Nonlocal Graph Convolutional Networks for Hyperspectral Image Classification

Lichao Mou, *Student Member, IEEE*, Xiaoqiang Lu, *Senior Member, IEEE*, Xuelong Li, *Fellow, IEEE*, and Xiao Xiang Zhu, *Senior Member, IEEE*

***Abstract*— Over the past few years making use of deep networks, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), classifying hyperspectral images has progressed significantly and gained increasing attention. In spite of being successful, these networks need an adequate supply of labeled training instances for supervised learning, which, however, is quite costly to collect. On the other hand, unlabeled data can be accessed in almost arbitrary amounts. Hence it would be conceptually of great interest to explore networks that are able to exploit labeled and unlabeled data simultaneously for hyperspectral image classification. In this article, we propose a novel graph-based semisupervised network called nonlocal graph convolutional network (nonlocal GCN). Unlike existing CNNs and RNNs that receive pixels or patches of a hyperspectral image as inputs, this network takes the whole image (including both labeled and unlabeled data) in. More specifically, a nonlocal graph is first calculated. Given this graph representation, a couple of graph convolutional layers are used to extract features. Finally, the semisupervised learning of the network is done by using a cross-entropy error over all labeled instances. Note that the nonlocal GCN is end-to-end trainable. We demonstrate in extensive experiments that compared with state-of-the-art spectral classifiers and spectral–spatial classification networks, the nonlocal GCN is able to offer competitive results and high-quality classification maps (with fine boundaries and without noisy scattered points of misclassification).**

***Index Terms*— Graph convolutional network (GCN), hyperspectral image classification, nonlocal graph, semisupervised learning.**

I. Introduction

HYPERSPECTRAL images can be used to differentiate various materials of interest by their abundant spectral bands. Hence, hyperspectral data classification has become an active and crucial research topic in the remote sensing community, and so far, a wide range of applications have benefited from the development of this direction, to name a few, urban planning, agriculture monitoring, and disaster relief operations.

To achieve better classification results, plenty of approaches have been developed over the past decades. On the one hand, some efforts have explored more discriminative feature representations, such as morphological features and texture features [1], [2]. Apart from these handcrafted features, subspace learning and sparse learning algorithms have also gotten much attention in the community. These methods mainly concentrate on transforming original spectral signatures into a learned, new feature space [3]–[5]. On the other hand, better classifiers from machine learning have also provided new insights for hyperspectral image classification, for instance, random forest and support vector machine (SVM).

Deep learning, which is mainly characterized by deep networks, has been quite successful in solving a wide range of problems (e.g., natural language processing [6]–[8], computer vision [9]–[19], and remote sensing [20], [21]). Recently, hyperspectral data classification has been approached by means of convolutional neural networks (CNNs) [20], [22]–[27] as well as recurrent neural networks (RNNs) [28]–[30]. At first, a very simple 1-D CNN, including only one convolution layer, has been investigated in [22]. Makantasis *et al.* [23] made use of a 2-D CNN to perform spectral–spatial classification. In [24], on the classification problem of crop types, the authors compare the performance of 1-D and 2-D CNNs and conclude that the 2-D CNN is superior to the 1-D CNN, but several tiny objects in the former's classification map are a little oversmoothed and misclassified. Following recent developments in 3-D CNN for video analysis [31] where the third dimensionality is usually the time axis, 3-D CNNs have also been studied in hyperspectral data classification. Chen *et al.* [25] introduced an ℓ_2 regularized 3-D CNN for learning spectral–spatial features, whereas [26] follows a similar idea for the purpose of hyperspectral data classification. Furthermore, better CNN architectures from computer vision, e.g., ResNet [13] and DenseNet [14], also provide new insights for this task [32]–[34].

Given that a pixel of a hyperspectral image can be deemed as an orderly spectra sequence in the spectral domain, RNNs are natural candidates to tackle such sequential data. A first, attempt in this direction can be found in [28], where an RNN

Manuscript received July 16, 2019; revised November 21, 2019; accepted December 9, 2019. Date of publication May 12, 2020; date of current version November 24, 2020. This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (So2Sat) under Grant ERC-2016-StG-714087, and in part by the Helmholtz Association through the Framework of Helmholtz Artificial Intelligence Cooperation Unit (HAICU) - Local Unit "Munich Unit @Aeronautics, Space and Transport (MASTr)" and Helmholtz Excellent Professorship "Data Science in Earth Observation - Big Data Fusion for Urban Research." *(Corresponding author: Xiao Xiang Zhu.)*

Lichao Mou and Xiao Xiang Zhu are with Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Wessling 82234, Germany, and also with Signal Processing in Earth Observation (SiPEO), Technical University of Munich (TUM), Munich 80333, Germany (e-mail: lichao.mou@dlr.de; xiaoxiang.zhu@dlr.de).

Xiaoqiang Lu is with the Key Laboratory of Spectral Imaging Technology CAS, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: luxq666666@gmail.com).

Xuelong Li is with the School of Computer Science, Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: xuelong_li@nwpu.edu.cn).

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

Digital Object Identifier 10.1109/TGRS.2020.2973363

model equipped with a new activation function and a modified gated recurrent unit is proposed for spectral classification. Wu and Prasad [29] proposed a hybrid convolutional and recurrent network, in which a couple of convolutional layers first learn midlevel feature representations, and the following recurrent layers are then used to model spectral contexts.

The aforementioned networks are both trained in a supervised fashion via backpropagation. In spite of the great success of the supervised networks, there is a technical hurdle in the application of supervised CNNs [20], [22]–[27], [35], [36] or RNNs [28]–[30] to hyperspectral data classification tasks: an adequate supply of manually annotated training samples as fuel. However, different hyperspectral imaging sensors, complicated atmospheric scattering conditions, and various categories of interest in different applications result in collecting a large, labeled data set such as ImageNet in computer vision for hyperspectral image classification being difficult. Also, making the labeled data set larger and larger has diminishing returns. In this case, it would be conceptually of great interest to explore how to access arbitrary amounts of unlabeled data.

Unsupervised feature learning, which is capable of learning useful, informative feature representations from unlabeled samples, is a solution and has attracted extensive attention in the community. For instance, in a pioneer work [37], the authors present an unsupervised CNN, and its weights are estimated via a sparse learning algorithm in a greedy layerwise fashion. Mou *et al.* [32] devised a residual learning-based unsupervised conv–deconv network, which is trained end-to-end by learning an identical mapping. Once these unsupervised networks are well-trained using unlabeled instances, they can be fine-tuned by a small amount of labeled data for hyperspectral image classification tasks. However, in these models, labeled and unlabeled data are separately involved in two stages, which fails to access the relationship between them. Hence, a question arouses our curiosity: can a network be trained in a supervised way with labeled and unlabeled instances simultaneously for the problem of hyperspectral image classification?

Graph-based semisupervised learning [38]–[40] is possible to provide a solution to the problem by harnessing the graph or manifold structure of data. The cluster assumption is widely used in most graph-based semisupervised learning approaches, and it assumes that nearby vertices on the same graph are apt to share the same class. Nevertheless, directly applying conventional networks (e.g., CNNs and RNNs) to a graph is quite challenging. Fortunately, several recent studies in machine leaning (see Section II) make convolutions on graphs possible. Now, graph convolutional networks (GCNs), which generalize convolutions to graphs of arbitrary structures, have gained increasing attention and have successfully been applied to a number of natural language processing (NLP) tasks. However, using GCNs to classify hyperspectral images has rarely been addressed so far. In this article, inspired by recent advances in GCNs and the nonlocal idea in vision tasks [41]–[43], we propose a semisupervised nonlocal GCN for hyperspectral data classification tasks. The network first represents the whole hyperspectral image as a nonlocal graph where each vertex in the graph represents a pixel in the image. Given the graph representation, we perform reasoning on the graph and infer the classification map of the whole image by applying graph convolutions. Note that the whole network is end-to-end trainable. The contributions of this article are threefold.

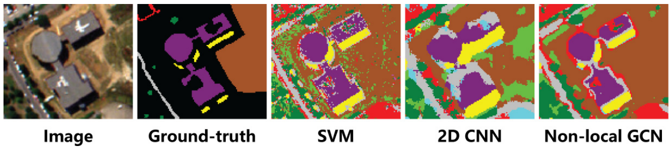


Fig. 1. Examples illustrating the limitations of spectral classifiers (e.g., SVM) and spectral–spatial classification networks (e.g., 2-D CNN). Classification map produced by SVM suffers from a salt-and-pepper effect, while 2-D CNN produces oversmoothed results. In contrast, our GCN framework-based method not only removes scattered points of misclassification but also preserves edge information well in classification results.

1) We perform hyperspectral image classification via a graph-based semisupervised network. Unlike existing networks such as CNNs and RNNs, which receive local portions of an image (e.g., pixels and patches) as inputs, our network takes the whole hyperspectral image in.
2) Unlike CNNs whose receptive fields are local regions in an image, the proposed method uses a nonlocal, data-driven graph representation for hyperspectral image classification tasks.
3) We carry out experiments on three benchmark data sets, and empirical results show the competitive performance of our network. Moreover, our network can offer higher quality classification maps (see Fig. 1).

The remainder of this article is organized as follows. After detailing deep learning in hyperspectral image classification in Section I. Section II briefly introduces GCNs. Section III details the proposed nonlocal GCN. Section IV verifies the proposed approach and presents the corresponding analysis and discussion. Finally, Section V concludes this article.

II. Preliminaries and Related Works

Several efforts have been made in machine learning for generalizing networks to graph data structures. In this section, we recall the basic principles of these works. The graph networks involve both CNNs and RNNs, but this work is more related to the former, i.e., GCNs. First, some notations used throughout this article are given. We consider an undirected graph, which can be encoded by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$. $\mathcal{V}$ denotes the vertex set with $|\mathcal{V}| = N$, and $\mathcal{E}$ is the edge set of the graph. $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix, where if there is an edge between vertex i and vertex j, entry a_{ij} represents the weight of the edge.

A. Graph Convolution From Spectral Perspective

We denote the diagonal degree matrix of $\boldsymbol{A}$ as $\boldsymbol{D}$, whose entry $d_{ii} = \sum_{j=1}^{N} a_{ij}$. Then, the Laplacian matrix of a graph $\mathcal{G}$ can be defined as

$$\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}. \tag{1}$$

The corresponding symmetrically normalized Laplacian matrix is as follows:

$$\boldsymbol{L}_{\text{sym}} := \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \tag{2}$$

where $\boldsymbol{I}$ is an identity matrix.

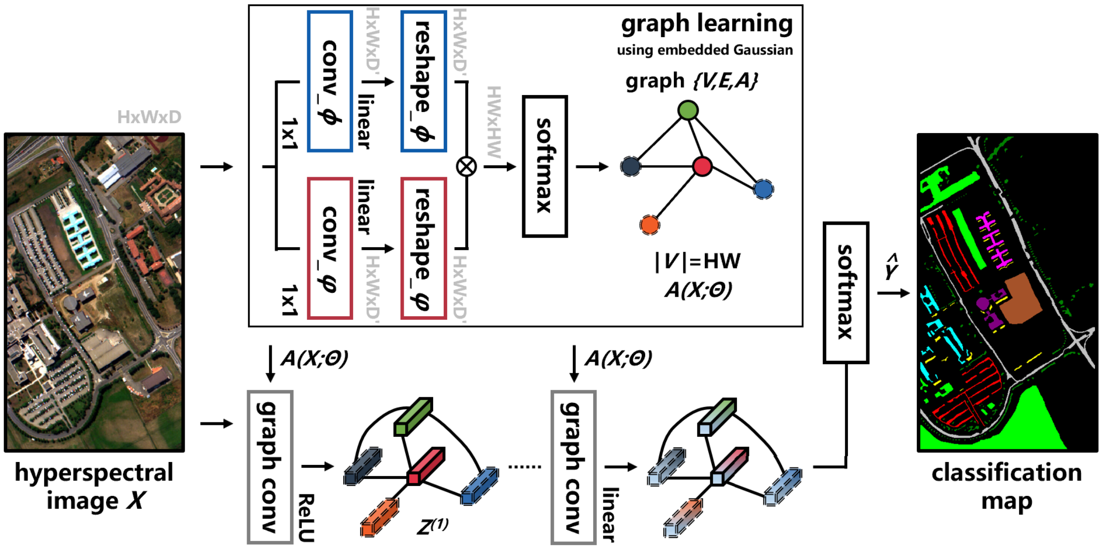


Fig. 2. Network overview. Unlike 1-D or 2-D CNNs which take as input pixels or patches, our network takes the whole hyperspectral image as input and performs information propagation by a couple of graph convolutions based on a learned nonlocal graph. Finally, a cross-entropy error over all labeled instances is exploited for semisupervised classification.

Given a graph signal $\boldsymbol{s} \in \mathbb{R}^N$ (a scalar for each node) and a filter $g_{\boldsymbol{\theta}} = \mathrm{diag}(\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^N$, the spectral convolution of $\boldsymbol{s}$ and $g_{\boldsymbol{\theta}}$ can be performed by decomposing $\boldsymbol{s}$ on the spectral domain and then multiplying each frequency by $g_{\boldsymbol{\theta}}$ [44]–[46]

$$g_{\boldsymbol{\theta}} \star \boldsymbol{s} = \boldsymbol{U} g_{\boldsymbol{\theta}} \boldsymbol{U}^{\mathrm{T}} \boldsymbol{s} \tag{3}$$

where $\boldsymbol{U}$ is the matrix of eigenvectors of $\boldsymbol{L}_{\mathrm{sym}}$ and can be computed by $\boldsymbol{L}_{\mathrm{sym}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}$. $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues of $\boldsymbol{L}_{\mathrm{sym}}$. In addition, $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{s}$ denotes the graph Fourier transform of $\boldsymbol{s}$. $g_{\boldsymbol{\theta}}$ can be framed as a function of the eigenvalues of $\boldsymbol{\Lambda}$, i.e., $g_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$.

However, note that evaluating (3) requires explicitly calculating the Laplacian eigenvectors, which is not computationally feasible for large graphs. To circumvent this problem, a possible way is to approximate the filter $g_{\boldsymbol{\theta}}$ by the Chebyshev polynomials up to the Kth order. Hence, Hammond *et al.* [47] proposed the following K-localized convolution on graphs:

$$g_{\boldsymbol{\theta}} \star \boldsymbol{s} \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k(\boldsymbol{L}_{\mathrm{sym}})\boldsymbol{s} \tag{4}$$

where T_k is the Chebyshev polynomials.

Recently, Kipf and Welling [48] simplified (4) by limiting $K = 1$ and further approximating the largest eigenvalue $\lambda_{\max} \approx 2$. By doing so, (4) can be rewritten as

$$g_{\boldsymbol{\theta}} \star \boldsymbol{s} \approx \boldsymbol{\theta}(\boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{s}. \tag{5}$$

Then, they consider a GCN with the following propagation rule:

$$\boldsymbol{H}^{(l+1)} = \sigma(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}) \tag{6}$$

where $\boldsymbol{H}^{(l)}$ and $\boldsymbol{H}^{(l+1)}$ are the input and output, respectively, and $\boldsymbol{W}^{(l)}$ represents the weights.

Our work follows this spectral stream, particularly the work of [48].

B. Graph Convolution From Spatial Perspective

Spatial graph convolution defines the convolutions directly on graph vertexes and their neighbors. However, one challenge for such methods is coming up with a way to handle different sized neighbors for each vertex. The interested reader is referred to [49]–[51] for more details of these algorithms.

III. Methodology

Our goal is to represent the whole hyperspectral image (including labeled and unlabeled data) as a holistic graph and perform reasoning on the graph for semisupervised classification. Fig. 2 shows the overview architecture of the proposed network.

A. Graph Representation in Hyperspectral Images

In a pioneer work [52] of making use of GCNs for hyperspectral data classification, the authors employ a fixed graph $\mathcal{G}$. However, regarding the construction of the graph, we have the following observations.

1) In airborne or spaceborne hyperspectral images, the classification of a pixel probably benefits from remote pixels instead of only its neighbors (see Fig. 3). Hence, we think that for classifying hyperspectral data, the connection relationship among the graph's vertices should not be constrained to adjacent nodes.
2) Hyperspectral images have intrinsic intraclass variations (samples in the same category may have different spectral signatures) and interclass similarities (samples in different classes may share similar spectral signatures), which means that there exists a semantic gap between spectral information and high-level semantics. In this case, a learnable graph, which helps to narrow the gap, would be more desired than a fixed graph.
3) Some graph-based learning algorithms compute the edges of a graph based on original spectra directly.

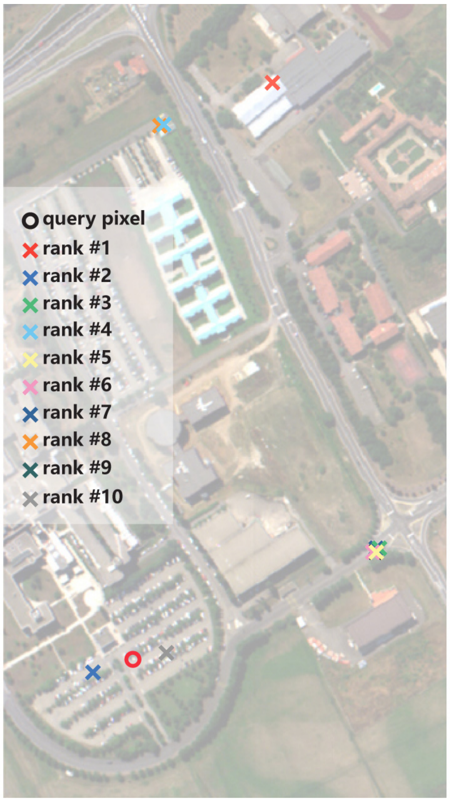


Fig. 3. Illustration of nonlocal self-similarity in a hyperspectral image. For a query pixel, the ten most similar pixels in the image are plotted. This shows us that the classification of a pixel probably benefits from remote pixels instead of only its neighbors. Note that the Gaussian distance is used to calculate the similarities among the pixels.

This way, however, is sensitive to spectral signature changes and easily affected by intraclass variations and interclass similarities of hyperspectral images. Therefore, doing so in a learnable latent space may be a better option.

Therefore, we hope that the graph structure in our model is nonlocal, data-driven, and can be adaptively learned in an end-to-end way. To this end, we construct the graph by measuring similarities between vertices (including labeled and unlabeled pixels) in a feature space. In this graph, a high confidence edge between two vertices indicates that: 1) the two pixels belong to the same category or 2) they are highly correlated for recognizing their labels. Note that the edges of this graph are computed between any pair of vertices.

Formally, denote by $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ all pixels in a hyperspectral image, where N represents the number of pixels, and each pixel $\boldsymbol{x}_i$ is a D-dimensional vector, where D is the number of bands. In our method, the pairwise similarity between every two hyperspectral pixels $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be modeled as

$$S(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \varphi(\boldsymbol{x}_j) \tag{7}$$

where ϕ and φ indicate two individual transformations, which map original spectral features to latent feature spaces.

Once the edges of the nonlocal graph are computed, a normalization is performed on each row of the adjacency matrix so that the sum of all edge values related to the same instance i is 1. Here, we make use of a softmax function for the purpose of normalization

TABLE I

IMPLEMENTATION OF THE GRAPH LEARNING MODULE USING THE EMBEDDED GAUSSIAN

Layer	Configuration	Output Shape	Connected to
conv_ϕ reshape_ϕ	$1 \times 1, 256$ -	$H \times W \times 256$ $HW \times 256$	input conv_ϕ
conv_φ reshape_φ	$1 \times 1, 256$ -	$H \times W \times 256$ $HW \times 256$	input conv_φ
multiply	-	$HW \times HW$	reshape_ϕ reshape_φ
softmax	-	$HW \times HW$	multiply

$$a_{ij} = \frac{\exp(S(\boldsymbol{x}_i, \boldsymbol{x}_j))}{\sum_{j=1}^{N} \exp(S(\boldsymbol{x}_i, \boldsymbol{x}_j))}. \tag{8}$$

The normalized $\boldsymbol{A} = \{a_{ij}\}$ is taken as the final form of the adjacent matrix in our model.

B. Instantiations of ϕ and φ

As shown in (7), ϕ and φ are of importance for constructing the graph in the proposed model. Next, we discuss how to instantialize them. More specifically, we consider two ways.

1) *Embedded Dot Product*: An embedded dot product similarity can be used to define the pairwise similarity function S as follows:

$$S(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{W}_{\phi}\boldsymbol{x}_i)^{\mathrm{T}}(\boldsymbol{W}_{\varphi}\boldsymbol{x}_j) \tag{9}$$

where $\boldsymbol{W}_{\phi}\boldsymbol{x}_i$ and $\boldsymbol{W}_{\varphi}\boldsymbol{x}_j$ are two embeddings.

2) *Embedded Gaussian*: Another way to compute similarities in a latent feature space is to harness the following embedded Gaussian function:

$$S(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp((\boldsymbol{W}_{\phi}\boldsymbol{x}_i)^{\mathrm{T}}(\boldsymbol{W}_{\varphi}\boldsymbol{x}_j)). \tag{10}$$

Note that a special case (nonembedded version) of the embedded Gaussian function is the conventional Gaussian function, which is parameter-free and written as follows:

$$S(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j). \tag{11}$$

As discussed earlier, the construction of the nonlocal graph in our model is flexible, and we believe that more alternative ways are possible and may be able to offer better performance in the future.

In addition, regarding the implementation of this nonlocal, data-driven graph, a traversing method for computing pairwise similarity between every two hyperspectral pixels is obviously not computationally feasible in a network. Therefore, we need a doable way to wrap (9) or (10) into the form of a network block. Table I shows the implementation of a graph learning block.

C. Graph Convolution

Here, we describe how to perform graph convolutions on the learnable graph described in the previous sections. Unlike conventional convolutions in CNNs, which operate on a regular, local grid, graph convolutions on a graph make it possible to allow every vertex to attend on every other vertex on the graph. Hence, the process of graph convolutions in our case can be deemed as a message passing inside the whole hyperspectral image. The outputs of a graph convolution layer are convolved features of each vertex. The forward propagation rule of graph convolutions is as follows:

$$\boldsymbol{Z}^{(l+1)} = \sigma(\boldsymbol{A}(\boldsymbol{X};\boldsymbol{\Theta})\boldsymbol{Z}^{(l)}\boldsymbol{W}^{(l+1)}) \tag{12}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{W}_{phi}, \boldsymbol{W}_{\varphi}\}$.

Here, $\boldsymbol{Z}^{(0)} = \boldsymbol{X}$. $\boldsymbol{W}^{(l+1)}$ denotes layer-specific learnable weights, and $\sigma(\cdot)$ represents an activation function (we use ReLU).

From (12), we can see that a graph convolution layer actually includes two steps: 1) generating a new feature representation from the input $\boldsymbol{Z}^{(l)}$ by performing a graph convolution, i.e., $\boldsymbol{A}(\boldsymbol{X};\boldsymbol{\Theta})\boldsymbol{Z}^{(l)}$ and 2) feeding the new generated feature $\boldsymbol{A}(\boldsymbol{X};\boldsymbol{\Theta})\boldsymbol{Z}^{(l)}$ to a fully connected layer. To figure out the unique asset of graph convolutions, we compare them with fully connected layers, in which the layerwise forward propagation rule is

$$\boldsymbol{Z}^{(l+1)} = \sigma(\boldsymbol{Z}^{(l)}\boldsymbol{W}^{(l+1)}). \tag{13}$$

From (12) and (13), we can see that their difference is an adjacency matrix applied on the left of $\boldsymbol{Z}^{(l)}$. The benefit that this matrix brings is Laplacian smoothing, which calculates new features of a vertex as a weighted average of features of its neighbors on a graph. Given that vertices in the same cluster are more likely to be densely connected, Laplacian smoothing allows them to have similar features, which makes the subsequent classifications much easier.

D. Semisupervised Classification

We make use of a softmax on output features of the last graph convolutional layer, that is

$$\hat{\boldsymbol{Y}} = \mathrm{softmax}(\boldsymbol{A}\boldsymbol{Z}^{(L-1)}\boldsymbol{W}^{(L)}). \tag{14}$$

For semisupervised classification tasks of hyperspectral images, we exploit the following loss function:

$$\mathcal{L} := -\sum_{i\in\mathcal{V}_l}\sum_{c=1}^{C}\boldsymbol{Y}_{ic}\ln\hat{\boldsymbol{Y}}_{ic} \tag{15}$$

where $\mathcal{V}_l$ is a set of indices of labeled instances and C is the number of classes that is also the dimension of output features. It can be seen from this equation that a cross-entropy loss is calculated over all labeled instances. By training, the network learns a message passing mechanism that is capable of propagating labels from labeled instances to unlabeled samples.

TABLE II

AMOUNTS OF TRAINING AND TEST DATA OF THE PAVIA UNIVERSITY DATA SET

Class No.	Class Name	Training	Test
1	Asphalt	548	6631
2	Meadows	540	18649
3	Gravel	392	2099
4	Trees	524	3064
5	Metal sheets	265	1345
6	Bare Soil	532	5029
7	Bitumen	375	1330
8	Bricks	514	3682
9	Shadows	231	947
	TOTAL	3921	42776

TABLE III

AMOUNTS OF TRAINING AND TEST DATA OF THE INDIAN PINES SCENE

Class No.	Class Name	Training	Test
1	Alfalfa	50	1384
2	Corn-notill	50	784
3	Corn-min	50	184
4	Corn	50	447
5	Grass-pasture	50	697
6	Grass-trees	50	439
7	Grass-pasture-mowed	50	918
8	Hay-windrowed	50	2418
9	Oats	50	564
10	Soybean-notill	50	162
11	Soybean-mintill	50	1244
12	Soybean-clean	50	330
13	Wheat	50	45
14	Woods	15	39
15	Buildings-grass-trees	15	11
16	Stone-steel-towers	15	5
	TOTAL	695	9671

IV. EXPERIMENTS AND ANALYSIS

A. Data Description

1) Pavia University Hyperspectral Data Set: The first data set was acquired over the city of Pavia, Italy, 2002, by an airborne instrument—Reflective Optics Spectrographic Imaging System (ROSIS). The aircraft was operated by the German Aerospace Center (DLR) within the context of the European Union-funded HySens project. The data set is made up of 640 × 340 pixels with a 1.3-m/pixel spatial resolution and 103 bands covering from 430 to 860 nm after removing 12 noisy channels. Besides unknown pixels, nine classes are manually annotated in the reference data. Table II shows the information about all nine categories.

2) Indian Pines Hyperspectral Data Set: The second data were collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over Northwestern Indiana, USA, 1992. It includes 145 × 145 pixels with a 20-m/pixel spatial resolution and 200 spectral bands covering from 400 to 2500 nm after removing 20 water absorption channels (220, 150–163, and 104–108). The ground truth includes 16 classes of interest, which are mostly various crops in different growth phases and detailed in Table III (black color in the ground truth indicates unknown samples). Since these 16 classes have

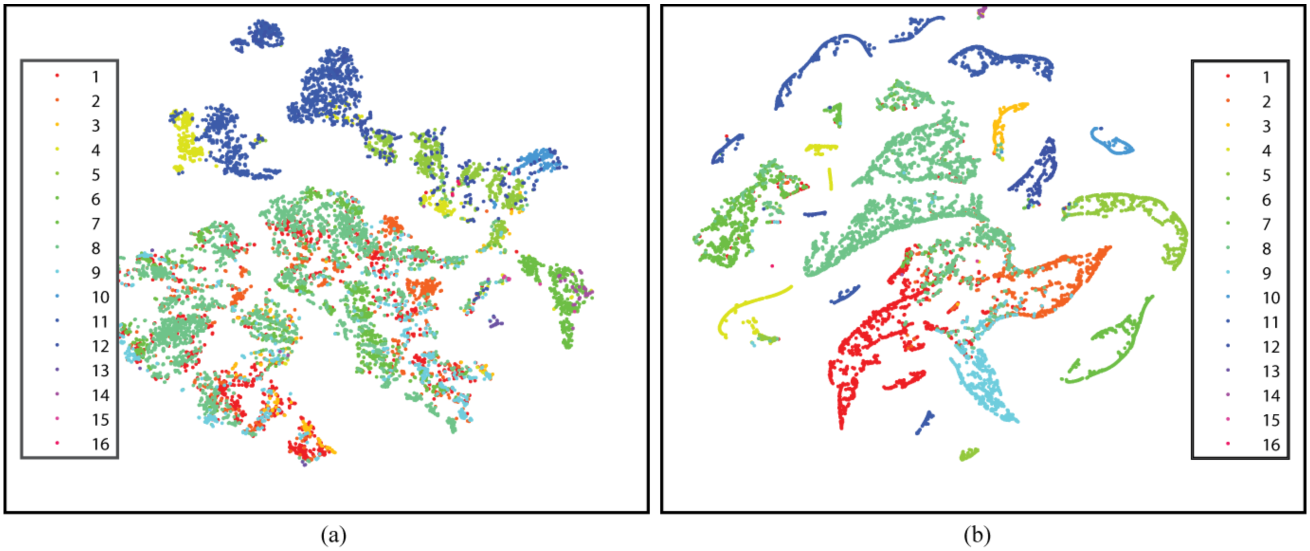


Fig. 4. (a) Visualization of original spectra and (a) outputs of the last graph convolution of the nonlocal GCN on the Indian Pines data set by t-SNE [56].

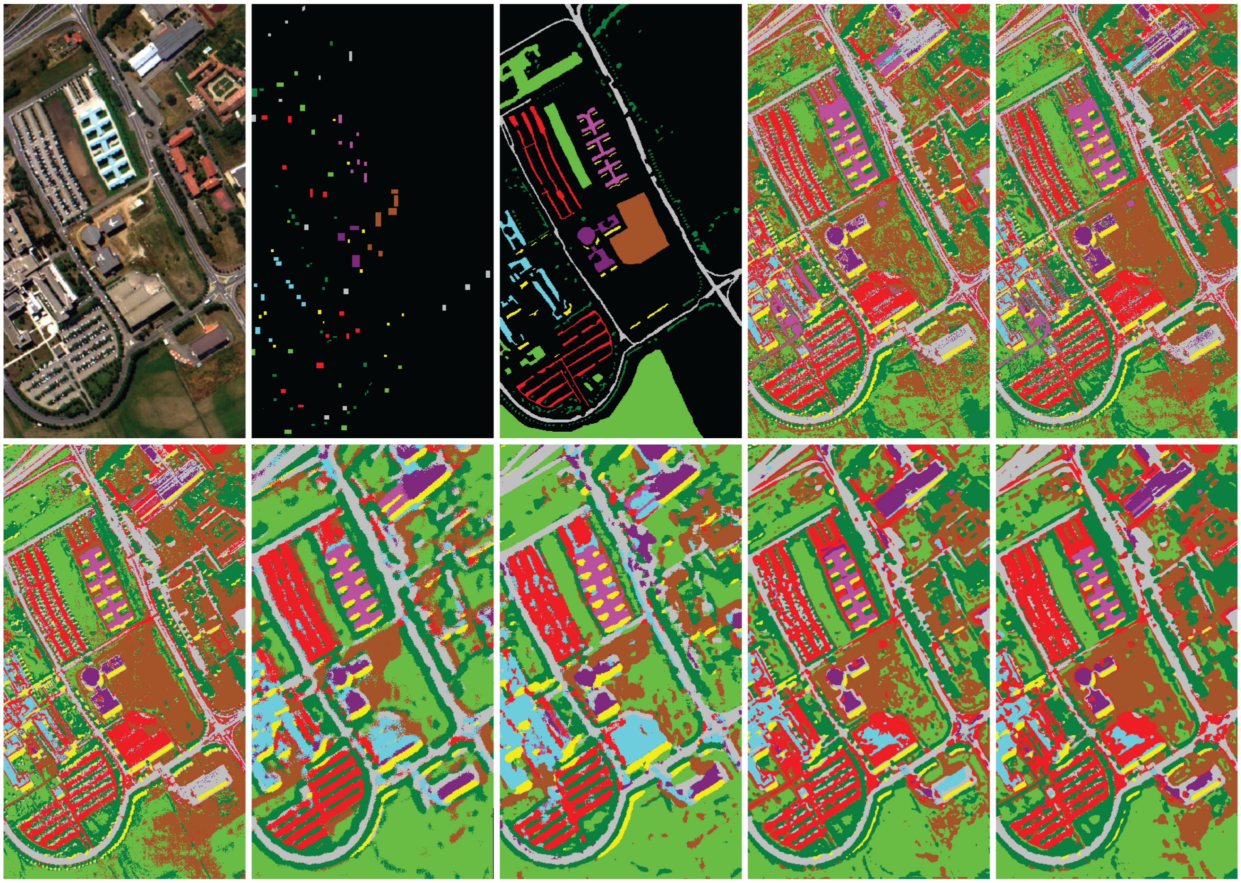


Fig. 5. Classification maps of different approaches for the Pavia University data set. (From Left to Right and Top to Bottom) Composite image, training samples, ground truth, RF-200, SVM-RBF, CCF-200, SICNN, 2-D CNN, GCN, and nonlocal GCN.

similar spectral signatures, the precise classification of this scene is hard.

3) Salinas Hyperspectral Data Set: The third data set was also gathered by the AVIRIS sensor over the region of Salinas Valley, CA, USA, and with a 3.7-m/pixel spatial resolution. The Salinas scene is composed of 224 spectral bands and 512 × 217 pixels. Like the Indian Pines data set, 20 water absorption bands (224, 154–167, and 108–112) of the Salinas scene have been discarded. The data set presents 16 classes related to vegetables, vineyard fields, and bare soils. Table IV shows the amounts of training and test data of this data set.

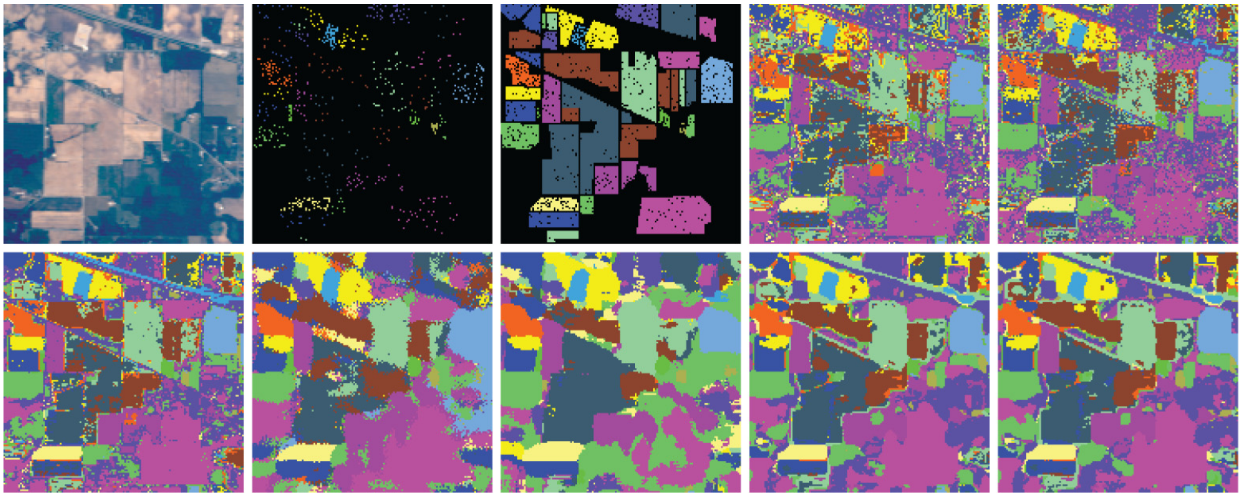


Fig. 6. Classification maps of different approaches for the Indian Pines data set. (From Left to Right and Top to Bottom) True-color composite image, training set, test set, RF-200, SVM-RBF, CCF-200, SICNN, 2-D CNN, GCN, and nonlocal GCN.

TABLE IV

AMOUNTS OF TRAINING AND TEST DATA OF THE SALINAS DATA

Class No.	Class Name	Training	Test
1	Brocoli_green_weeds_1	50	1959
2	Brocoli_green_weeds_2	50	3676
3	Fallow	50	1926
4	Fallow_rough_plow	50	1344
5	Fallow_smooth	50	2628
6	Stubble	50	3909
7	Celery	50	3529
8	Grapes_untrained	50	11221
9	Soil_vinyard_develop	50	6153
10	Corn_senesced_green_weeds	50	3228
11	Lettuce_romaine_4wk	50	1018
12	Lettuce_romaine_5wk	50	1877
13	Lettuce_romaine_6wk	50	866
14	Lettuce_romaine_7wk	50	1020
15	Vinyard_untrained	50	7218
16	Vinyard_vertical_trellis	50	1757
	TOTAL	800	53329

B. Experiment Setup

To quantitatively compare different models for hyperspectral data classification tasks from various aspects, the following measurements are considered.

1) *Overall Accuracy (OA):* This criterion is calculated as the fraction of test samples that are differentiated correctly.
2) *Per-Class Accuracy:* To access the performance with respect to each category in a data set, we also compute per-class accuracy. This measurement is particularly useful when the class labels are not uniformly distributed.
3) *Average Accuracy (AA):* This criterion is computed as the average of all per-class accuracies.
4) *Kappa Coefficient:* This statistic criterion is a robustness measurement with the degree of agreement.

If the number of samples for each category is identical, OA and AA are equal. However, the category distribution suffers from an imbalanced phenomenon in practice. Adopting OA alone is not precise since rare categories are commonly ignored. Therefore, AA is also utilized to evaluate the performance of different classification models. Strong differences between the OA and AA may indicate that a specific class is incorrectly classified with a high proportion. In addition, the kappa coefficient is generally thought to be a more robust measure than a simple percent agreement calculation.

Furthermore, we make use of a statistical test to validate the significance of classification accuracies produced by various methods. Given that samples used for two classification models are not independent, McNemar's test can be harnessed to estimate the significance of the difference of two classification maps, and McNemar's test can be performed by

$$z_{12} = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \tag{16}$$

where f_{ij} is the amount of data correctly recognized by method i and incorrectly recognized by j. McNemar's test is a statistical test for paired nominal data, and we can use McNemar's test to compare the predicted accuracies of two models. In McNemar's test, the null hypothesis, which means that none of the two models performs better than the other, is rejected at $p = 0.05$ ($|z| > 1.96$), which indicates the significance level.

Competitors included in our comparison are given in the following.

1) *RF-200:* A random forest being composed of 200 decision trees.
2) *SVM-RBF:* An SVM[1] having the widely used radial basis function (RBF) kernel. We make use of five-fold cross validation to search optimal hyperparameters γ (the spread of the RBF kernel) and C (controlling the magnitude of penalization during the model optimization) in the range of $\gamma = 2^{-3}, 2^{-2}, \ldots, 2^{4}$ and $C = 10^{-2}, 10^{-1}, \ldots, 10^{4}$.
3) *CCF-200:* A canonical correlation forest (CCF)[2] [53] with 200 trees.

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

[2] https://github.com/twgr/ccfs

4) *SICNN:* A CNN model, which makes an attempt at solving the curse of dimensionality by first utilizing a computational intelligence (particle swarm optimization) algorithm to choose informative spectral bands and then training a 2-D CNN using the selected bands. The used network is made up of three convolutional layers. The first two convolutional layers are followed max-pooling layers and their fields of view are 4 × 4 and 5 × 5, respectively. The last convolutional layer is equipped with 4 × 4 filters. Moreover, 32, 64, and 128 convolutional filters are used separately for those three convolutional layers. For more details, refer to [54].
5) *2-D CNN:* The exact architecture of the 2-D CNN is a VGG-like network, in which we utilize three convolutional blocks and 3 × 3 filters for all the blocks. Spatial shrinkage is operated by three max-pooling layers following the convolutional blocks. Each convolutional block in this 2-D CNN has two convolutional layers, and 32, 64, and 128 filters are used for convolutional layers of those three blocks, respectively.
6) *GCN:* To evaluate the superiority of the proposed approach, we perform an ablation study, i.e., using a GCN introduced in [48] which has no the proposed graph learning module but other parts the same as the nonlocal GCN. This GCN uses a fixed graph calculated according to original spectrum signatures using a Gaussian distance.

Note that in order to make our model completely comparable with other investigated approaches, for the first two data sets, we use standard training and test sets. In addition, for the Salinas scene, training samples are generated by simple random sampling. In both hyperspectral data sets, 10% samples of the training set are randomly selected as validation samples. In other words, in the network training phase, we use 90% samples of the training set to iteratively update and optimize network weights and the remaining ones as validation to tune hyperparameters of networks. Regarding training details, Nesterov Adam [55] algorithm is chosen to optimize networks, as it provides much faster convergence compared to other optimizers. Almost all parameters of this optimizer are set as recommended in [55]. We utilize a relatively small learning rate of 2e−04. We set the batch size as 1 and the number of epochs as 800. Since a fully connected graph is calculated, the computational and memory overheads of the proposed method are high. We train it on a NVIDIA DGX-1 server with 4 Tesla V100 GPUs.

C. Embedded Dot Product Versus Embedded Gaussian

Table V compares two instantiations of ϕ and φ, namely the embedded dot product and embedded Gaussian, in our network. As we can see, they perform similarly on the Pavia University, Indian Pines, and Salinas data sets. Compared with the performance of GCN, experimental results show that the instantiation of the nonlocal graph of our model is not the key to the improvement; instead, it is more likely that the nonlocal graph itself is of importance, and it is insensitive to instantiations. In this article, we use the embedded Gaussian to implement our nonlocal GCN.

TABLE V
COMPARISON OF EMBEDDED DOT PRODUCT AND EMBEDDED GAUSSIAN IN THE NONLOCAL GCN IN TERMS OF OA

	Pavia Uni.	Indian Pines	Salinas
Embedded Dot Product	88.22	87.73	92.16
Embedded Gaussian	90.04	87.92	92.48

TABLE VI
RATIOS OF WITHIN-CLASS TO BETWEEN-CLASS SIMILARITY OF ORIGINAL SPECTRA AND OUTPUTS OF GRAPH CONVOLUTIONS ON THE THREE DATA SETS. SMALLER IS BETTER

Data Set	Original Spectrums	Outputs of Graph Convs
Pavia University	3.9088	**1.5675**
Indian Pines	4.8561	**1.8692**
Salinas	0.9498	**0.7928**

D. Analysis of Graph Convolutions

To understand how graph convolutions work, we make use of t-SNE technique to visualize original spectra and outputs of the last graph convolution of our model on the Indian Pines scene in Fig. 4. As shown in this figure, after several graph convolutions, samples of some categories gather together and come into several groups, while in the original spectral domain, these samples may be completely mixed (e.g., classes #1 and #2). It seems that the proposed GCN improves the classification results by minimizing intraclass variance.

In order to quantitatively prove this, we evaluate the model using an index called the ratio of within-class to between-class similarity, which is defined as follows:

$$\boldsymbol{S} = \frac{\mathrm{trace}(\boldsymbol{S}_w)}{\mathrm{trace}(\boldsymbol{S}_b)} \tag{17}$$

where $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ are within-class scatter matrix and between-class scatter matrix, respectively, and defined as

$$\boldsymbol{S}_w = \sum_c \sum_{i \in c} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^{\mathrm{T}} \tag{18}$$

$$\boldsymbol{S}_b = \sum_c N_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^{\mathrm{T}} \tag{19}$$

where $\boldsymbol{\mu}_c = (1/N_c) \sum_{i \in c} \boldsymbol{x}_i$ and N_c denotes the amount of test data belonging to the cth category. Moreover, $\boldsymbol{\mu} = (1/N) \sum_{i=1}^{N} \boldsymbol{x}_i$ is the sample mean of the whole test set.

Table VI reports the calculated ratios of within-class to between-class similarity of original spectra and outputs of graph convolutions on both data sets. We can observe that convolved features in the same category have a higher similarity. Hence, the results demonstrate that this model can minimize intraclass variance.

E. Results and Discussion

Tables VII–IX give the information about per-class accuracies, OAs, AAs, and kappa coefficients obtained by various classification methods on the three data sets. For spectral classification approaches, CCF-200 outperforms RF-200 and SVM-RBF. With respect to the obtained classification results, neural networks, including SICNN, 2-D CNN, GCN, and the proposed nonlocal GCN, show better performance than

TABLE VII

ACCURACY COMPARISONS FOR THE PAVIA UNIVERSITY SCENE. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	SICNN	2D CNN	GCN	Non-local GCN
1	Asphalt	81.54	82.37	86.59	84.21	83.85	78.89	**86.80**
2	Meadows	55.39	67.87	72.33	91.10	**96.09**	90.50	88.74
3	Gravel	53.07	69.18	71.75	64.36	**81.47**	71.70	70.84
4	Trees	98.76	98.37	**99.09**	95.53	96.12	98.76	98.43
5	Metal Sheets	99.11	99.41	99.78	97.70	98.74	**99.93**	99.85
6	Bare Soil	79.10	93.64	**97.26**	56.53	49.79	79.08	94.37
7	Bitumen	84.36	91.20	**91.88**	77.29	79.32	71.20	86.24
8	Bricks	91.39	92.59	94.92	95.57	88.89	92.83	**96.74**
9	Shadows	97.47	96.94	**98.73**	96.20	94.19	97.47	95.78
OA	-	71.53	79.89	83.36	85.25	86.93	87.08	**90.04**
AA	-	82.24	87.95	90.26	84.28	85.38	86.71	**90.87**
Kappa	-	0.6504	0.7491	0.7905	0.8041	0.8242	0.8307	**0.8706**

TABLE VIII

ACCURACY COMPARISONS FOR THE INDIAN PINES SCENE. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	SICNN	2D CNN	GCN	Non-local GCN
1	Alfalfa	56.65	71.39	76.37	79.84	54.77	76.66	**83.09**
2	Corn-notill	55.48	71.05	77.93	92.47	**96.94**	86.10	89.03
3	Corn-min	82.07	86.96	94.57	99.46	99.46	**100**	**100**
4	Corn	85.23	91.72	94.41	93.29	**96.87**	93.06	93.51
5	Grass-pasture	80.20	85.80	91.39	92.68	**94.12**	92.25	**94.12**
6	Grass-trees	94.99	93.85	97.04	96.58	96.81	96.81	**98.18**
7	Grass-pasture-mowed	77.02	75.38	90.96	86.82	**91.29**	88.24	88.24
8	Hay-windrowed	57.94	59.88	69.48	69.52	**93.05**	76.80	78.78
9	Oats	62.94	76.24	**89.01**	83.69	87.59	80.85	86.70
10	Soybean-notill	95.06	96.91	98.77	**100**	**100**	99.38	99.38
11	Soybean-mintill	88.67	79.58	93.73	**96.70**	68.57	93.89	94.94
12	Soybean-clean	53.33	74.84	74.55	96.97	88.48	93.64	**97.27**
13	Wheat	97.78	97.78	**100**	**100**	**100**	**100**	**100**
14	Woods	56.41	79.49	**97.44**	94.87	82.05	92.31	**97.44**
15	Buildings-grass-trees	81.82	**100**	90.91	**100**	**100**	**100**	**100**
16	Stone-steel-towers	**100**	**100**	**100**	**100**	**100**	**100**	**100**
OA	-	69.31	74.24	82.87	85.13	84.76	85.43	**87.92**
AA	-	76.60	83.80	89.78	92.68	90.62	91.87	**93.79**
Kappa	-	0.6538	0.7093	0.8059	0.8313	0.8261	0.8342	**0.8625**

TABLE IX

ACCURACY COMPARISONS FOR THE SALINAS DATA. BOLD NUMBERS INDICATE THE BEST PERFORMANCE

Class No.	Class Name	RF-200	SVM-RBF	CCF-200	2D CNN	GCN	Non-local GCN
1	Brocoli_green_weeds_1	99.29	98.98	99.49	71.57	99.59	**99.69**
2	Brocoli_green_weeds_2	99.21	99.67	**99.95**	99.86	98.07	99.21
3	Fallow	97.72	98.70	99.43	88.89	91.95	**99.79**
4	Fallow_rough_plow	97.62	97.77	**99.33**	98.14	97.84	98.29
5	Fallow_smooth	97.95	98.33	98.82	98.17	98.06	**99.28**
6	Stubble	99.41	99.72	99.80	**100**	99.00	99.80
7	Celery	99.23	99.46	**99.66**	97.00	99.29	99.04
8	Grapes_untrained	61.92	70.37	67.56	70.79	**82.25**	79.11
9	Soil_vinyard_develop	98.70	98.59	99.19	**99.45**	97.11	97.74
10	Corn_senesced_green_weeds	85.56	93.74	93.80	**96.19**	91.60	95.01
11	Lettuce_romaine_4wk	91.75	94.70	95.87	**96.37**	90.77	94.60
12	Lettuce_romaine_5wk	98.24	99.89	99.95	**100**	**100**	**100**
13	Lettuce_romaine_6wk	97.69	97.81	98.15	**100**	98.96	98.96
14	Lettuce_romaine_7wk	92.25	97.35	96.86	98.33	97.35	**99.41**
15	Vinyard_untrained	70.32	71.53	80.77	**91.22**	70.44	84.26
16	Vinyard_vertical_trellis	96.98	**98.18**	**98.18**	93.00	97.10	98.01
OA	-	86.02	88.82	89.72	90.25	90.37	**92.48**
AA	-	92.74	94.67	95.43	93.69	94.34	**96.39**
Kappa	-	0.8450	0.8757	0.8858	0.8918	0.8928	**0.9164**

those traditional machine learning models (i.e., random forest, SVM, and CCF) in regard to OA and kappa coefficient, mainly because they are capable of extracting hierarchical

feature representations and 2) spatial information can be fully exploited in them. These two properties make the networks more robust in finding appropriate decision boundaries and

TABLE X

ASSESSMENTS OF THE SIGNIFICANCE OF CLASSIFICATION ACCURACIES OF THE PROPOSED METHOD COMPARED TO OTHER INVESTIGATED APPROACHES FOR THE THREE DATA SETS

Data Set	RF-200	SVM-RBF	CCF-200	SICNN	2D CNN	GCN
Pavia University	74.659	48.108	34.179	21.957	15.334	21.314
Indian Pines	35.900	28.932	12.855	6.491	7.189	10.498
Salinas	22.464	44.909	42.164	-	47.985	7.178

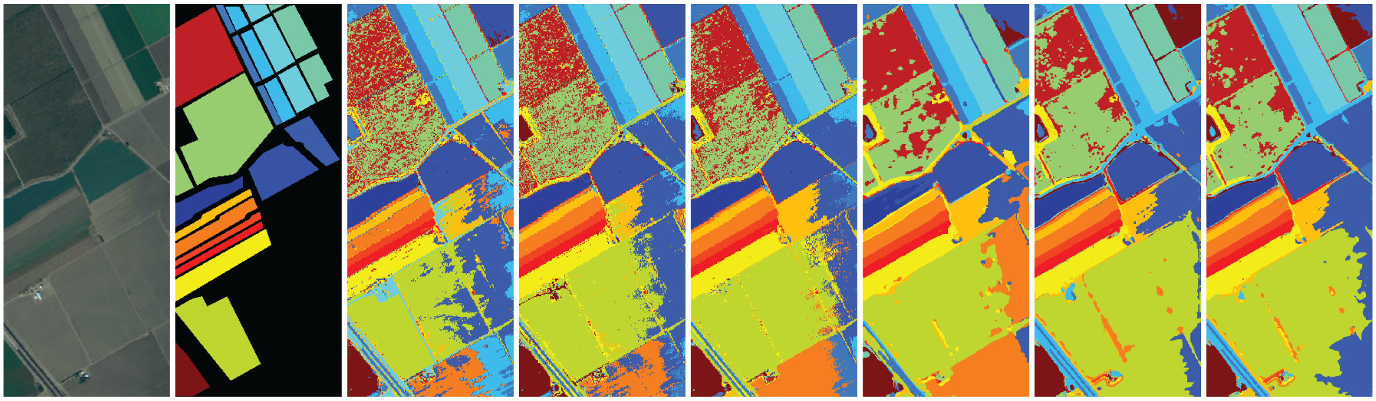


Fig. 7. Classification maps of different approaches for the Salinas data set. (From Left to Right) True-color composite of the hyperspectral image, reference data, RF-200, SVM-RBF, CCF-200, 2-D CNN, GCN, and nonlocal GCN.

enable the models to handle nonlinearly separable data more efficiently.

Among network models, the proposed nonlocal GCN outperforms the 2-D CNN on all indexes on all three data sets. Specifically, our network increases accuracies significantly by 3.11% of OA, 5.49% of AA, and 0.0464 of Kappa coefficient on the Pavia University data set; by 3.16% of OA, 3.17% of AA, and 0.0364 of Kappa coefficient on the Indian Pines data set; and by 2.23% of OA, 2.70% of AA, and 0.0246 of Kappa coefficient on the Salinas scene. This shows the effectiveness of the GCN framework for hyperspectral image classification tasks. On the other hand, in comparison with GCN, the nonlocal GCN is capable of achieving accuracy increments of 2.96%, 4.16%, and 0.0399 for OA, AA, and Kappa coefficient, respectively, on the Pavia University scene. Regarding the Indian Pines scene, the accuracy increments on OA, AA, and Kappa coefficient are separately 2.49%, 1.92%, and 0.0283, respectively. This observation reveals that compared to GCN that uses a fixed graph, our data-driven nonlocal GCN can offer better results.

Table X demonstrates the results of McNemar's test, in which we compute our method and other competitors in terms of the significance of the difference between their classification results. We can see that on both data sets, the improvement of accuracies yielded by our approach is statistically significant compared with other methods. Figs. 5–7 show the classification maps produced by RF-200, SVM-RBF, CCF-200, SICNN, 2-D CNN, GCN, and nonlocal GCN on the three scenes. As shown in these figures, spectral classifiers (i.e., random forest, SVM, and CCF) lead to salt-and-pepper noised classification maps. Although spectral–spatial classification networks (SICNN and 2-D CNN) address this issue, they also result in another problem: oversmoothed classification maps. In contrast, GCN-based methods not only remove noisy scattered points of misclassification from classification maps but also preserve edge information well. Figs. 5–7 show classification results of different methods on the three data sets.

V. CONCLUSION

In this article, a semisupervised nonlocal GCN is proposed for hyperspectral image classification. This network takes as input the whole hyperspectral image instead of its local portions (e.g., pixels and patches), providing a fresh perspective to the classification of hyperspectral imagery using networks. Based on several of our observations from this task, we propose a network module to learn a nonlocal graph representation for the input image. Afterward, a couple of graph convolutional layers are used to extract useful features, depending on the learned graph representation. Both the graph learning module and graph convolutional layers are jointly optimized during training. In addition, a cross-entropy error over all labeled instances is used as the loss function of the nonlocal GCN to achieve the semisupervised classification. Extensive experiments validate the effectiveness of the proposed network. In the future, we will carry out further research to explore dynamic graph-based convolutional networks for hyperspectral image classification.

On the other hand, although we valid that a GCN with a graph learning can provide satisfactory classification results, one shortcoming of this method we can see is high computational and GPU memory overheads. This limits its use in large-scale classification tasks, for example, at the moment we fail to train a model on the Houston data set due to the problem of out of memory. In the future, a promising and important direction is to study how to greatly reduce these overheads. Using superpixels as vertices in the graph is a potential solution, but the oversegmentation algorithm cannot be integrated into an end-to-end network. We believe that lightweight versions are possible.

References

[1] T. C. Bau, S. Sarkar, and G. Healey, "Hyperspectral region classification using a three-dimensional Gabor filterbank," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 9, pp. 3457–3464, Sep. 2010.

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

[3] 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.

[4] Y. Yuan, D. Ma, and Q. Wang, "Hyperspectral anomaly detection via sparse dictionary learning method of capped norm," *IEEE Access*, vol. 7, pp. 16132–16144, 2019.

[5] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2966–2977, Dec. 2016.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1–9.

[7] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–6.

[8] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–15.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[11] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[12] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1–9.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," 2016, *arXiv:1606.00915*. [Online]. Available: http://arxiv.org/abs/1606.00915

[17] X. Lu, Y. Chen, and X. Li, "Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 106–120, Jan. 2018.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 91–99.

[19] B. Zhao, X. Li, and X. Lu, "HSA-RNN: Hierarchical structure-adaptive RNN for video summarization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7405–7414.

[20] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[21] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2183–2195, Apr. 2018.

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

[23] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4959–4962.

[24] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.

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

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

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

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

[30] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral–spatial–temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, Feb. 2019.

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

[32] L. Mou, P. Ghamisi, and X. Xiang Zhu, "Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.

[33] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep&Dense convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 10, no. 9, p. 1454, 2018.

[34] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019, doi: 10.1109/TGRS.2018.2860125.

[35] Z. Xiong, Y. Yuan, and Q. Wang, "AI-NET: Attention inception neural networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 2647–2650.

[36] Y. Yuan, J. Fang, X. Lu, and Y. Feng, "Remote sensing image scene classification using rearranged local features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1779–1792, Mar. 2019.

[37] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

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

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

[40] F. de Morsier, M. Borgeaud, V. Gass, J.-P. Thiran, and D. Tuia, "Kernel low-rank and sparse graph for unsupervised and semi-supervised classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3410–3420, Jun. 2016.

[41] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 60–65.

[42] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[43] L. Mou, Y. Hua, and X. X. Zhu, "A relation-augmented fully convolutional network for semantic segmentation in aerial scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12416–12425.

[44] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–14.

[45] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[46] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[47] 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.

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

[49] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1993–2001.

[50] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 2014–2023.

[51] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–20.

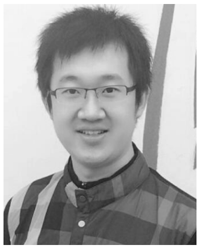
[52] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.

[53] T. Rainforth and F. Wood, "Canonical correlation forests," 2015, *arXiv:1507.05444*. [Online]. Available: http://arxiv.org/abs/1507.05444

[54] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.

[55] T. Dozat. *Incorporating Nesterov Momentum Into Adam*. [Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf

[56] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014.



Lichao Mou (Student Member, IEEE) received the bachelor's degree in automation from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2012, and the master's degree in signal and information processing from the University of Chinese Academy of Sciences (UCAS), China, in 2015. He is pursuing the Ph.D. degree with German Aerospace Center (DLR), Wessling, Germany, and the Technical University of Munich (TUM), Munich, Germany.

In 2015, he was with Computer Vision Group, University of Freiburg, Freiburg, Germany. In 2019, he was a Visiting Researcher with the University of Cambridge, U.K. His research interests include remote sensing, computer vision, and machine learning, especially deep networks and their applications in remote sensing.

Mr. Mou was a recipient of the first place in the 2016 IEEE GRSS Data Fusion Contest and finalists for the Best Student Paper Award at the 2017 Joint Urban Remote Sensing Event and 2019 Joint Urban Remote Sensing Event.

Xiaoqiang Lu (Senior Member, IEEE) is a Full Professor with the Key Laboratory of Spectral Imaging Technology CAS, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

His research interests include pattern recognition, machine learning, hyperspectral image analysis, cellular automata, and medical imaging.

Xuelong Li (Fellow, IEEE) is a Full Professor with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China.

Xiao Xiang Zhu (Senior Member, IEEE) received the master's (M.Sc.) degree, the Doctor of Engineering (Dr.-Ing.) degree, and the "Habilitation" in the field of signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

Since 2019, she has been a Co-Coordinator of the Munich Data Science Research School (www.mu-ds.de), and has also been heading the Helmholtz Artificial Intelligence Cooperation Unit (HAICU) – Research Field "Aeronautics, Space and Transport." Since May 2020, she has been leading one of the three German international future AI labs, named "AI4EO – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond," Munich. She was a Guest Scientist or Visiting Professor at the Italian National Research Council (CNR-IREA), Naples, Italy, Fudan University, Shanghai, China, University of Tokyo, Tokyo, Japan, and the University of California, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. She is currently the Professor for Signal Processing in Earth Observation (www.sipeo.bgu.tum.de) at TUM and the Head of the Department "EO Data Science" at the Remote Sensing Technology Institute, German Aerospace Center (DLR). Her main research interests are remote sensing and earth observation, signal processing, machine learning, and data science, with a special application focus on global urban mapping.

Dr. Zhu is a member of the Young Academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities, the German National Academy of Sciences Leopoldina, and the Bavarian Academy of Sciences and Humanities. She is an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.