows us to train large-scale graph networks in a minibatch fashion. Due to their batchwise network training strategy, our newly proposed miniGCNs are more flexible, in the sense that they not only yield lower computational cost and stable local optima in the training phase but also can directly predict the new input samples, i.e., the out-of-sample cases, with no need to retrain the network. More significantly, our trainable minibatch strategy makes it possible to jointly use CNNs and GCNs for extracting more diverse and discriminative feature representations for the HS image classification task. To exploit this property, we have further investigated several fusion modules: A, M, and C that integrate CNNs and miniGCNs in an end-to-end trainable fashion. Our experimental results, conducted on three widely used HS data sets, demonstrate the effectiveness and superiority of our newly proposed miniGCNs compared to the traditional GCNs. Also, the FuNet (with different fusion strategies) has been shown to be superior to using single model (e.g., CNNs and miniGCNs).

In the future, we will investigate the possible combination of different deep networks and our miniGCNs and also develop more advanced fusion modules, e.g., weighted fusion, to fully exploit the rich spectral information contained in HS images.

Acknowledgment

The authors would like to the Hyperspectral Image Analysis Group, University of Houston, for providing the CASI University of Houston data sets and the IEEE GRSS DFC2013.

References

[1] J. Anderson, *A Land Use Land Cover Classification Systems for Use With Remote Sensor Data*, vol. 964. Washington, DC, USA: US Government Printing Office, 1976.

[2] B. Rasti *et al.*, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep," *IEEE Geosci. Remote Sens. Mag.*, early access, Apr. 29, 2020, doi: 10.1109/MGRS.2020.2979764.

[3] J. Kang, D. Hong, J. Liu, G. Baier, N. Yokoya, and B. Demir, "Learning convolutional sparse coding on complex domain for interferometric phase restoration," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 9, 2020, doi: 10.1109/TNNLS.2020.2979546.

[4] R. Huang, D. Hong, Y. Xu, W. Yao, and U. Stilla, "Multi-scale local context embedding for LiDAR point cloud classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 721–725, Apr. 2020.

[5] J. M. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[6] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[7] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[8] J. Peng and Q. Du, "Robust joint sparse representation based on maximum correntropy criterion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7152–7164, Dec. 2017.

[9] J. Peng, W. Sun, and Q. Du, "Self-paced joint sparse representation for the classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1183–1194, Feb. 2019.

[10] S. Liu, Q. Du, X. Tong, A. Samat, and L. Bruzzone, "Unsupervised change detection in multispectral remote sensing images via spectral-spatial band expansion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3578–3587, Sep. 2019.

[11] D. Hong, N. Yokoya, J. Chanussot, J. Xu, and X. X. Zhu, "Learning to propagate labels on graphs: An iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 35–49, Dec. 2019.

[12] S. Liu, D. Marinelli, L. Bruzzone, and F. Bovolo, "A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 140–158, Jun. 2019.

[13] L. Wang, J. Peng, and W. Sun, "Spatial–spectral squeeze-and-excitation residual network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 7, p. 884, Apr. 2019.

[14] A. Samat, E. Li, W. Wang, S. Liu, C. Lin, and J. Abuduwaili, "Meta-XGBoost for hyperspectral image classification using extended MSER-guided morphological profiles," *Remote Sens.*, vol. 12, no. 12, p. 1973, Jun. 2020.

[15] D. Hong, N. Yokoya, G.-S. Xia, J. Chanussot, and X. X. Zhu, "X-ModalNet: A semi-supervised deep cross-modal network for classification of remote sensing data," *ISPRS J. Photogramm. Remote Sens.*, vol. 167, pp. 12–23, Sep. 2020.

[16] 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.

[17] 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.

[18] A. Samat, C. Persello, S. Liu, E. Li, Z. Miao, and J. Abuduwaili, "Classification of VHR multispectral images using ExtraTrees and maximally stable extremal region-guided morphological profile," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 9, pp. 3179–3195, Sep. 2018.

[19] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3747–3762, Oct. 2010.

[20] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[21] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.

[22] 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.

[23] L. Gao, D. Hong, J. Yao, B. Zhang, P. Gamba, and J. Chanussot, "Spectral superresolution of multispectral imagery with joint sparse and low-rank learning," *IEEE Trans. Geosci. Remote Sens.*, early access, Jun. 18, 2020, doi: 10.1109/TGRS.2020.3000684.

[24] D. Hong, N. Yokoya, and X. X. Zhu, "Learning a robust local manifold representation for hyperspectral dimensionality reduction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2960–2975, Jun. 2017.

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

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

[27] 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.

[28] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 12, p. 1330, Dec. 2017.

[29] H. Wu and S. Prasad, "Convolutional recurrent neural networks forhyperspectral data classification," *Remote Sens.*, vol. 9, no. 3, p. 298, Mar. 2017.

[30] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[31] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, early access, Jun. 25, 2020, doi: 10.1109/TGRS.2020.3003341.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[33] F. F. Shahraki and S. Prasad, "Graph convolutional neural networks for hyperspectral data classification," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Nov. 2018, pp. 968–972.

[34] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Yan 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.

[35] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," 2019, *arXiv:1909.11953*. [Online]. Available: http://arxiv.org/abs/1909.11953

[36] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "CoSpace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, Jul. 2019.

[37] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 193–205, Jan. 2019.

[38] F. R. Chung and F. C. Graham, *Spectral Graph Theory*, vol. 92. Providence, RI, USA: American Mathematical Society, 1997.

[39] C. D. McGillem and G. R. Cooper, *Continuous and Discrete Signal and System Analysis*. London, U.K.: Oxford Univ. Press, 1991.

[40] 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.

[41] R. S. Michalski, "A theory and methodology of inductive learning," *Mach. Learn.*, vol. 110, pp. 83–134, Oct. 1983.

[42] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "GraphSAINT: Graph sampling based inductive learning method," 2019, *arXiv:1907.04931*. [Online]. Available: http://arxiv.org/abs/1907.04931

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

Danfeng Hong (Member, IEEE) received the M.Sc. degree *(summa cum laude)* in computer vision from the College of Information Engineering, Qingdao University, Qingdao, China, in 2015, and the Dr.-Ing degree *(summa cum laude)* from Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Munich, Germany, in 2019.

Since 2015, he has been a Research Associate with the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, Germany. He is a Research Scientist and leads a Spectral Vision Working Group, IMF, DLR, and also an Adjunct Scientist with GIPSA-lab, Grenoble INP, CNRS, Univ. Grenoble Alpes, Grenoble, France. His research interests include signal/image processing and analysis, hyperspectral remote sensing, machine/deep learning, and artificial intelligence and their applications in earth vision.

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

He is a Professor with the Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, CAS. He has also been a Visiting Scholar with the University of Extremadura, Cáceres, Spain, in 2014, and with Mississippi State University (MSU), Starkville, MS, USA, in 2016. In the last ten years, he was the PI of ten scientific research projects at national and ministerial levels, including projects by the National Natural Science Foundation of China from 2010 to 2012, 2016 to 2019, and 2018 to 2020, and by the Key Research Program of the CAS from 2013 to 2015. He has published more than 160 peer-reviewed papers, and there are more than 80 journal papers included by SCI. He has coauthored an academic book, *Hyperspectral Image Classification and Target Detection*. He obtained 28 National Invention Patents in China. His research focuses on hyperspectral image processing and information extraction.

Dr. Gao was awarded the Outstanding Science and Technology Achievement Prize of the CAS in 2016 and was supported by the China National Science Fund for Excellent Young Scholars in 2017, and won the Second Prize of the State Scientific and Technological Progress Award in 2018. He received the recognition of the Best Reviewer of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing in 2015 and the IEEE Transactions on Geoscience and Remote Sensing in 2017.

Jing Yao received the B.Sc. degree from Northwest University, Xi'an, China, in 2014. He is pursuing the Ph.D. degree with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an.

From 2019 to 2020, he is a Visiting Student with Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Munich, Germany, and at the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, Germany. His research interests include low-rank modeling, hyperspectral image analysis, and deep learning-based image processing methods.

Bing Zhang (Fellow, IEEE) received the B.S. degree in geography from Peking University, Beijing, China, in 1991, and the M.S. and Ph.D. degrees in remote sensing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, in 1994 and 2003, respectively.

He is a Full Professor and the Deputy Director of the Aerospace Information Research Institute, CAS, where he has been leading lots of key scientific projects in the area of hyperspectral remote sensing for more than 25 years. He has authored more than 300 publications, including more than 170 journal papers. He has edited six books/contributed book chapters on hyperspectral image processing and subsequent applications. His creative achievements were rewarded ten important prizes from Chinese Government and special government allowances of the Chinese State Council. His research interests include the development of mathematical and physical models and image processing software for the analysis of hyperspectral remote sensing data in many different areas.

Dr. Zhang was a Student Paper Competition Committee Member in IGARSS from 2015 to 2019. He was awarded the National Science Foundation for Distinguished Young Scholars of China in 2013 and the 2016 Outstanding Science and Technology Achievement Prize of the Chinese Academy of Sciences, the highest level of Awards for the CAS scholars. His creative achievements were rewarded ten important prizes from Chinese Government and special government allowances of the Chinese State Council. He is serving as an Associate Editor for the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. He has been serving as a Technical Committee Member for the IEEE Workshop on Hyperspectral Image and Signal Processing since 2011, as the President of the Hyperspectral Remote Sensing Committee of the China National Committee of International Society for Digital Earth since 2012, and as the Standing Director of the Chinese Society of Space Research (CSSR) since 2016.

Antonio Plaza (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 1999 and 2002, respectively.

He is the Head of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. He has authored more than 600 publications, including over 200 JCR journal articles (over 160 in IEEE journals), 23 book chapters, and around 300 peer-reviewed conference proceeding papers. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza is a fellow of the IEEE for contributions to hyperspectral data processing and parallel computing of earth observation data. He was a member of the Editorial Board of the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER from 2011 to 2012 and the *IEEE Geoscience and Remote Sensing Magazine* in 2013. He was also a member of the Steering Committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). He received the recognition as a Best Reviewer of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2009 and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2010, for which he has served as an Associate Editor from 2007 to 2012. He was also a recipient of the Most Highly Cited Paper from 2005 to 2010 in the *Journal of Parallel and Distributed Computing*, the 2013 Best Paper Award of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), and the Best Column Award of the *IEEE Signal Processing Magazine* in 2015. He received the Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He has served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) from 2011 to 2012 and as the President of the Spanish Chapter of the IEEE GRSS from 2012 to 2016. He has reviewed more than 500 manuscripts for over 50 different journals. He has served as the Editor-in-Chief for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017. He has guestedited ten special issues on hyperspectral remote sensing for different journals. He is also an Associate Editor of the IEEE ACCESS (received the recognition as an Outstanding Associate Editor of the journal in 2017). Additional information can be found at http://www.umbc.edu/rssipl/people/aplaza

Jocelyn Chanussot (Fellow, IEEE) received the M.Sc. degree in electrical engineering from the Grenoble Institute of Technology (Grenoble INP), Grenoble, France, in 1995, and the Ph.D. degree from the Université de Savoie, Annecy, France, in 1998.

Since 1999, he has been with Grenoble INP, where he is a Professor of signal and image processing. He has been a Visiting Scholar at Stanford University, Stanford, CA, USA, KTH, Stockholm, Sweden, and NUS, Singapore. Since 2013, he has been an Adjunct Professor with the University of Iceland, Reykjavik, Iceland. From 2015 to 2017, he was a Visiting Professor at the University of California at Los Angeles (UCLA), Los Angeles, CA, USA. He holds the AXA Chair in remote sensing and is an Adjunct Professor at the Chinese Academy of Sciences, Aerospace Information Research Institute, Beijing, China. His research interests include image analysis, hyperspectral remote sensing, data fusion, machine learning, and artificial intelligence.

Dr. Chanussot was the founding President of the IEEE GEOSCIENCE AND REMOTE SENSING French Chapter from 2007 to 2010, which received the 2010 IEEE GRS-S Chapter Excellence Award. He was the Vice-President of the IEEE Geoscience and Remote Sensing Society in charge of meetings and symposia from 2017 to 2019. He has received multiple outstanding paper awards. He was the General Chair of the first IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing (WHISPERS). He was the Chair (2009–2011) and Co-Chair (2005–2008) of the GRS Data Fusion Technical Committee. He was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society from 2006 to 2008 and the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing in 2009. He is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the PROCEEDINGS OF THE IEEE. He was the Editor-in-Chief of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING from 2011 to 2015. In 2014, he served as a Guest Editor for the *IEEE Signal Processing Magazine*. He is a member of the Institut Universitaire de France from 2012 to 2017 and a Highly Cited Researcher of the Clarivate Analytics/Thomson Reuters from 2018 to 2019.