An Offset Graph U-Net for Hyperspectral Image Classification

Rong Chen, Gemine Vivone, *Senior Member, IEEE*, Guanghui Li, Chenglong Dai, and Jocelyn Chanussot, *Fellow, IEEE*

***Abstract*— Graph convolutional networks (GCNs) have recently received increasing attention in hyperspectral image (HSI) classification, benefiting from their superiority in conducting shape adaptive convolutions on arbitrary non-Euclidean structure data. However, the performance of GCN heavily depends on the quality of the initial graph. Conventional GCN-based methods only adopt spectral–spatial similarity to build the initial graph without extracting other contextual information from neighboring nodes. In addition, most GCN-based methods use shallow layers, which cannot extract deep discriminative features from HSIs under the limited number of training samples. To solve these issues, we propose a superpixel feature learning via offset graph U-Net for HSI classification, which can learn deep discriminative features from HSIs. Multiple strategies of measuring similarity among superpixels are utilized to build the initial graph, including spectral information, spatial information, and context-aware information among nodes, making the initial graph more accurate. Furthermore, the graph U-Net structure, containing the graph pooling layer and the graph unpooling layer, is helpful in constructing deep GCN layers and learning multiscale features, which can alleviate the oversmoothing problem. Moreover, an offset module is introduced to emphasize the local spectral–spatial information. Finally, we comprehensively evaluate the proposed method on three public datasets. The experimental results demonstrate the superiority of the proposed approach compared with other state-of-the-art methods.**

***Index Terms*— Classification, graph convolutional network (GCN), graph U-Net, hyperspectral imaging, multiresolution analysis, remote sensing, superpixel feature learning.**

I. Introduction

HYPERSPECTRAL images (HSIs) consist of many narrow contiguous bands that carry the spectral information leading to their use for mineral exploration [1], agricultural evaluation [2], and other fields [3]. HSI classification is a hot topic. It aims to assign each pixel to a certain class based on the image contents [4], [5].

Over the past few decades, various traditional pattern recognition methods have been proposed for HSI classification, such as support vector machine (SVM) [6], K-nearest neighbor classifier [7], and multinomial logistic regression (MLR) [8]. Subsequently, sparse representation [9] and collaborative representation [10] have been widely exploited for HSI classification. Despite the success achieved by these methods, they heavily rely on handcrafted configurations. To solve this problem, deep learning-based methods offer an ideal solution to automatically extract high-level discriminative features by fusing the low-level features without handcrafted feature engineering. Convolutional neural networks (CNNs) [11] are the most representative methods and have been extensively investigated thanks to their ability to explore spectral–spatial features from HSIs. Various CNN-based methods have been developed for HSI classification from lightweight CNNs [12] to heavyweight CNNs [13] and from single-channel CNNs [14] to multichannel CNNs [15], generating more and more complex architectures. Furthermore, to make the CNN-based network easier to train, dense connection [16] and residual connection [17], [18] were introduced. Subsequently, to fully use spectral–spatial information, a two-stream CNN approach [19] was employed to perform deep feature fusion and extract local and global spectral–spatial features under the limited training sample size. Furthermore, the attention mechanism [20] has been utilized for HSI classification to emphasize essential features, such as the spectral–spatial attention network [21] and the attention-based adaptive residual network [22]. However, while feature extraction is powerful, some limitations exist in CNN-based methods. On the one hand, CNNs fail to precisely extract the geometric variations of different object regions due to the convolution on regular square areas. On the other hand, CNNs require many training parameters, and they are prone to overfitting when the training samples are scarce.

Besides CNNs, some other high-performance networks have been used for HSI classification. Recurrent neural networks (RNNs) [23] have been proposed to extract spectral information from HSIs, regarding spectral bands as a sequence, thus effectively capturing the correlation among adjacent spectral bands. Subsequently, Hang et al. [24] presented a cascaded RNN to extract the redundant and complementary information of HSIs by employing gated recurrent units (GRUs). Meanwhile, generative adversarial networks

Manuscript received 31 March 2023; revised 26 June 2023; accepted 6 August 2023. Date of publication 22 August 2023; date of current version 1 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62072216 and in part by the China Scholarship Council (CSC) under Grant 202206790097. *(Corresponding author: Guanghui Li.)*

Rong Chen, Guanghui Li, and Chenglong Dai are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China (e-mail: jnuc_r@163.com; ghli@jiangnan.edu.cn; chenglongdai@jiangnan.edu.cn).

Gemine Vivone is with the National Research Council, Institute of Methodologies for Environmental Analysis (CNR-IMAA), 85050 Tito, Italy, and also with the National Biodiversity Future Center (NBFC), 90133 Palermo, Italy (e-mail: gemine.vivone@imaa.cnr.it).

Jocelyn Chanussot is with CNRS, Grenoble Institute of Technology (Grenoble INP), GIPSA-Lab, University of Grenoble Alpes, 38000 Grenoble, France (e-mail: jocelyn.chanussot@grenoble-inp.fr).

Digital Object Identifier 10.1109/TGRS.2023.3307609

(GANs) [25], [26], [27], stacked autoencoders (SAEs) [28], and capsule networks [29] have also been adopted for HSI classification. Moreover, leveraging on transformer [30], transformer-based methods have been applied for HSI classification. The raw transformer in [30] has been exploited for natural language processing (NLP) via adopting a self-attention mechanism. Sun et al. [31] proposed a spectral–spatial feature tokenization transformer (SSFTT) to extract high-level spectral–spatial features from HSI. Furthermore, morphological attention transformer (morphFormer) [32], dual context network with transformer [33], and spectral-swin transformer [34] have been considered for HSI classification.

The recently developed graph convolutional network (GCN) has received increasing attention thanks to its ability to arbitrarily process structured graphs, thus exhaustively extracting spectral–spatial features and preserving the class boundaries [35], [36]. Unlike conventional CNNs that adopt shape-fixed convolutional kernels, GCNs can conduct convolutions on arbitrarily irregular image regions to learn geometric variations of diverse land covers. Hence, several GCN-based methods [37], [38], [39] have been proposed for HSI classification to model spatial topologies of HSIs and learn the relationships among different objects. For instance, a spectral–spatial GCN (S^{2}GCN) [40] was designed to explore spectral–spatial information based on spectral similarity and spatial distance. Mou et al. [41] presented a nonlocal GCN to capture the correlations among any two pixels in HSIs. Nevertheless, treating each pixel as a graph node in these methods may result in a huge amount of calculation, limiting its applicability. To solve this issue, superpixel-based GCN methods [42], [43], [44], which consider each superpixel as a graph node, were proposed to capture large-scale spatial topologies of HSIs reducing the computational burden. Subsequently, to simultaneously extract superpixel-level features and pixel-level features from HSIs, Liu et al. [45] presented a CNN-enhanced GCN (CEGCN), which introduced a GCN (extracting superpixel-level features) and CNN (extracting pixel-level features) into a network framework. Furthermore, GCN methods based on multifeature extraction [46], [47], [48] were proposed to fully use multiresolution features from HSIs. Liu et al. [49] merged neighboring nodes and transformed them into multiresolution graphs to extract features from a hierarchical perspective. Subsequently, Lin et al. [50] presented a context-aware attention graph U-Net (CAGU), which can not only extract intraclass embeddings via attentional graph U-Net but also capture context-aware information by GRU. Besides, to alleviate the oversmoothing problem caused by the deepening of GCN layers, Guo et al. [51] presented a two-stream graph U-Net method to build a deep network. However, the performance of this method is reduced when a limited number of training samples are considered.

Overall, the existing GCN-based methods still suffer from some challenges. First, the performance of GCN-based methods heavily relies upon the quality of the initial graph. Hence, a critical issue is how to describe the similarity of neighboring nodes in a more accurate fashion. Indeed, conventional GCN methods only adopt spectral–spatial similarity to build an initial graph without extracting the context information within and among superpixels. Second, due to the spectral variability and noise present in raw HSI data, deep GCN layers work fine in extracting discriminative features from HSIs. Third, acquiring labeling pixels in HSIs requires experts' knowledge and is quite time-consuming and expensive. Hence, it is important to achieve good performance for HSI classification with a small number of labeled samples.

To tackle the abovementioned issues, a simple but effective superpixel feature learning via offset graph U-Net is proposed for HSI classification to extract deep discriminative spectral–spatial features. Unlike conventional GCN-based methods that only employed spectral–spatial similarity to construct the raw graph, we explore the multiple spectral–spatial information among superpixels to build a graph containing spectral–spatial information and context-aware information. In addition, we propose the use of an offset graph U-Net to extract spectral–spatial features from HSIs for the following two reasons: 1) graph U-Nets, including the graph pooling layer (gPool) and the graph unpooling layer (gUnpool), can extract multiresolution features on the multilevel graph and also increase the depth of the network with limited computing resources and 2) the offset module can emphasize the local spectral–spatial information extracted by GCNs, improving HSI classification performance. More specifically, we transform (through a spectral transformation) the raw HSI to reduce the useless information and segment it into superpixels. Afterward, we conduct feature extraction by an offset graph U-Net after building a multifeature adjacency matrix to extract spectral–spatial features from HSIs. Finally, the superpixel-level features are converted into pixel-level features for pixel-level classification.

In summary, the main contributions of this work can be summarized as follows.

1) We propose a superpixel feature learning via offset graph U-Net for HSI classification, which can learn deep discriminative spectral–spatial features and multiresolution features using graphs working at different scales with a reduced number of training parameters. Besides, the gPool and the gUnpool can alleviate the oversmoothing problem when deepening the depth of the network.
2) We propose to use multiple spectral–spatial information among superpixels to measure the similarity among nodes, thus getting a more accurate version of the initial graph and enhancing the features' representation ability. Furthermore, in this phase, a new CNN-based spectral extraction module is considered to support the graph construction when redundant and noisy information is taken in input (which is often the case for HSIs).
3) To emphasize the local spectral–spatial information, an offset module is proposed for graph U-Nets to further improve the classification accuracy.

The remaining of this article is organized as follows. Section II presents the related works. Section III describes the proposed method. The wide experimental analysis on three HSI datasets is provided to the readers in Section IV. Finally, the concluding remarks are drawn in Section V.

II. Related Works

A. Graph Convolutional Network

GCN [52], a multilayer neural network, is adopted for classification on arbitrary non-Euclidean data, which can directly operate on graphs and generate node embeddings by aggregating features from neighboring nodes. More specifically, given an undirected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively, the adjacency matrix $\mathbf{A}$ of $\mathcal{G}$ (representing the edge weight between each pair of nodes) is calculated as

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \mathrm{N}\big(\mathbf{x}_j\big) \text{ or } \mathbf{x}_j \in \mathrm{N}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the graph nodes and $\mathrm{N}(\mathbf{x}_i)$ is the set of neighbors of $\mathbf{x}_i$.

The convolution operation on graphs can be expressed as a signal $\mathbf{x}$ multiplied by a filter $g_\theta = \mathrm{diag}(\boldsymbol{\theta})$ (where $\mathrm{diag}(\cdot)$ indicates a diagonal matrix) in the Fourier domain, which is defined as

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

where $\star$ is the multiplication operation, $\cdot^\top$ is the transpose operator, $\mathbf{U}$ denotes the eigenvectors of the normalized graph Laplacian denoted by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, $\mathbf{I}$ represents the identity matrix, $\mathbf{D}$ denotes the degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$, $\boldsymbol{\Lambda}$ is the diagonal matrix of its eigenvalues, and $\mathbf{U}^\top\mathbf{x}$ is the graph Fourier transform of $\mathbf{x}$. To reduce the high computational complexity of the eigenvector decomposition, Hammond et al. [53] proposed to approximate $g_\theta(\boldsymbol{\Lambda})$ by the Chebyshev polynomials, $T_k(\mathbf{x})$, up to the Kth order

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

where $\boldsymbol{\theta}'_k$ is a vector of the Chebyshev coefficients and $\widetilde{\boldsymbol{\Lambda}} = (2/\lambda_{\max})\boldsymbol{\Lambda} - \mathbf{I}$, with $\lambda_{\max}$ being the largest eigenvalue of $\mathbf{L}$. The Chebyshev polynomials are recursively defined as $T_k(\mathbf{x}) = 2\mathbf{x}T_{k-1}(\mathbf{x}) - T_{k-2}(\mathbf{x})$ with $T_0(\mathbf{x}) = 1$ and $T_1(\mathbf{x}) = \mathbf{x}$. Thus, we can approximate the convolution of the signal $\mathbf{x}$ by the filter $g_{\theta'}$ as

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

where $\widetilde{\mathbf{L}} = (2/\lambda_{\max})\mathbf{L} - \mathbf{I}$, which can be verified by $(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top)^k = \mathbf{U}\boldsymbol{\Lambda}^k\mathbf{U}^\top$. It is worth noting that $T_k(\cdot)$ is the Kth-order polynomials of the Laplacian. Kipf and Welling [52] approximated $\lambda_{\max}$ to 2. Hence, when $K = 1$, (4) is given as follows:

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

where $\boldsymbol{\theta}'_0$ and $\boldsymbol{\theta}'_1$ are two free parameters. Given $\boldsymbol{\theta} = \boldsymbol{\theta}'_0 = -\boldsymbol{\theta}'_1$, (5) can be simplified as

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

To further reduce the number of parameters, Kipf and Welling [52] set $\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \rightarrow \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}$ with $\widetilde{\mathbf{A}} = \mathbf{A}+\mathbf{I}$ and $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$. Thus, the final graph convolution can be expressed as

$$\mathbf{H}^{(l+1)} = \sigma\Big(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}\Big) \tag{7}$$

where $\mathbf{H}^{(l+1)}$ is the output of the $(l+1)$th layer, $\sigma(\cdot)$ denotes the activation function, and $\mathbf{W}^{(l+1)}$ represents a trainable parameter.

B. Graph U-Nets

Graph U-Nets [54], an encoder–decoder module, introduce the gPool (downsampling) and graph unpooling (upsampling) layers on graph data. In gPool, a new graph is obtained by selectively choosing a subset of nodes from the original graph, resulting in a smaller graph structure. The process of graph pooling for layer l is expressed as

$$\mathbf{y} = \mathbf{X}^l\mathbf{P}^l/\|\mathbf{P}^l\|_1 \tag{8}$$

$$\mathrm{idx} = \mathrm{rank}(\mathbf{y}, k) \tag{9}$$

$$\hat{\mathbf{y}} = \mathrm{sigmoid}(\mathbf{y}(\mathrm{idx})) \tag{10}$$

$$\hat{X}^l = \mathbf{X}^l(\mathrm{idx}, :) \tag{11}$$

$$\mathbf{A}^{l+1} = \mathbf{A}^l(\mathrm{idx}, \mathrm{idx}) \tag{12}$$

$$\mathbf{X}^{l+1} = \hat{\mathbf{X}}^l \odot \big(\hat{\mathbf{y}}\mathbf{1}^\top\big) \tag{13}$$

where $\mathbf{X}$ is the graph node representation, $\mathbf{P}$ denotes a trainable projection vector, $\|\cdot\|_1$ represents the l_1-norm, $\mathrm{rank}(\mathbf{y}, k)$ is the node ranking function, $\mathrm{sigmoid}(\cdot)$ is the sigmoid function, which selects the k-largest values in $\mathbf{y}$, idx is the index of the nodes of the new graph, $\mathbf{A}^{l+1}$ and $\mathbf{X}^{l+1}$ indicate the new adjacency matrix and the new graph representation, respectively, $\odot$ is the elementwise matrix multiplication, and $\mathbf{1}^\top$ is an all-ones vector.

The gPool aims to reduce the number of graph nodes. Instead, in the gUnpool, the number of graph nodes needs to be upsampled to restore them to their original structure. To accomplish this, the positions of the nodes selected in the corresponding gPool are stored, and this information is exploited to reposition the nodes to their original locations within the graph. The process of gUnpool is denoted as

$$\mathbf{X}^{l+1} = \mathrm{distribute}\big(\mathbf{0}, \mathbf{X}^l, \mathrm{idx}\big) \tag{14}$$

where idx is the index of the nodes in the corresponding gPool, $\mathbf{0}$ denotes the (initially empty; i.e., an all-zeros matrix) graph representation for the new graph, and $\mathrm{distribute}(\cdot)$ arranges (based on the indexes specified in idx) the row vectors in $\mathbf{X}^l$ into the graph representation $\mathbf{0}$.

III. Proposed Approach

The structure of the proposed method is shown in Fig. 1, which contains three components: graph construction, feature extraction, and pixelwise classification. In the following, each component will be described in detail.

A. Notation

The mathematical notation used in this article is given as follows. Scalars are denoted by lowercase letters (e.g., x). Matrices (2-D and 3-D arrays) and vectors are expressed in

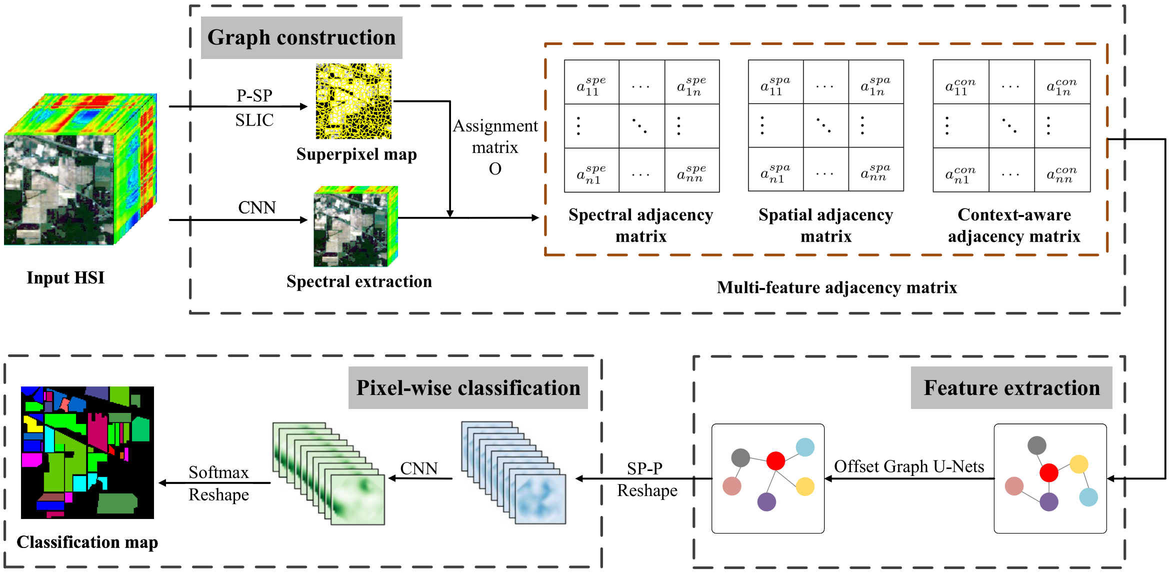

Graph construction
P-SP
SLIC
Superpixel map
Assignment matrix O
CNN
Spectral extraction
Input HSI
a_{11}^{spe} … a_{1n}^{spe} a_{n1}^{spe} … a_{nn}^{spe}
Spectral adjacency matrix
a_{11}^{spa} … a_{1n}^{spa} a_{n1}^{spa} … a_{nn}^{spa}
Spatial adjacency matrix
a_{11}^{con} … a_{1n}^{con} a_{n1}^{con} … a_{nn}^{con}
Context-aware adjacency matrix
Multi-feature adjacency matrix
Pixel-wise classification
Feature extraction
Classification map
Softmax
Reshape
CNN
SP-P
Reshape
Offset Graph U-Nets

Fig. 1. Architecture of the proposed method.

uppercase bold letters (e.g., $\mathbf{X}$) and lowercase bold letters (e.g., $\mathbf{x}$), respectively. Euler script letters are used for sets (e.g., $\mathcal{G}$). We define HSIs as $\tilde{X} \in \mathbb{R}^{H\times W\times B}$ with H, W, and B being the height, the width, and the number of spectral bands, respectively. $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ is the graph representation of $\mathcal{G}$, where M is the number of nodes (superpixels). Finally, $\mathbf{x}_i$ refers to the ith node.

B. Graph Construction

As shown in Fig. 1, the graph construction contains three parts: superpixel map, spectral extraction, and multifeature adjacency matrix. First, we need to convert pixel-level HSI into superpixel-level (P-SP) based on a superpixel map. Meanwhile, spectral extraction is used to reduce the noise and redundant information among spectral bands. Then, we can obtain the initial graph representation by an assignment matrix $\mathbf{O}$. Finally, the multifeature adjacency matrix is constructed to obtain the graph.

1) Superpixel Map: The computational complexity of graph convolution on the original HSI may be unacceptable due to a large number of pixels. To solve this problem, the simple linear iterative clustering (SLIC) [55] is exploited to divide the HSI into several homogeneous superpixels with high spectral–spatial similarity, which creates segmentation by iteratively local clustering using the k-means algorithm. In this way, the long-range relationship among pixels can be more effectively characterized because superpixel dependencies are much longer than those extracted by pixel-level convolutions [56]. Specifically, the HSI $\tilde{X} \in \mathbb{R}^{H\times W\times B}$ is segmented into $M = \lceil (H \times W)/\sigma \rceil$ superpixels, where σ is the parameter used to control the size of the graph. Each superpixel is regarded as a graph node, greatly improving the computational efficiency. Let $\mathbf{O} \in \mathbb{R}^{HW\times M}$ be the assignment matrix among pixels and superpixels obtained by the SLIC. Thus, we have

$$\mathbf{O}_{i,j} = \begin{cases} 1, & \text{if } \tilde{x}_i \in S_j \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $\mathbf{O}_{i,j}$ is the ith pixel belonging to the jth superpixel, $\tilde{x}_i$ denotes the ith pixel in $\bar{X} \in \mathbb{R}^{HW\times B}$, $\bar{X} = \text{Flatten}(\tilde{X})$, Flatten$(\cdot)$ is the flatten operator, and S_j indicates the jth superpixel.

2) Spectral Extraction: HSIs consist of hundreds of spectral bands, thus presenting redundant information and noise. To extract more discriminative spectral features from HSIs, a nonlinear transformation is adopted to each pixel, using a two-layer standard convolution with kernel size 1×1 to process $\tilde{X}$. In 2-D convolution, the output feature of a neuron $f_{i,j}^{xy}$ at position (x, y) in the jth feature map of the ith layer is denoted as

$$f_{ij}^{xy} = \Phi\left(\sum_r w_{ijr} f_{(i-1)r}^{(x+1)(y+1)} + b_{ij}\right) \tag{16}$$

where r is the number of kernels, Φ denotes the activation function, and w and b represent the kernel weight and bias, respectively. After the end of the spectral extraction, the redundant information and the uncorrelated noise are reduced.

Hence, the representation of each node (i.e., superpixel) is the mean of the spectral bands of the pixels in the corresponding superpixel, which can be expressed as

$$\mathbf{X} = \mathbf{O}^{\top}\bar{X} \tag{17}$$

where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ is initial graph representation and M is the number of nodes. $\bar{X} = \text{Flatten}(\tilde{X})$.

3) Multifeature Adjacency Matrix: The quality of the predefined graph strongly influences the performance of GCNs. Thus, it is crucial to accurately draw the relationships among nodes. Furthermore, every superpixel represents a perceptually significant region, and its shape and size can be dynamically adjusted to adapt it to several spatial structures. Conventional GCN methods only exploit the spectral information among graph nodes when constructing the adjacency matrix, ignoring the internal spatial context information. Hence, we propose, in this work, a multifeature adjacency matrix (i.e., the spectral adjacency matrix, the spatial adjacency matrix, and the context-aware adjacency matrix) to effectively extract spectral–spatial information within and among superpixels.

More in detail, the spectral adjacency matrix is computed by

$$\mathbf{A}_{ij}^{\text{spe}} = \begin{cases} \|\mathbf{x}_i - \mathbf{x}_j\|_1^2, & i \neq j \\ 0, & i = j \end{cases} \tag{18}$$

where $\mathbf{x}_i$ is the representation of the ith graph node. In general, one land cover in HSIs is divided into many smaller compact regions during the segmentation process. Hence, the spatially neighboring superpixels tend to be in the same category. Therefore, the spatial information is introduced to measure the similarity of graph nodes. The spatial adjacency matrix is expressed as

$$\mathbf{A}_{ij}^{\text{spa}} = \begin{cases} \|p_i - p_j\|_1^2, & i \neq j \\ 0, & i = j \end{cases} \tag{19}$$

where p_i is the spatial centroid coordinate of $\mathbf{x}_i$.

To exploit more spectral–spatial information among superpixels, we introduce the context-aware adjacency matrix, which describes the contextual relationships among neighboring nodes. A weighted averaging procedure is carried out among the neighboring superpixels, S_{ij} (see Fig. 2), $j = 1, 2, \ldots, J$, of the central superpixel, S_i. Thus, we have

$$\mathbf{A}_{ij}^{\text{con}} = \begin{cases} \|\acute{x}_i - \acute{x}_j\|_1^2, & i \neq j \\ 0, & i = j \end{cases} \tag{20}$$

where $\acute{x}_i$ is the ith weighted average superpixel defined as

$$\acute{x}_i = \left(\sum_{j=1}^{J} \mu_{ij} \mathbf{x}_{ij} \right) \otimes \mathbf{x}_i. \tag{21}$$

$\otimes$ is the elementwise product, and μ_{ij} denotes the weight, which is expressed as [57]

$$\mu_{ij} = \frac{\exp\left(-\|\mathbf{x}_{ij} - \mathbf{x}_i\|_2^2 / h\right)}{\sum_{j=1}^{J} \exp\left(-\|\mathbf{x}_{ij} - \mathbf{x}_i\|_2^2 / h\right)}. \tag{22}$$

J is the number of neighboring superpixels of S_i, and h is empirically set to 15.

Therefore, $\mathbf{A}$ can be calculated as

$$\mathbf{A}_{ij} = \begin{cases} \Psi_{\eta,\varepsilon}\left(\mathbf{A}_{ij}^{c}\right), & \text{if } \mathbf{x}_i \in \mathrm{N}\left(\mathbf{x}_j\right) \text{ or } \mathbf{x}_j \in \mathrm{N}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

where $\mathbf{A}_{ij}^{c} = (\mathbf{A}_{ij}^{\text{spe}}, \mathbf{A}_{ij}^{\text{spa}}, \mathbf{A}_{ij}^{\text{con}})$; $\Psi_{\eta,\varepsilon}(\mathbf{A}_{ij}^{c}) = \exp(-\|\mathbf{A}_{ij}^{\text{spe}} + \eta\mathbf{A}_{ij}^{\text{spa}} + \varepsilon\mathbf{A}_{ij}^{\text{con}}\|^2)$; η and ε are the spatial and the context-aware coefficients, respectively, which are training parameters with initial values equal to 1; and $\mathrm{N}(\mathbf{x}_j)$ and $\mathrm{N}(\mathbf{x}_i)$ denote the set of neighbors of $\mathbf{x}_j$ and $\mathbf{x}_i$, respectively. In this way, three kinds of superpixel-based information (spectral, spatial, and context-aware ones) are considered in the adjacency matrix.

C. Feature Extraction

GCNs are widely utilized for data in non-Euclidean spaces and can effectively preserve class boundaries for HSI classification. However, traditional GCN methods adopted (one or two) shallow layers to extract features from HSIs. Instead, it is still challenging to extract deep discriminative features, especially in complex HSI scenes. Hence, the gPool is introduced to GCNs to reduce graph size and increase the receptive field. In addition, gPools can preserve nodes that contain critical information while discarding irrelevant graph nodes. In this article, an offset module is designed to emphasize the crucial local spectral–spatial information. The illustration of the offset graph U-Net module is shown in Fig. 3. First, we exploit a GCN layer to learn low-dimensional representations. Afterward, the encoder is constructed by stacking several blocks, each consisting of a GCN layer followed by a gPool layer. In the decoder part, the same number of blocks is stacked, each consisting of a GCN layer, a gUnpool layer, and an offset module. Finally, two GCN layers and an offset module are used to further extract discriminative features. The graph convolution operation in the proposed method is expressed as

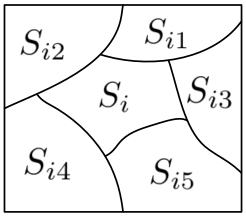


Fig. 2. Instance showing the current superpixel, S_i, and its neighboring superpixels, $S_{i1}, \ldots, S_{i5}$.

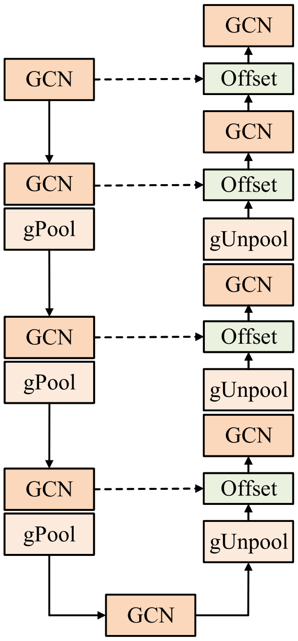


Fig. 3. Illustration of the offset graph U-Net module. The dotted line refers to the skip connection.

$$\mathbf{H}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}\right) \tag{24}$$

where $\mathbf{H}^{(l+1)}$ is the output of the $(l+1)$th layer, $\sigma(\cdot)$ denotes the activation function, and $\mathbf{W}^{(l+1)}$ represents a trainable parameter.

The initial graph embedding is fed into the offset graph U-Net, and after a series of graph convolution and gPools, the graph features are extracted by the GCN in a fine-to-coarse manner. Afterward, these graph features are fed into the subsequent graph convolution, gUnpools, and the offset

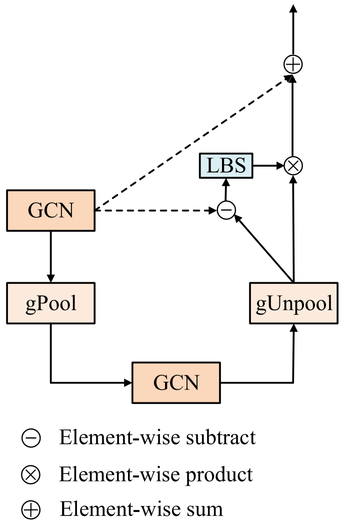


Fig. 4. Illustration of the offset module. The dotted line refers to the skip connection.

TABLE I
STATISTICS OF THE INDIAN PINES DATASET

Class No.	Class Name	Training	Testing
1	Alfalfa	30	16
2	Corn-notill	30	1398
3	Corn-mintill	30	800
4	Corn	30	207
5	Grass-pasture	30	453
6	Grass-trees	30	700
7	Grass-pasture-mowed	15	13
8	Hay-windrowed	30	448
9	Oats	15	5
10	Soybean-notill	30	942
11	Soybean-mintill	30	2425
12	Soybean-clean	30	563
13	Wheat	30	175
14	Woods	30	1235
15	Buildings-grass-trees	30	356
16	Stone-steel-towers	30	63
	Total	450	9799

TABLE II
STATISTICS OF THE PAVIAU DATASET

Class No.	Class Name	Training	Testing
1	Asphalt	30	6601
2	Meadows	30	18619
3	Gravel	30	2069
4	Trees	30	3034
5	Metal Sheets	30	1315
6	Bare Soil	30	4999
7	Bitumen	30	1300
8	Bricks	30	3652
9	Shadows	30	917
	Total	270	42506

module to be fused in a coarse-to-fine manner. This U-Net structure tends to extract multiresolution spectral–spatial information from HSIs. We adopt skip connections between the pooling–unpooling pairs to better fuse different stages features and prevent feature loss during graph unpooling. Besides, the GCN is utilized for gathering information from adjacent nodes. As a result, the crucial information transmitted by the neighboring nodes is represented by the offset features before and after the extraction. The illustration of the offset module is shown in Fig. 4. The offset features are obtained by the elementwise subtraction of the gPool's input features from the gUnpool's output features, where the subtraction operator acts as a filter. Afterward, the offset features are fed into the LBS (i.e., the linear, batch norm, and sigmoid function) block to obtain the weighted features. Then, we multiply the gUnpool's output features and the weighted features adding the outcome to the gPool's input features in an elementwise manner. Hence, the output of the offset module is

$$\mathbf{H}_{\text{off}} = \text{LBS}\big(\mathbf{H}_{gP} - \mathbf{H}_{gU}\big) \otimes \mathbf{H}_{gU} + \mathbf{H}_{gP} \tag{25}$$

where $\mathbf{H}_{gP}$ denotes the input feature of the gPool, $\mathbf{H}_{gU}$ is the output feature of the gUnpool, and $\otimes$ indicates the elementwise product operation. Thus, with the support of the offset module, the proposed GCN can aggregate useful features within neighboring nodes to improve classification performance.

D. Pixelwise Classification

The task of HSI classification aims to assign a specific label to each pixel. After feature extraction, the superpixel-level features, $\mathbf{H}_{\text{final}}$, extracted by the proposed offset graph U-Net, need to be reprojected into pixel level (SP-P) for conducting pixelwise classification. More specifically, the SP-P assignment is accomplished by linearly interpolating pixel features based on the assignment matrix $\mathbf{O}$. Thus, we have that the pixel-level features, $\mathbf{P}$, are defined as

$$\mathbf{P} = \text{Reshape}(\mathbf{O}\mathbf{H}_{\text{final}}) \tag{26}$$

where Reshape$(\cdot)$ refers to the reconstruction of the spatial dimensions of data where the flatten operator has been applied. To further explore the local spectral–spatial information, the pixel-level features, $\mathbf{P}$, are fed into 2-D convolution layers. In the end, a softmax layer is used for predicting per-pixel class probabilities. In this work, the classic cross-entropy error is adopted as HSI classification loss, $\mathcal{L}$, which is defined as

$$\mathcal{L} = -\sum_{i=1}^{T}\sum_{j=1}^{C}\mathbf{Y}_{ij}\log\big(\hat{Y}_{ij}\big) \tag{27}$$

where T and C are the numbers of training samples and classes, respectively, and log is the natural logarithm, while $\mathbf{Y}$ and $\hat{Y}$ denote the label and predicted matrices, respectively.

IV. EXPERIMENTAL ANALYSIS

To evaluate the effectiveness of the proposed method, we conducted extensive experiments on three public HSI datasets, i.e., Indian Pines, the University of Pavia (PaviaU), and Salinas. Four evaluation metrics, i.e., per-class accuracy, overall accuracy (OA), average accuracy (AA), and Kappa coefficient (Kappa), have been used to evaluate the performance of the compared approaches.

A. Datasets

In these experiments, we evaluate the proposed method on three widely used HSIs. The false-color image and the ground-truth map of the three datasets are shown in Figs. 5–7. Besides,

TABLE III
STATISTICS OF THE SALINAS DATASET

Class No.	Class Name	Training	Testing
1	Weeds1	30	1979
2	Weeds2	30	3696
3	Fallow	30	1946
4	Fallow plow	30	1364
5	Fallow smooth	30	2648
6	Stubble	30	3929
7	Celery	30	3549
8	Grapes untrained	30	11241
9	Soil	30	6173
10	Corn	30	3248
11	Lettuce-4wk	30	1038
12	Lettuce-5wk	30	1897
13	Lettuce-6wk	30	886
14	Lettuce-7wk	30	1040
15	Vinyard untrained	30	7238
16	Vinyard trellis	30	1777
	Total	480	53649

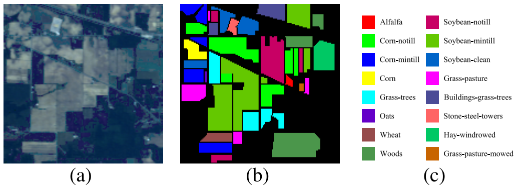


Fig. 5. (a) False-color representation of the Indian Pines dataset (bands 29, 19, and 9). (b) Ground truth for the Indian Pines dataset. (c) Color map.

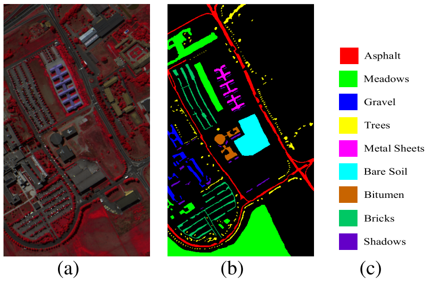


Fig. 6. (a) False-color representation of the PaviaU dataset (bands 80, 60, and 35). (b) Ground truth for the PaviaU dataset. (c) Color map.

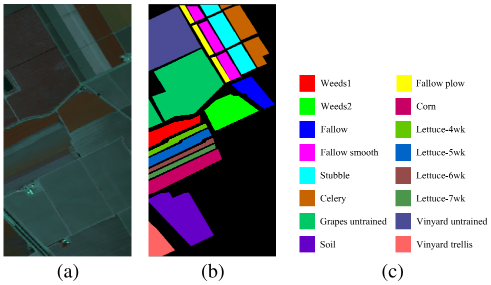


Fig. 7. (a) False-color representation of the Salinas dataset (bands 36, 30, and 35). (b) Ground truth for the Salinas dataset. (c) Color map.

TABLE IV
INFLUENCE OF DIFFERENT NUMBERS OF EPOCHS. BOLD: BEST; UNDERLINE: SECOND BEST

Epoches	Indian Pines	PaviaU	Salinas
100	94.99-1.37	91.75-0.70	99.30-0.48
200	95.51-0.85	94.23-0.67	99.62-0.38
300	<u>95.97-0.42</u>	<u>95.48-1.25</u>	<u>99.63-0.35</u>
400	**96.84-0.55**	**95.66-1.64**	**99.68-0.19**
500	95.50-0.12	93.90-0.20	99.57-0.38

Tables I–III detail the statistics of the three datasets, including the size of training and testing samples.

The first dataset is Indian Pines, gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over Northwest Indian. This dataset contains 16 classes and has 145 × 145 pixels with 20-m/pixel spatial resolution. After removing 24 water absorption bands, there are 200 spectral bands left in Indian Pines covering the spectrum from 0.4 to 2.5 μm.

The second dataset is PaviaU, collected by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor over PaviaU, Italy. This dataset contains nine classes and has 610 × 340 pixels with 1.3-m/pixel spatial resolution. After removing 12 noisy bands, there are 103 spectral bands left in PaviaU covering the spectrum from 0.43 to 0.86 μm.

The third dataset is Salinas, acquired by the AVIRIS sensor over Salinas Valley, USA. This dataset contains 16 classes and has 512 × 217 pixels with 3.7-m/pixel spatial resolution. After removing 20 water absorption bands, there are 204 spectral bands left covering the spectrum from 0.4 to 2.5 μm.

B. Experimental Setup

In the experiments, the proposed method is implemented in Pytorch with an Adam optimizer. The learning rate is $5e-4$. Table IV reports the performance of the proposed method varying the number of epochs. We can see that the proposed method achieves the best accuracy when the number of training epochs is 400. For each dataset, we randomly choose 30 examples (i.e., pixels) per class for training or 15 examples if there are less than 30 examples in the corresponding category. To demonstrate the effectiveness of the proposed method, several state-of-the-art HSI classification methods are used for comparison, including the spatial–spectral CNN-based methods (two-dimensional CNN (2DCNN) [58], 3-D deep learning approach (3DCNN) [59], spectral–spatial residual network (SSRN) [60], and attention-based adaptive spectral–spatial kernel Resnet (A^2S^2K) [22]), the GAN-based method (3DGAN [25]), the transformer-based method (SSTN [61]), and the GCN-based methods (dual graph U-Nets (DGUs) [51], residual GCN (DRGCN) [62], and multilevel superpixel structured graph U-Nets (MSSGUs) [49]). The hyperparameters of the compared approaches are in agreement with the ones used in the original papers. All these methods are run ten times to obtain stable and reliable results on each HSI dataset, and the AA and the related standard deviation are reported. All the experiments are executed on Intel core i7-10700, 2.9 GHz, 16-GB RAM, and GTX 1660Ti GPU.

C. Experimental Results

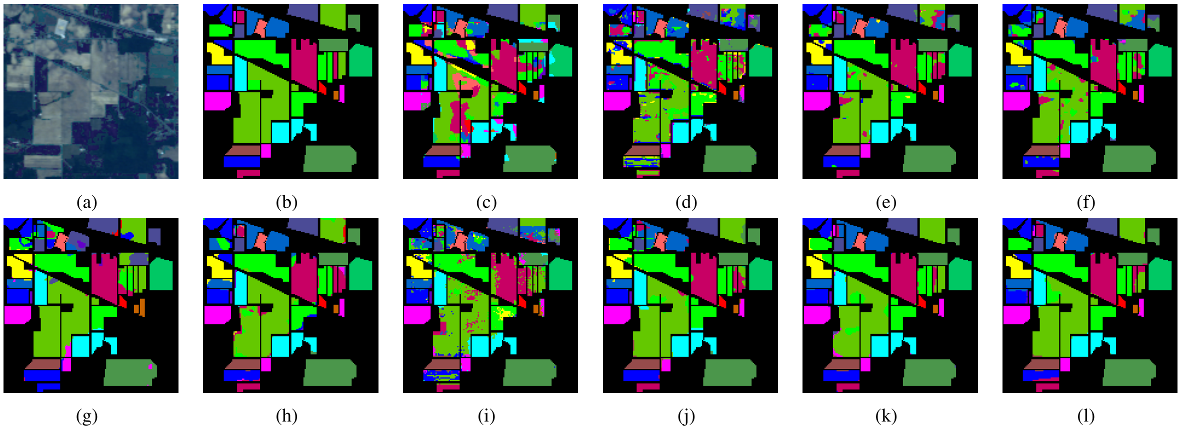
To evaluate the effectiveness of the proposed method, the classification results of the compared methods on the Indian Pines, PaviaU, and Salinas datasets are shown in Tables V–VII. The best and second-best scores are highlighted in bold and underlined, respectively. Besides, Figs. 8–10 show the classification maps on the three datasets for all the compared approaches.

1) Results on the Indian Pines Dataset: For the Indian Pines dataset, the classification results obtained by the various approaches are reported in Table V. It can be observed that

TABLE V

CLASSIFICATION PERFORMANCE ON THE INDIAN PINES DATASET. BOLD: BEST; UNDERLINE: SECOND BEST

Class No.	2DCNN [58]	3DCNN [59]	SSRN [60]	A^{2}S^{2}K [22]	3DGAN [25]	SSTN [61]	DGU [51]	DRGCN [62]	MSSGU [49]	Proposed
1	99.11±1.53	96.87±5.03	**100.00±0.00**	80.11±8.63	97.32±2.35	98.93±2.83	**100.00±0.00**	97.51±3.06	**100.00±0.00**	**100.00±0.00**
2	54.39±4.92	70.90±4.85	89.55±3.68	87.64±3.87	78.46±4.09	84.56±3.93	68.06±8.61	88.26±5.23	92.19±4.26	**94.84±2.63**
3	24.95±4.23	65.70±6.09	83.11±4.06	85.96±5.46	85.47±3.25	85.16±5.44	55.16±5.41	90.55±1.71	**97.50±3.21**	96.12±3.14
4	**37.53±4.76**	**84.73±4.10**	**99.71±0.44**	85.06±7.34	83.95±5.04	98.89±1.70	88.74±3.71	95.45±5.24	99.75±0.32	**99.95±0.28**
5	41.98±4.35	86.35±8.02	89.27±2.59	92.08±3.18	85.39±5.58	91.05±2.69	76.21±8.55	93.37±1.38	95.95±1.94	**97.77±2.36**
6	76.98±7.18	92.68±4.88	**99.57±0.62**	95.46±1.87	88.75±1.47	97.16±1.45	97.99±1.02	95.42±1.81	98.12±0.86	98.17±0.73
7	**100.00±0.00**	96.92±3.76	**100.00±0.00**	51.58±10.63	62.47±11.69	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
8	48.52±5.16	96.80±1.99	96.87±9.37	99.54±0.39	95.02±1.36	99.47±0.23	94.91±0.13	99.73±0.21	**100.00±0.00**	**100.00±0.00**
9	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	35.64±6.09	80.06±6.28	99.43±1.38	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
10	59.19±8.15	78.57±5.26	90.34±3.67	66.35±6.84	80.19±4.92	88.41±3.91	68.89±4.12	88.95±5.24	**95.08±4.43**	94.02±2.76
11	47.59±6.97	64.39±5.27	79.90±4.81	**97.84±1.46**	88.21±2.35	74.82±5.45	62.34±6.72	88.46±5.35	91.36±2.95	93.02±1.80
12	46.14±6.10	77.94±5.59	94.72±2.50	84.46±7.93	75.47±6.85	95.53±2.28	69.51±6.05	93.64±2.44	**96.98±1.49**	95.91±4.39
13	99.48±1.03	99.54±0.66	99.77±0.52	84.83±3.97	85.54±5.36	99.65±0.43	99.48±1.03	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
14	61.40±4.29	86.38±12.11	98.06±1.64	98.67±0.64	92.17±2.81	97.42±1.56	93.56±2.63	97.82±3.79	**99.12±0.36**	98.76±2.85
15	18.37±4.51	73.51±8.23	75.11±4.42	88.09±5.98	85.25±4.86	98.36±2.61	56.57±4.29	98.87±1.26	98.85±2.50	**99.85±0.51**
16	98.73±1.55	98.73±1.18	99.52±0.72	84.96±4.61	73.42±6.59	97.98±3.12	**100.00±0.00**	**100.00±0.00**	99.84±0.46	**100.00±0.0**
OA	51.53±1.54	76.58±2.60	89.08±1.48	87.26±1.43	85.93±2.24	92.43±1.67	73.62±2.14	92.26±1.35	95.85±0.79	**96.84±0.55**
AA	63.45±1.05	85.62±1.33	93.46±0.88	80.04±1.37	83.62±1.48	93.05±1.16	83.23±1.27	95.55±0.64	97.79±0.37	**98.31±0.29**
Kappa	46.64±1.58	73.59±2.79	87.62±1.66	86.02±1.97	84.09±2.37	91.93±1.39	70.29±2.34	91.11±1.52	95.26±0.90	**96.38±0.62**
Training time	103.02	1370.36	830.13	7540.02	684.54	927.61	1422.99	21.06	45.25	**13.89**
Testing time	0.53	83.62	54.15	12.04	8.23	3.07	255.67	1.31	1.03	**0.38**
Running time (s)	103.55	1453.98	884.28	7541.06	692.77	930.68	1678.66	22.37	46.28	**14.27**
Parameters (K)	185.2	46.11	364.19	370.79	10460.3	**20.98**	142.42	87.82	133.05	143.82


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

(g) (h) (i) (j) (k) (l)

Fig. 8. Classification maps of the different methods on the Indian Pines dataset. (a) False-color image (bands 29, 19, and 9). (b) Ground truth. (c) 2DCNN (OA = 51.53%). (d) 3DCNN (OA = 76.58%). (e) SSRN (OA = 89.08%). (f) A^{2}S^{2}K (OA = 87.26%). (g) 3DGAN (OA = 85.93%). (h) SSTN (OA = 92.43%). (i) DGU (OA = 73.62%). (j) DRGCN (OA = 92.26%). (k) MSSGU (OA = 95.85%). (l) Proposed (OA = 96.84%).

TABLE VI

CLASSIFICATION PERFORMANCE ON THE PAVIAU DATASET. BOLD: BEST; UNDERLINE: SECOND BEST

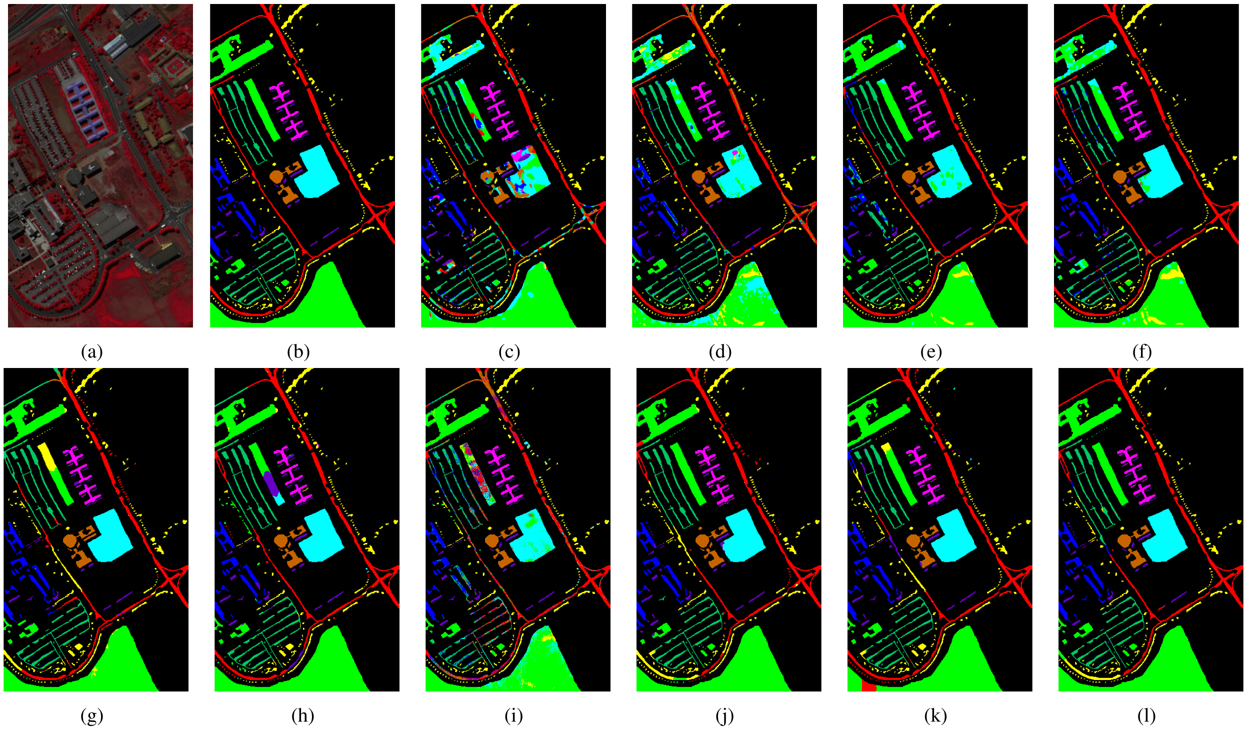
Class No.	2DCNN [58]	3DCNN [59]	SSRN [60]	A^{2}S^{2}K [22]	3DGAN [25]	SSTN [61]	DGU [51]	DRGCN [62]	MSSGU [49]	Proposed
1	57.01±4.31	79.42±6.58	90.07±3.68	94.25±3.03	78.64±1.56	93.09±1.79	61.73±5.49	84.21±3.49	**97.94±2.37**	91.15±2.15
2	51.67±5.15	69.36±6.11	87.42±0.31	84.72±6.76	93.00±1.72	89.62±3.37	71.81±3.04	93.16±4.88	85.62±3.41	**94.64±3.89**
3	48.41±5.82	84.73±4.92	70.79±6.78	89.34±7.38	90.18±3.94	86.33±5.34	75.33±3.78	97.14±0.95	**100.00±0.00**	98.67±2.89
4	77.52±3.11	92.81±1.49	96.53±0.59	80.06±5.37	**98.62±1.49**	93.24±1.53	87.71±1.62	79.21±4.01	88.79±4.25	89.78±3.18
5	96.56±5.71	99.61±0.42	97.26±0.16	99.64±0.14	99.08±0.17	99.34±0.12	99.07±0.13	99.61±0.31	99.75±0.09	**99.94±0.09**
6	46.85±7.95	78.12±4.84	92.47±0.64	86.41±8.63	96.66±1.59	98.23±1.34	88.94±3.33	99.34±1.09	99.06±1.24	**99.69±0.29**
7	75.24±6.69	89.48±5.88	99.51±0.33	86.79±7.84	95.36±2.25	99.15±0.23	98.19±0.93	99.21±0.89	**100.00±0.00**	**100.00±0.00**
8	83.97±4.22	85.76±3.38	94.86±4.15	81.24±4.83	78.41±3.97	93.23±4.12	59.43±5.01	93.35±3.57	**99.13±0.85**	97.59±1.70
9	86.22±5.81	**99.45±0.49**	96.34±1.81	94.48±5.96	98.38±0.24	97.81±1.29	96.66±1.18	85.34±4.33	98.96±1.16	95.13±2.41
OA	59.26±3.29	77.99±6.05	89.42±1.67	85.31±1.64	90.52±2.67	93.42±2.43	74.71±1.69	91.93±1.12	93.49±1.63	**95.66±1.64**
AA	69.27±2.91	86.53±2.14	91.19±0.74	92.63±2.01	91.05±1.15	94.31±0.69	82.18±0.87	92.28±0.71	94.65±0.71	**95.98±0.61**
Kappa	51.23±3.44	72.55±6.46	86.29±0.98	81.49±1.97	88.59±2.09	90.78±1.91	68.41±1.97	89.46±1.63	90.08±1.96	**94.27±2.07**
Training time	**68.21**	353.32	1545.99	2824.25	821.63	2583.39	1160.87	873.89	2162.43	575.26
Testing time	3.92	277.94	41.04	14.72	10.45	10.42	654.98	11.66	10.25	**0.95**
Running time (s)	**72.13**	631.26	1587.03	2838.97	832.08	2593.81	1815.85	885.55	2172.68	576.21
Parameters (K)	184.3	35.53	216.56	220.75	10460.2	**16.21**	100.06	87.39	119.53	84.78

TABLE VII

CLASSIFICATION PERFORMANCE ON THE SALINAS DATASET. BOLD: BEST; UNDERLINE: SECOND BEST

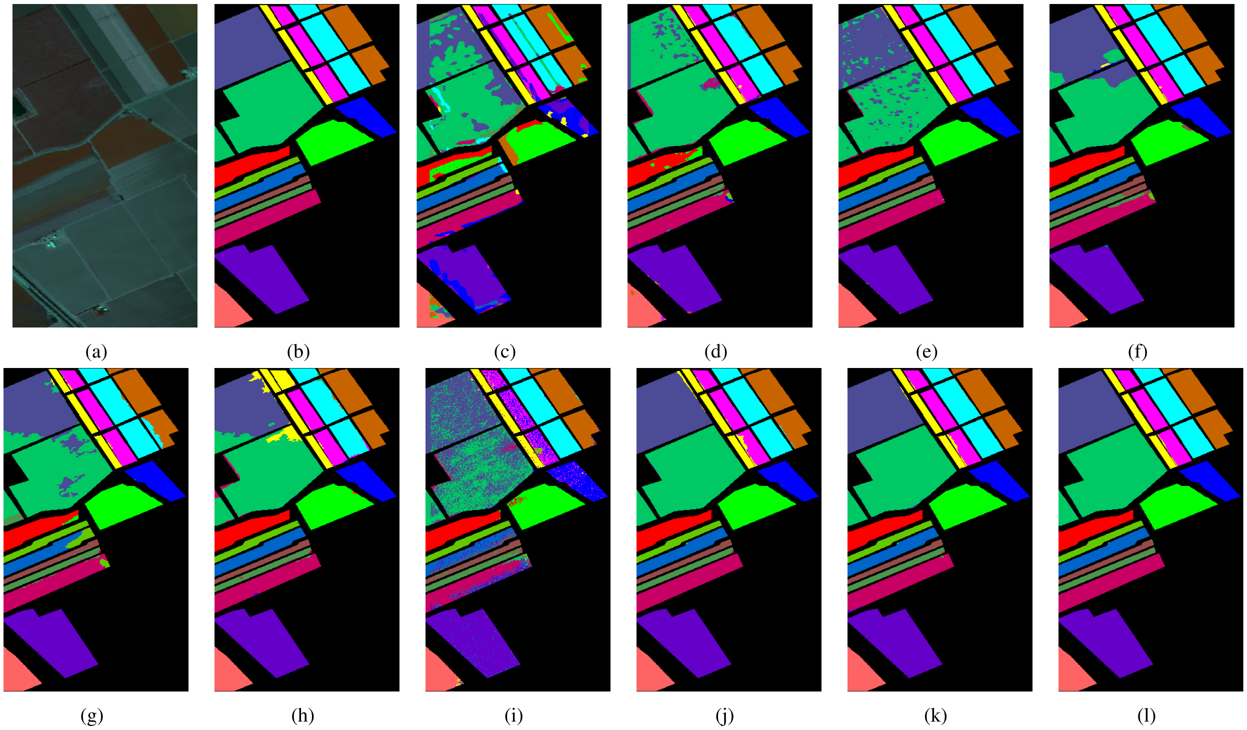
Class No.	2DCNN [58]	3DCNN [59]	SSRN [60]	A^{2}S^{2}K [22]	3DGAN [25]	SSTN [61]	DGU [51]	DRGCN [62]	MSSGU [49]	Proposed
1	56.76±11.64	93.15±4.64	99.46±0.11	99.84±0.32	95.61±0.83	99.23±5.48	92.16±1.67	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
2	71.62±8.52	97.98±3.41	99.61±0.56	99.56±0.14	99.54±0.29	99.50±0.12	98.74±1.29	**100.00±0.00**	99.98±0.03	**100.00±0.00**
3	68.83±6.25	97.78±2.71	98.43±1.63	99.73±0.09	99.06±0.25	96.73±0.37	87.14±0.81	**100.00±0.00**	**100.00±0.00**	**100.00±0.00**
4	91.71±1.23	99.06±0.51	86.50±1.74	99.58±0.54	91.76±0.34	99.06±1.41	90.38±3.14	99.58±0.31	99.84±0.14	**99.91±0.26**
5	74.36±5.94	96.51±1.84	93.32±6.01	99.13±0.39	98.03±0.83	96.43±2.19	80.91±8.23	98.02±1.26	**99.47±0.18**	98.86±0.75
6	82.46±4.22	98.44±0.86	99.03±0.35	99.48±0.06	96.91±0.15	99.37±0.72	96.64±1.22	99.97±0.03	**100.00±0.00**	**100.00±0.00**
7	78.38±7.74	98.43±0.51	92.83±5.67	99.87±0.07	98.95±1.35	99.28±0.44	97.31±0.47	99.89±0.04	99.87±0.26	**99.97±0.08**
8	62.01±9.19	70.67±12.86	83.06±10.51	89.12±2.43	79.47±2.68	80.59±5.07	63.31±15.82	99.46±0.62	97.49±0.83	**99.91±0.14**
9	81.09±7.59	98.68±0.56	99.36±0.24	99.57±0.06	99.83±0.06	99.42±0.47	89.33±9.19	**100.00±0.00**	**100.00±0.00**	99.71±0.86
10	80.31±2.99	86.69±3.34	94.59±0.47	98.63±0.74	97.95±0.61	94.55±1.39	79.93±7.51	97.37±3.06	**99.01±0.16**	98.71±1.69
11	86.31±1.36	94.57±1.32	99.06±0.11	**100.00±0.00**	99.86±0.21	99.16±0.92	94.63±1.57	**100.00±0.00**	99.97±0.05	**100.00±0.00**
12	81.36±3.52	98.06±0.02	98.63±0.34	99.87±0.38	99.35±1.39	99.27±0.63	69.19±15.45	99.83±0.18	99.81±0.18	**99.94±0.12**
13	87.68±1.22	97.85±0.73	93.88±4.31	99.03±0.43	98.83±0.92	99.54±0.57	80.37±7.16	98.38±2.01	99.06±0.34	**100.00±0.00**
14	86.08±2.53	94.33±1.49	94.82±3.16	99.45±0.16	99.57±0.73	97.73±0.74	80.77±6.72	98.01±1.67	99.43±0.38	**99.77±0.21**
15	49.14±10.85	70.01±33.94	99.03±0.32	86.21±3.64	88.78±3.49	82.49±3.92	53.88±0.23	**99.73±0.21**	99.28±0.09	99.14±1.34
16	18.39±5.21	84.57±3.68	95.47±3.69	99.52±0.61	96.16±1.37	97.04±3.17	74.23±3.88	99.51±0.74	98.96±0.24	**99.89±0.21**
OA	67.24±2.23	82.53±4.55	94.09±1.37	95.17±0.68	93.62±0.74	94.88±0.67	78.48±2.21	99.48±0.32	99.45±0.16	**99.68±0.19**
AA	71.09±1.54	90.92±1.53	94.98±1.43	98.03±0.19	97.43±0.19	96.09±1.47	83.06±1.83	99.36±0.29	99.49±0.11	**99.73±0.13**
Kappa	64.06±2.37	80.71±4.89	93.43±1.02	94.57±0.83	92.98±0.73	93.91±1.39	76.18±2.39	99.43±0.36	99.35±0.24	**99.64±0.21**
Training time	121.78	1923.24	5531.75	10707.6	726.68	2852.58	817.47	425.13	1222.40	**119.40**
Testing time	2.09	638.79	27.83	151.93	10.95	20.53	519.34	6.22	5.95	**0.13**
Running time (s)	123.87	2562.03	5559.58	10859.53	737.66	2873.11	1336.81	431.35	1228.35	**119.53**
Parameters (K)	185.2	47.89	401.06	408.24	10460.3	**21.37**	142.42	87.82	136.17	53.03

the proposed method achieves the best and most stable (i.e., with the smallest standard deviation) performance among the competitors concerning OA, AA, and Kappa, respectively. Furthermore, it is worth noting that the proposed method


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

(g) (h) (i) (j) (k) (l)

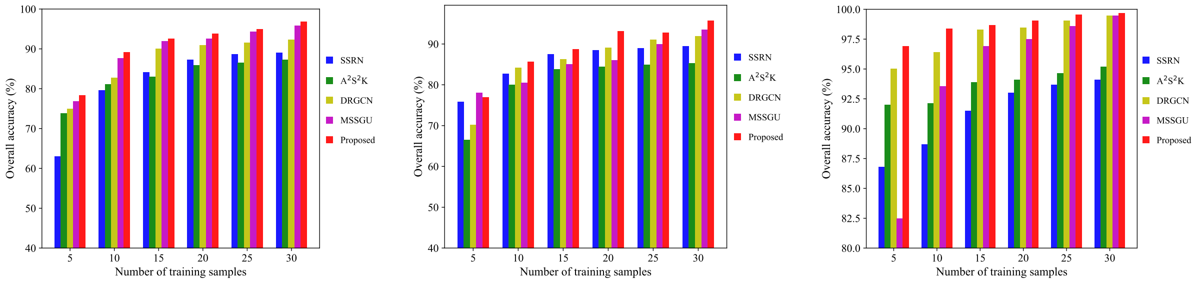
Fig. 9. Classification maps of the different methods on the PaviaU dataset. (a) False-color image (bands 80, 60, and 35). (b) Ground truth. (c) 2DCNN (OA = 59.26%). (d) 3DCNN (OA = 77.99%). (e) SSRN (OA = 89.42%). (f) A^2S^2K (OA = 85.31%). (g) 3DGAN (OA = 90.52%). (h) SSTN (OA = 93.42%). (i) DGU (OA = 74.71%). (j) DRGCN (OA = 91.93%). (k) MSSGU (OA = 93.49%). (l) Proposed (OA = 95.66%).


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

(g) (h) (i) (j) (k) (l)

Fig. 10. Classification maps of the different methods on the Salinas dataset. (a) False-color image (bands 36, 30, and 35). (b) Ground truth. (c) 2DCNN (OA = 67.24%). (d) 3DCNN (OA = 82.53%). (e) SSRN (OA = 94.09%). (f) A^2S^2K (OA = 95.17%). (g) 3DGAN (OA = 93.62%). (h) SSTN (OA = 94.88%). (i) DGU (OA = 78.48%). (j) DRGCN (OA = 99.48%). (k) MSSGU (OA = 99.45%). (l) Proposed (OA = 99.68%).

ranks first and second for ten and six classes, respectively. Moreover, the proposed method is the fastest approach among the competitors, and the number of parameters is limited. This demonstrates the proposed method's effectiveness, efficiency, and stability compared to the state-of-the-art approaches used for comparison. About the CNN-based methods, SSRN and A^2S^2K achieve better performance, benefiting from adopting the dense and residual connections. The improvement of the proposed method over both the GCN-based methods (i.e., DRCCN and MSSGU) is higher than 4.58% and 0.99% on OA, respectively. Furthermore, since the proposed method incorporates the multiple superpixel information to build a



(a) (b) (c)

Fig. 11. OAs obtained by some representative methods varying the numbers of training samples per class. (a) Indian Pines. (b) PaviaU. (c) Salinas.

TABLE VIII

Performance (OA) of the Proposed Method on the Indian Pines Dataset Under Different Network Depths and Hidden Kernel Sizes for the GCN Layers. Bold: Best; Underline: Second Best

Network depth Hidden kernel size	1	2	3	4	5
20	93.44±2.28	93.51±1.09	93.81±1.45	92.85±2.61	90.83±4.25
30	94.68±1.29	94.96±0.91	93.94±1.74	93.76±2.47	94.31±1.99
40	95.11±0.91	93.51±2.06	94.73±1.41	94.79±1.15	93.98±2.59
50	94.36±1.97	95.13±0.74	**96.84±0.55**	95.45±0.84	95.54±0.91
60	94.13±2.41	94.49±1.96	95.38±0.99	94.78±1.98	93.55±2.87

TABLE IX

Performance (OA) of the Proposed Method on the PaviaU Dataset Under Different Network Depths and Hidden Kernel Sizes for the GCN Layers. Bold: Best; Underline: Second Best

Network depth Hidden kernel size	1	2	3	4	5
20	92.30±3.44	90.13±3.99	91.71±3.66	91.77±3.03	90.11±3.92
30	93.48±1.87	93.31±2.10	91.98±3.54	93.22±1.87	93.21±1.52
40	93.34±2.32	**95.66±1.64**	90.96±9.05	92.78±2.87	93.60±1.73
50	93.63±2.32	92.05±4.17	94.37±1.93	93.80±1.85	92.27±3.57
60	94.13±2.53	94.16±2.31	94.88±1.06	93.77±2.62	93.41±1.73

TABLE X

Performance (OA) of the Proposed Method on the Salinas Dataset Under Different Network Depths and Hidden Kernel Sizes for the GCN Layers. Bold: Best; Underline: Second Best

Network depth Hidden kernel size	1	2	3	4	5
20	99.34-0.51	99.50-0.38	99.41-0.50	99.21-0.69	99.33-0.50
30	99.59-0.33	**99.68-0.19**	99.58-0.23	99.59-0.33	99.54-0.35
40	99.23-0.13	99.64-0.34	99.60-0.27	99.58-0.38	99.52-0.34
50	99.31-0.65	99.67-0.21	99.47-0.45	98.14-4.41	99.29-0.79
60	99.50-0.41	99.53-0.39	99.65-0.20	99.69-0.20	99.34-0.34

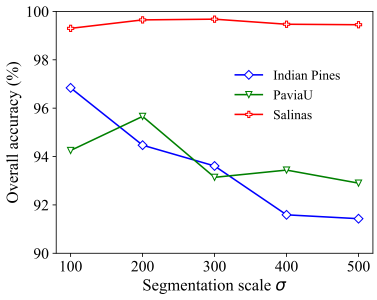


Fig. 12. Performance of the proposed method with different segmentation scales on the three datasets.

more accurate graph, it performs better than MSSGU on some land cover categories, such as "Grass-pasture (ID = 5)" and "Stone-steel-towers (ID = 16)."

Fig. 8 depicts the classification maps obtained by the compared methods on the Indian Pines dataset, where the false-color image and the ground-truth map are shown in Fig. 8(a) and (b), respectively. We can observe that the classification maps acquired by the CNN-based methods (i.e., 2DCNN and 3DCNN) are quite confusing. By contrast, the classification map of the proposed method is smoother, even showing fewer misclassifications compared with the other techniques.

2) Results on the PaviaU Dataset: Table VI reports the classification results of the compared methods on the PaviaU dataset. The proposed approach achieves the optimal classification performance in terms of OA, AA, and Kappa, respectively, which again demonstrates the strength of the proposed method in extracting features from HSIs. Specifically, compared with the CNN-based methods, the proposed method outperforms SSRN and A^{2}S^{2}K by 6.24% and 10.35% on OA, respectively, which indicates that the spectral features and spatial correlations extracted by our method are more effective. Besides, our method outperforms the second-best approach (MSSGU) on OA, AA, and Kappa by 2.17%, 1.33%, and 4.19%, respectively. Regarding the computational complexity, the running time and the number of parameters of the proposed method are very low compared with GCN-based methods and several CNN-based methods (e.g., SSRN and A^{2}S^{2}K), thus implying that the proposed method is efficient.

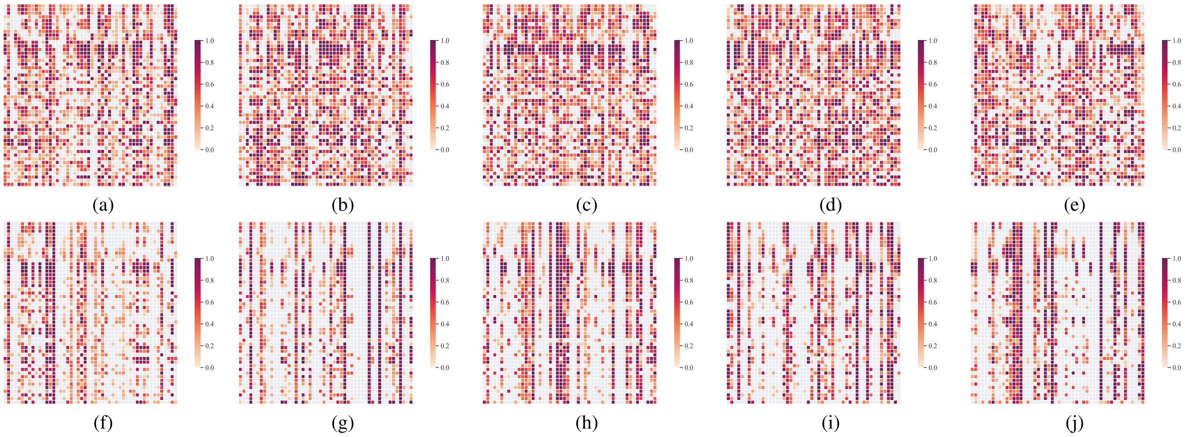


Fig. 13. Visualization of the five layers features (first 50 nodes) extracted by the proposed method with/without graph pooling–unpooling module on the Indian Pines dataset. (a)–(e) From first to fifth layers with graph pooling–unpooling module. (f)–(j) From first to fifth layers without graph pooling–unpooling module.

Fig. 9 depicts the classification maps generated by the compared methods on the PaviaU dataset. The false-color image and the ground-truth map are also presented. We can observe that the classification map of the proposed approach is the closest to the ground truth, showing fewer misclassifications and less salt-and-pepper noise. This outcome is in agreement with the classification results in Table VI. The reason is that the proposed method can effectively exploit the deep discriminative features to recognize different land covers. On the contrary, the classification maps produced by the rest of the methods show more misclassifications. For instance, due to the use of fixed convolutional kernels, the classification map of A^{2}S^{2}K yields more errors than the proposed method.

3) Results on the Salinas Dataset: Table VII reports the classification results of the compared methods on the Salinas dataset. It can be found that almost all the approaches achieve better performance on the Salinas dataset than on the Indian Pines and PaviaU datasets. This may be due to the fact that the boundaries of the Salinas dataset are more regular, and large-scale land covers are easier to be distinguished. Similar to the classification results of the previous two datasets (Indian Pines and PaviaU), the proposed method gets the best classification results. Besides, it is worth to be remarked that the classification performance of DGU is poor on the three datasets. The reason could be that DGU builds the initial graph among pixels in the image patch and cannot effectively extract the global spectral–spatial features. Furthermore, in the case of similar spectral signatures as for “Grapes untrained” (class = 8) and “Vineyard untrained” (class = 15), the proposed method can distinguish them well. In contrast, CNN-based methods cannot do a good job because of their low ability to model large-scale spatial topology. Moreover, it is worth to be noted that the running time of the proposed method is the shortest. Although it ranks second for the number of parameters, the difference with respect to the first position is very limited (just 5.14k parameters more than 3DCNN).

The classification map produced by the compared methods is presented in Fig. 10. Again, the classification map of the proposed method is less noisy, and we can observe fewer misclassifications with respect to the competitors, thus supporting the superiority of the proposed approach.

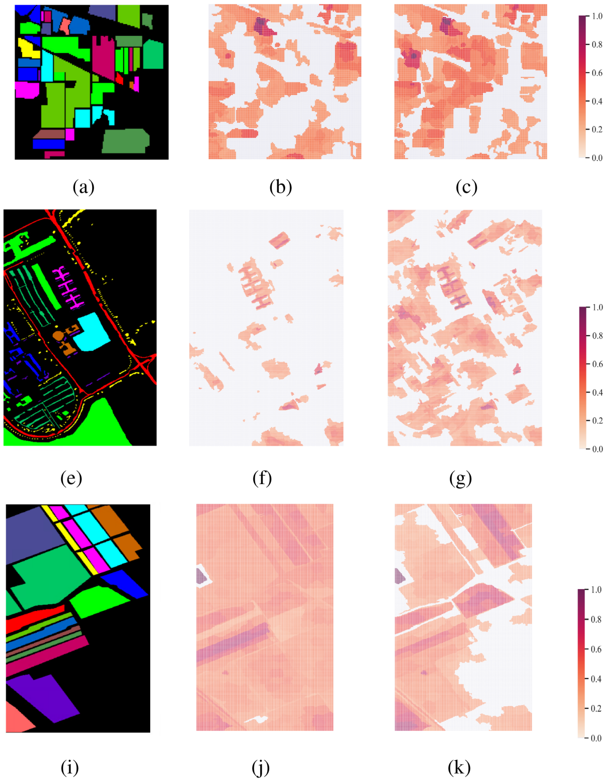


Fig. 14. Visualization of the proposed method without/with offset module on the three datasets. (a), (e), and (i), (b), (f), and (j), and (c), (g), and (k) Ground truth, without offset module, and with offset module, respectively.

TABLE XI
EFFECTIVENESS OF THE GRAPH POOLING–UNPOOLING MODULE ON THE INDIAN PINES DATASET

Network depth	w/o pool-unpool	proposed method
1	52.97±7.20	94.36±1.97
2	39.33±7.87	95.13±0.74
3	36.23±10.42	96.84±0.55
4	33.78±9.73	95.45±0.84
5	32.33±12.01	95.54±0.91

D. Impact of the Number of the Training Samples

The classification results under a small sample size are essential in practical applications for HSI classification. Therefore, we study the feature learning capabilities of the proposed method under this condition. Hence, we selected five, ten,

TABLE XII

EFFECTIVENESS OF THE DIFFERENT MODULES OF THE PROPOSED APPROACH ON THE THREE DATASETS. BOLD: BEST; UNDERLINE: SECOND BEST

Dataset	Methods	OA	AA	Kappa
Indian Pines	w/o-Spectral extraction	95.49±0.63	97.56±0.43	94.84±0.72
	w/o-Multi-feature adjacency matrix	93.51±1.48	95.84±1.07	92.73±1.59
	w/o-Offest module	93.70±1.50	96.56±1.36	92.81±1.71
	Proposed	**96.84±0.55**	**98.31±0.29**	**96.38±0.62**
PaviaU	w/o-Spectral extraction	90.07±3.36	91.60±1.92	87.12±4.16
	w/o-Multi-feature adjacency matrix	91.95±3.97	92.45±3.05	87.96±4.09
	w/o-Offest module	92.10±3.64	93.46±3.18	89.71±4.62
	Proposed	**95.66±1.64**	**95.98±0.61**	**94.27±2.07**
Salinas	w/o-Spectral extraction	99.61±0.16	99.62±0.18	99.57±0.18
	w/o-Multi-feature adjacency matrix	99.42±0.36	99.45±0.26	99.50±0.19
	w/o-Offest module	99.58±0.31	99.63±0.28	99.53±0.34
	Proposed	**99.68±0.19**	**99.73±0.13**	**99.64±0.21**

15, 20, 25, and 30 training samples per category. The OAs obtained by some representative methods on the three datasets are reported in Fig. 11. We can observe that the proposed method generally gets higher classification results than the other competitors. Besides, when we have just five samples per class, the proposed method gets slightly lower performance than MSSGU on the PaviaU dataset. The reason may be that our approach incorporates multiple superpixel information to build deep GCNs, which leads to overfitting when dealing with several irregular boundaries in the PaviaU dataset. In addition, the classification performance of the CNN-based methods (i.e., 2DCNN and 3DCNN) dramatically drops when the sample size decreases. Finally, the classification accuracies of DRGCN and MSSGU slightly decrease when the sample size is reduced, thus demonstrating the powerful feature extraction ability of our GCN-based method.

E. Impact of the Size of the Segmentation Scale

The segmentation scale, σ, can control the size of the initial graph and can significantly impact the classification performance. Thus, we studied its impact on our approach. More specifically, we set σ from 100 to 500 with a step size equal to 100. The performance of the proposed method with different segmentation scales on the three datasets is shown in Fig. 12. We can observe fewer fluctuations of the classification accuracy on the Salinas dataset with respect to the Indian Pines and PaviaU datasets when the segmentation scale increases. The reason is related to the fact that the Salinas dataset contains many large-scale land cover regions and regular boundaries, and thus, it is insensitive to the segmentation scale. In contrast, since the Indian Pines dataset consists of many smaller land cover regions, lower values of the segmentation scale parameter, σ, tend to properly map pixels in different categories to different superpixels. Hence, the accuracy of the proposed method decreases with the increase in σ. The PaviaU dataset is also sensitive to changes in the segmentation scale in a similar way as the Indian Pines dataset. In this article, the segmentation scale, σ, is set to 100 for Indian Pines and 200 for PaviaU and Salinas.

F. Impact of the Network Depths and Hidden Kernel Sizes

The network depth and the hidden kernel size of GCN layers in the proposed method are two important hyperparameters. Deepening the network or increasing the number of hidden kernels may not improve classification performance due to the overfitting problem. Therefore, an appropriate network depth with a suitable kernel size can enhance stability and performance. Hence, we set the network depth (the number of blocks containing gPools) to 1, 2, 3, 4, and 5, respectively. The hidden kernel size of GCN layers is set to 20, 30, 40, 50, and 60, respectively. We utilize the grid search strategy to find the optimal parameter. The OAs of the proposed method under different hyperparameters on the Indian Pines, PaviaU, and Salinas datasets are reported in Tables VIII–X, respectively. We can observe that both the network depth and the hidden kernel size affect classification performance. Besides, the best performance is obtained with a relatively small network depth. Finally, by fixing the network depth, we can observe an improvement in the performance when increasing the hidden kernel size until reaching a maximum. After that, the decrease in performance is clear. As a result, for the Indian Pines dataset, the network depth and the hidden kernel size of the GCN layers are set to 3 and 50, respectively. For the PaviaU dataset, the network depth and the hidden kernel size of the GCN layers are set to 2 and 40, respectively. For the Salinas dataset, the network depth and the hidden kernel size of the GCN layers are set to 2 and 30, respectively.

G. Impact of the Graph Pooling–Unpooling Module

Oversmoothing is a common problem for GCN-based methods when increasing the depth of GCNs. Graph pooling–unpooling modules can alleviate this problem. As shown in Fig. 13, we visualize the five layers features (first 50 nodes) extracted by the proposed method with and without the graph pooling–unpooling module on the Indian Pines dataset. We can observe from the first row that the proposed method with the graph pooling–unpooling module can extract discriminative features when deepening the depth of the network, and no smoothing occurred. On the contrary, the second row without the graph pooling–unpooling module easily suffers from the oversmoothing problem even with fewer network layers. Besides, Table XI reports the performance of the proposed method with and without the graph pooling–unpooling module; "w/o pool-unpool" stands for without the graph pooling–unpooling module. It can be observed that "w/o pool-unpool" achieves poor performance on the Indian Pines dataset, and the classification accuracy dramatically drops with the increase in the depth of GCN layers.

H. Impact of the Offset Module

In this section, we study the impact of the offset module. More specifically, we visualize the output weights of the offset module on the three datasets, as shown in Fig. 14. We can see from the third column that the proposed method with the offset module can clearly capture the discriminative spatial–spectral information, which can emphasize the specific regions of HSIs, enabling the network to make more informed classification decisions. The regions extracted by the offset module are closer to the ground truth (first column). Comparatively, the second column without the offset module is not sensitive to spatial–spectral features, especially in the PaviaU dataset where just few features have been captured. Moreover, for the Salinas dataset, the proposed method without the offset module emphasizes all of the spectral–spatial features, not favoring any network decision-making.

I. Ablation Study

The proposed method consists of three main components, i.e., the spectral extraction, the multifeature adjacency matrix, and the offset module. We conduct here an ablation study to explore the contribution of these three main components on the three datasets; "w/o-Spectral extraction," "w/o-multifeature adjacency matrix," and "w/o-Offset module" will refer to the reduced models via removing the spectral extraction, the multifeature adjacency matrix (spatial and context-aware information), and the offset module, respectively. Table XII shows the overall accuracies of the abovementioned methods on the three datasets, where the best and second-best values are highlighted in bold and underlined, respectively. We can see that the proposed method gets the best accuracy on the three datasets, and the measured classification performance obtains lower values when any part is removed. This analysis indicates that each part makes an important contribution to improving HSI classification performance. Besides, the "w/o-multifeature adjacency matrix" achieves the lowest accuracy (considering OA, AA, and Kappa) on the three datasets. Hence, the use of spatial and contextual information is of crucial importance to build a more accurate graph.

V. Conclusion

In this article, a superpixel feature learning via offset graph U-Net is proposed for HSI classification. Instead of just exploiting the spectral–spatial similarity to build the initial graph, we consider multiple spectral–spatial information among superpixels to build a more accurate graph, such as spectral–spatial information and context-aware information among neighboring nodes. In addition, an offset module is proposed to pay more attention to the local spectral–spatial information extracted by GCN layers. Thus, the proposed method not only effectively extracts the deep discriminative features but also extracts multiresolution features from HSIs thanks to the use of the graph U-Net structure in the proposed framework. The experimental results on three widely used HSI datasets demonstrate that the proposed approach outperforms other state-of-the-art methods in both accuracy and efficiency. Future developments go toward the exploration of different U-Net-based architectures, including state-of-the-art mechanisms, such as attention.

References

[1] H. Shirmard, E. Farahbakhsh, R. D. Müller, and R. Chandra, "A review of machine learning in processing remote sensing data for mineral exploration," *Remote Sens. Environ.*, vol. 268, Jan. 2022, Art. no. 112750.

[2] G. Tagliabue et al., "Hybrid retrieval of crop traits from multi-temporal PRISMA hyperspectral imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 187, pp. 362–377, May 2022.

[3] G. Lu and B. Fei, "Medical hyperspectral imaging: A review," *J. Biomed. Opt.*, vol. 19, no. 1, Jan. 2014, Art. no. 010901.

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

[5] C. Cheng, H. Li, J. Peng, W. Cui, and L. Zhang, "Deep high-order tensor convolutional sparse coding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5520611.

[6] 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.

[7] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[8] D. Böhning, "Multinomial logistic regression algorithm," *Ann. Inst. Stat. Math.*, vol. 44, no. 1, pp. 197–200, Oct. 1992.

[9] 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.

[10] W. Li, Q. Du, F. Zhang, and W. Hu, "Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 389–393, Feb. 2015.

[11] J. Gu et al., "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[12] 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.

[13] 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, Jan. 2017.

[14] 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.

[15] C. Chen, J.-J. Zhang, C.-H. Zheng, Q. Yan, and L.-N. Xun, "Classification of hyperspectral data using a multi-channel convolutional neural network," in *Proc. 14th Int. Conf., Intell. Comput. Methodologies (ICIC)*. Wuhan, China: Springer, Jul. 2018, pp. 81–92.

[16] 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. 2261–2269.

[17] 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. 770–778.

[18] L. Dang, P. Pang, and J. Lee, "Depth-wise separable convolution neural network with residual connection for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 20, p. 3408, Oct. 2020.

[19] X. Li, M. Ding, and A. Pižurica, "Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2615–2629, Apr. 2020.

[20] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.

[21] X. Mei et al., "Spectral–spatial attention networks for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 8, p. 963, Apr. 2019.

[22] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.

[23] 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.

[24] 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.

[25] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

[26] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.

[27] J. Feng, N. Zhao, R. Shang, X. Zhang, and L. Jiao, "Self-supervised divide-and-conquer generative adversarial network for classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5536517.

[28] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.

[29] M. E. Paoletti et al., "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[30] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[31] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214.

[32] S. K. Roy, A. Deria, C. Shah, J. M. Haut, Q. Du, and A. Plaza, "Spectral–spatial morphological attention transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5503615.

[33] D. Wang, J. Zhang, B. Du, L. Zhang, and D. Tao, "DCN-T: Dual context network with transformer for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 32, pp. 2536–2551, 2023.

[34] Y. Peng, J. Ren, J. Wang, and M. Shi, "Spectral-Swin transformer with spatial feature extraction enhancement for hyperspectral image classification," *Remote Sens.*, vol. 15, no. 10, p. 2696, May 2023.

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

[36] H. Zhang, J. Zou, and L. Zhang, "EMS-GCN: An end-to-end mixhop superpixel-based graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526116.

[37] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[38] L. Yu, J. Peng, N. Chen, W. Sun, and Q. Du, "Two-branch deeper graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5506514.

[39] Z. Xue, Z. Liu, and M. Zhang, "DSR-GCN: Differentiated-scale restricted graph convolutional network for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5504918.

[40] 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.

[41] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8246–8257, Dec. 2020.

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

[43] H. Hu, M. Yao, F. He, and F. Zhang, "Graph neural network via edge convolution for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[44] J. Bai et al., "Hyperspectral image classification based on superpixel feature subdivision and adaptive graph structure," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5524415.

[45] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "CNN-enhanced graph convolutional network with pixel- and superpixel-level feature fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, Oct. 2021.

[46] Y. Li, B. Xi, J. Li, R. Song, Y. Xiao, and J. Chanussot, "SGML: A symmetric graph metric learning framework for efficient hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 609–622, 2022.

[47] J. Jiang, J. Ma, and X. Liu, "Multilayer spectral–spatial graphs for label noisy robust hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 839–852, Feb. 2022.

[48] C. Zhao, B. Qin, S. Feng, and W. Zhu, "Multiple superpixel graphs learning based on adaptive multiscale segmentation for hyperspectral image classification," *Remote Sens.*, vol. 14, no. 3, p. 681, Jan. 2022.

[49] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "Multilevel superpixel structured graph U-Nets for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5516115.

[50] M. Lin, W. Jing, D. Di, G. Chen, and H. Song, "Context-aware attentional graph U-Net for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[51] F. Guo, Z. Li, Z. Xin, X. Zhu, L. Wang, and J. Zhang, "Dual graph U-Nets for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 8160–8170, 2021.

[52] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017, pp. 1–14.

[53] 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.

[54] H. Gao and S. Ji, "Graph U-Nets," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2083–2092.

[55] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[56] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

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

[58] 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.

[59] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

[60] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[61] Z. Zhong, Y. Li, L. Ma, J. Li, and W.-S. Zheng, "Spectral–spatial transformer network for hyperspectral image classification: A factorized architecture search framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5514715.

[62] R. Chen, G. Li, and C. Dai, "DRGCN: Dual residual graph convolutional network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

Rong Chen received the master's degree in computer science and technology from the Shandong University of Technology, Zibo, China, in 2020. She is currently pursuing the Ph.D. degree with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.

Her research interests are processing, remote sensing, and deep learning.

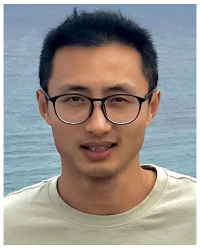
Gemine Vivone (Senior Member, IEEE) received the B.Sc. (summa cum laude), M.Sc. (summa cum laude), and Ph.D. (Hons.) degrees in information engineering from the University of Salerno, Salerno, Italy, in 2008, 2011, and 2014, respectively.

In 2019, he was an Assistant Professor with the University of Salerno. In 2014, he joined the North Atlantic Treaty Organization (NATO) Science and Technology Organization (STO), Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, as a Scientist. In 2013 and 2019, he was a Visiting Scholar and a Professor with the Grenoble Institute of Technology (INPG), Grenoble, France, respectively. He is currently a Senior Researcher with the National Research Council, Rome, Italy. His main research interests focus on statistical signal processing, classification of remotely sensed images, data fusion, and tracking algorithms.

Dr. Vivone is a member of the IEEE Task Force on Deep Vision in Space. He is an Editorial Board Member of *Scientific Reports* (Nature) and *Remote Sensing* (MDPI). He received the IEEE GRSS Early Career Award in 2021, the Symposium Best Paper Award at the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) in 2015, and the Best Reviewer Award of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS) in 2017. Moreover, he is listed in the World's Top 2% Scientists by Stanford University. He is the Co-Chair of the IEEE GRSS Image Analysis and Data Fusion Technical Committee. He was the Leader of the Image and Signal Processing Working Group of the IEEE Image Analysis and Data Fusion Technical Committee from 2020 to 2021. He has served as a guest associate editor for several special issues. He is an Associate Editor of IEEE TGRS and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL).

Guanghui Li received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He has published over 90 papers in journals or conferences. His research interests include the Internet of Things, edge computing, fault-tolerant computing, and nondestructive testing and evaluation.



Chenglong Dai received the M.S. and Ph.D. degrees in computer science and technology from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014 and 2020, respectively.

He is currently a Lecturer with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He has published several related papers in prestigious journals and top conferences, including IEEE TKDE, IEEE TCYB, ACM TKDD, ACM TIST, and SIAM SDM (awarded the Best Paper Award in Data Science Track). His research interests include brain–computer interfaces (BCIs), EEG data mining, and machine learning.

Dr. Dai has served as a Recognition Reviewer for *Knowledge-Based Systems* and IJCNN from 2018 to 2022.

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 currently a Professor of signal and image processing. He has been a Visiting Scholar with Stanford University, Stanford, CA, USA; the KTH Royal Institute of Technology, Stockholm, Sweden; and the National University of Singapore (NUS), Singapore. Since 2013, he has been an Adjunct Professor with the University of Iceland, Reykjavík, Iceland. From 2015 to 2017, he was a Visiting Professor with the University of California at Los Angeles (UCLA), Los Angeles, CA, USA. He currently holds the AXA Chair in remote sensing and is an Adjunct Professor with 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 a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society from 2006 to 2008. He was a member of the Institut Universitaire de France from 2012 to 2017 and has been a Highly Cited Researcher (Clarivate Analytics/Thomson Reuters) since 2018. He 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 received multiple outstanding paper awards. From 2017 to 2019, he was the Vice-President of the IEEE Geoscience and Remote Sensing Society, in charge of meetings and symposia. 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 and the Co-Chair of the GRS Data Fusion Technical Committee from 2009 to 2011 and 2005 to 2008, respectively. He was the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing in 2009. 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 has served as a Guest Editor for the *IEEE Signal Processing Magazine*. 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.