DFL-LC: Deep Feature Learning With Label Consistencies for Hyperspectral Image Classification

Siyuan Liu, Yun Cao, Yuebin Wang, *Member, IEEE*, Junhuan Peng,
P. Takis Mathiopoulos, *Senior Member, IEEE*, and Yong Li

***Abstract*—Deep learning approaches have recently been widely applied to the classification of hyperspectral images (HSIs) and achieve good capability. Deep learning can effectively extract features from HSI data compared with other traditional hand-crafted methods. Most deep learning methods extract image features through traditional convolution, which has demonstrated impressive ability in HSI classification. However, traditional convolution can only operate convolutions with fixed size and weight on regular square image regions. Moreover, it refers to the spectral features of the adjacent pixels but ignores the spectral features of long-range data with the training sample. Although a graph convolution network (GCN) can process irregular image regions, the pixels' relationships for graph construction cannot be well ensured with limited iterations. Hence, the extracted features have limited performance with the GCN. Aiming to extract more representative and discriminative image features, in this article, the deep feature learning with label consistencies (DFL-LC) method is developed to realize HSI classification. In the proposed method, a multiscale convolutional neural network is adopted to obtain basic HSI features, and the GCN can further capture relationships between pixels and extract more representative HSI features. For obtaining discriminative features, we add the label consistency of single pixels and label consistency of group pixels regularization in the objective function. It can maintain label consistency for the general and long-range data and alleviate deficiently labeled samples. The experimental results on three representative datasets fully demonstrate that the DFL-LC method is superior to other methods in both quantitative and qualitative aspects.**

Manuscript received December 1, 2020; revised December 24, 2020; accepted March 1, 2021. Date of publication March 3, 2021; date of current version April 12, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 41801241, in part by the Fundamental Research Funds for the Central Universities under Grant 292018029 and Grant 375201906, in part by the Key Research and Development Projects of Shanxi Province under Grant 201903D121142, in part by the Open Fund of the State Key Laboratory of Remote Sensing Science under Grant OFSLRSS201923, and in part by the Guizhou Science and Technology Plan Project under Grant Qiankehezhicheng [2020] 4Y022. *(Siyuan Liu and Yun Cao contributed equally to this work.) (Corresponding author: Yuebin Wang.)*

Siyuan Liu, Yun Cao, and Junhuan Peng are with the School of Land Science and Technology, China University of Geosciences, Beijing 100083, China (e-mail: l_s_y1995@163.com; cy12160019@163.com; pengjunhuan@163.com).

Yuebin Wang is with the School of Land Science and Technology, China University of Geosciences, Beijing 100083, China, and also with the State Key Laboratory of Remote Sensing Science, Beijing Normal University, Beijing 100875, China (e-mail: xxgcdxwyb@163.com).

P. Takis Mathiopoulos is with the Department of Informatics and Telecommunications, National and Kapodestrian University of Athens, 15784 Athens, Greece (e-mail: mathio@di.uoa.gr).

Yong Li is with the School of Electrical Engineering, Guangxi University, Nanning 530004, China (e-mail: yongli@gxu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3063679

***Index Terms*—Graph convolutional network, hyperspectral image (HSI), image classification, label consistencies, multiscale convolution.**

I. Introduction

THERE are several hundred channels in hyperspectral images (HSIs) that contain high-resolution spectral information of land covers. Each pixel in HSIs corresponds to the spectral reflectance of a particular wavelength, so it can be considered as a high-dimensional vector. Many spectral signatures have largely been used in HSI classification of land covers. In the past few decades, HSI classification has been developed into a significant part of remote sensing. In general, it is challenging for traditional machine learning to realize precise classification for the complex characteristics of HSIs. In addition, the inherent nonlinear relationship between the corresponding class and the spectral information is processed by HSI recognition [1]. As a powerful tool for extracting features, deep learning is widely adopted in several image processing tasks, which can effectively solve nonlinear problems. Therefore, deep learning has also been used for HSI classification and has shown good performance.

In the initial stage of HSI classification research, there were a number of methods focusing on detecting the role of the spectral characteristics of HSIs. Thus, numerous classification methods have been proposed in HSI classification, such as support vector machine (SVM) [2], multinomial logistic regression [3], and dynamic subspace detection [4]. Although most HSI classifications based on spectral and spatial information have obtained excellent performance, they are heavily dependent upon hand-crafted features. Moreover, traditional hand-crafted methods are limited. On the one hand, hand-crafted features are regarded as shallow features, so images can change considerably where the imaging environment is sharp [5]. On the other hand, most hand-crafted features rely on expert knowledge, limited by human factors. Moreover, crafting hand-crafted labels usually requires considerable time, limiting the applicability of those methods in different scenarios [1].

In recent years, deep learning has become a development trend in HSI classification and has achieved good performance. Deep learning methods can effectively exploit features from HSI data compared with other traditional hand-crafted methods. The process of deep learning is automatic, which makes it more suitable to deal with various situations. Because different networks can extract different feature types, deep learning is considered to be a significant feature extraction approach in HSI classification. Thus, we can divide the deep learning networks to classify HSI into spectral feature networks (e.g., DBN [6], 1-D convolutional neural network (CNN) [7], 1-D GAN [8], [9], and RNN [10]), spatial-feature networks (e.g., 2-D CNN [11], FCN-8 [12], and spectral–spatial-feature-based classification (SSFC) [13]), and spectral–spatial-feature networks (e.g., SAE [14], 3-D CNN [15], and 3-D GAN [8]).

In deep learning approaches, CNNs have become a powerful tool in HSI classification methods, which can effectively extract spatial and spectral features. It has achieved impressive performance to classify HSI. Chen *et al.* [11] applied PCA to reduce the dimension of HSI first and then used the 2-D CNN to extract the spatial features within the pixel neighborhood. The above method combines PCA and CNN, which extracts spatial features and greatly reduces the computational cost. Liang and Li [16] proposed a sparse representation method to improve the feature representation ability and the classification accuracy. Deep spatial features extracted by the CNN are encoded into low-dimensional sparse features. The SSFC framework [13] is proposed to classify HSI, in which balanced local discriminant embedding and CNN are used to extract spectral and spatial features, respectively. However, the traditional CNN can only perform convolutions with fixed size and weight on regular square image regions. It only refers to the spectral features of the adjacent pixels but ignores the spectral features of long-range data with the sample. For example, some pixels are usually in the same class in different positions in HSI. These pixels should have similar features. Therefore, their classification performances need to be further improved.

Moreover, with the rapid development of graph theories, graph convolutional networks (GCNs) have been widely used in various applications, such as text classification [17]–[20] and semantic segmentation [21]–[24]. In addition, the GCN has made great progress in image classification [25]–[27]. The GCN can process irregular image regions. The learned hidden layers in the GCN can encode both features of node and local graph structure. Therefore, the GCN can flexibly retain class boundaries while adequately exploiting image features. However, it is not appropriate when the GCN is directly applied in HSI classification. The GCN can assemble and transform features from a defined graph containing the neighbor information of each graph node. In the GCN, the neighborhood structure of the graph adaptively governs the graph convolution operation. Although the GCN can capture relationships based on the predefined graph that contains global information, pixels' relationships for graph construction cannot be well ensured. The main reason lies in that accurate image features cannot be obtained only with the limited iterations of the deep learning framework. Thus, this deep learning framework still cannot ensure the quality of HSI features with the predefined graph.

Inspired by the above discussions, aiming to learn more representative and discriminative image features of HSI, label consistency (LC) is embedded into the deep learning framework in this article. LC includes not only the label consistency of single pixels (LCSP) but also the label consistency of group pixels (LCGP). LCSP means that LC is maintained by calculating the error between the predicted label and the real label of the sample. Moreover, LCGP refers to considering long-range data by introducing a group label matrix to solve problems such as different ground objects in the same spectrum. LCGP can also realize label reuse on the basis of limited labeled data so that the model has better learning ability.

In this article, deep feature learning with label consistencies (DFL-LC) is proposed, which considers both LCSP and LCGP, and its framework is show in Fig. 1. In this approach, we adopt the multiscale convolutional neural network (MSCNN) to extract basic HSI features. The features obtained from the MSCNN are further fed into the GCN, which considers pixels' relationships by constructing an adjacency matrix. The output layer of the GCN is activated by the ReLU function. In order to enhance the performance of HSI classification, LC should also be applied to the deep learning framework. LC includes not only LCSP but also LCGP. With LC, the cross-entropy loss is used to calculate the difference between the outputs and the real labels to keep LCSP. Moreover, to keep LC for the long-range data and alleviate deficiently labeled samples, LCGP regularization is added in the objective function. Finally, an iterative optimization algorithm is used to optimize the objective function.

The main contributions of this article are summarized as follows.

1) DFL-LC is developed to extract HSI features and ensure LC, whose structure contains the MSCNN and the GCN. The LC constraint is embedded in the objective function, and end-to-end optimization is implemented.

2) In DFL-LC, we formulate two kinds of constraints to boost the classification accuracy: LCSP and LCGP constraints. LCSP ensures LC between the outputs and the real labels of the sample. LCGP refers to considering the long-range data and alleviating deficiently labeled sample problem.

3) DFL-LC is optimized through an iterative algorithm. The test results on three representative datasets demonstrate that the DFL-LC method is superior to the relevant latest HSI classification methods.

II. Related Works

A. Feature Extraction

There is abundant spatial and spectral information in the HSI, which is important to efficiently and accurately exploit spatial and spectral features to classify HSIs. According to the label of data, classification methods can be divided into supervised, semisupervised, and unsupervised methods.

We need a large amount of labeled data in supervised methods. Liu *et al.* [6] proposed an effective classification model based on active learning and DBN, in which the active learning algorithm is used to repeatedly select high-quality labeled samples for training, and DBN is used to deeply extract spectral features. In [28], a diversified DBN model was proposed, in which the classification performance of the model is significantly improved by normalizing the DBN pretraining and fine-tuning progress. Semisupervised methods need less labeled data compared with supervised methods. In [29], a semisupervised deep feature

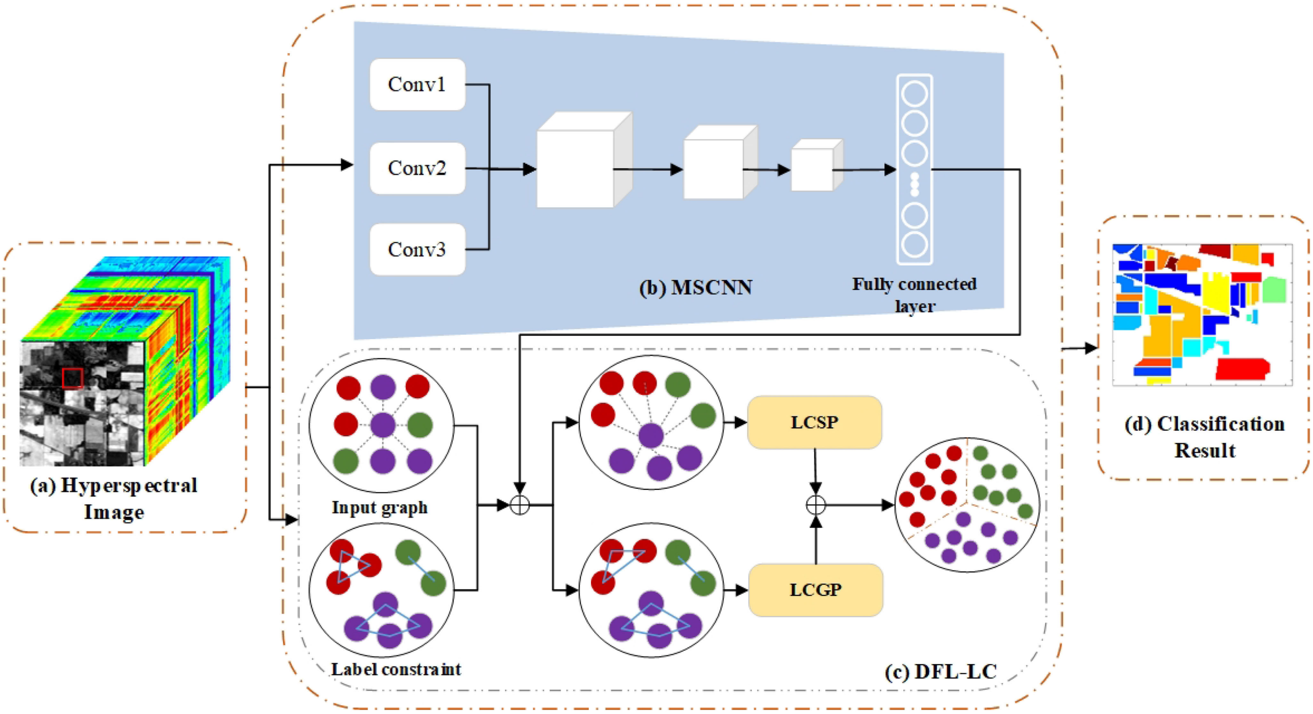


Fig. 1. Framework of DFL-LC. (a) HSI data. In (b), the features of samples are extracted by the MSCNN. (c) Learning process of the GCN. (d) Result of the HSI classification. In (c), the samples of different colors represent different classes. The adjacency matrix is constructed according to the relationship between samples and is combined with the basic features extracted by the MSCNN as the input of graph convolution. Finally, we embed the LCSP and LCGP constraints into the objective function to ensure LC.

learning method was proposed with feature consistency, where the CNN is used to extract spectral–spatial features, and fully connected layers are used to model feature consistency. Sun *et al.* [14] proposed a semisupervised method to obtain features by training SAE using a batch training scheme. Then, a mean convergence method is used to generate deep characteristics by further fusing the spectral and local spatial features. HSI data can often be represented as a 3-D cube. Therefore, it can provide a more effective method to simultaneously extract spectral and spatial features of HSIs by performing 3-D convolution in spectral and spatial dimensions. In [30], a subspace learning with the conditional random field (CRF) method was developed to obtain the subspace of the HSI pixels using the semisupervised approach, in which the CRF is embedded in subspace learning to classify HSI. Different from supervised and semisupervised methods, unsupervised methods do not use labeled data. Some traditional methods (PCA [31] and ICA [32]) can effectively extract spectral features, but these linear models only have simple linear processing, which makes it difficult to process complex spectral features in HSIs. Kuo *et al.* [33] proposed kernel nonparametric weighted feature extraction, which combined linear and nonlinear transformation.

However, the traditional CNN only refers to the spectral features of the adjacent pixels but ignores the spectral features of long-range data with the sample, which only perform convolutions with fixed size and weight on regular square image regions. Therefore, the GCN is introduced to realize HSI classification, which encodes the graph structure to consider long-range data.

B. *Graph Convolution*

The GCN has been extensively explored in the problem areas of supervised, semisupervised, and unsupervised networks. Gori *et al.* first proposed the concept of graph neural network, which can process graph data [34]. Compared with the CNN and the RNN, the advantage of the GCN is that it can process non-Euclidean data with graph structure. The GCN is a multilayer neural network that operates directly on a graph and studies the features of the graph through the eigenvalues and eigenvectors of the Laplace matrix of the graph. The GCN can correctly transform the graph into a new discriminative space by integrating the adjacency relationships and features of the nodes in the graph.

Recently, the GCN has been diffusely used for text classification. Hamilton *et al.* proposed an inductive framework named Graph SAGE, which efficiently generates node embeddings for previously unseen data utilizing node features [17]. By sampling and aggregating features from the local neighborhood of the node, it learns a function that generates embedding, instead of training individual embeddings for each node. A fast approximation localized graph convolution was proposed to avoid numerical instabilities and explosion or vanishing of gradients [18]. It can encode both features of node and graph structure and lead to more efficient filtering operations, because the GCN was simplified by the first-order approximation of graph convolution. Monti *et al.* proposed a unified framework that generalizes the

CNN to non-Euclidean domains, such as graphs and manifolds, and learns stationary and local features [19].

The GCN is also widely used in image classification. Garcia and Bruna used a reasoning prism to study the problem of few-shot learning on part of the graph observation model, which is composed of a set of input images that can be observed or not observed with labels [35]. Wang *et al.* proposed a method based on the GCN, which uses semantic embeddings and categorical relationships to classify images [36]. In this method, given a learning knowledge graph, the method inserts each node (representing a visual category) as input semantics. Some scholars also use the GCN to realize HSI classification. Qin *et al.* proposed a spectral–spatial GCN to approximate convolution by using adjacency nodes in the graph [37]. Thus, this method takes full advantage of the current pixel spatial information in the process of approximate convolution. Wan *et al.* proposed a multiscale dynamic GCN, whose graph is dynamically updated during graph convolution, and its input graphs have different neighborhood scales to utilize multiscale information in HSIs [38]. The GCN can capture relationships based on the predefined graph that contains global information, but the pixels' relationships for graph construction cannot be well ensured. Therefore, LC is embedded into the deep learning framework in this article to learn more features of HSIs.

III. Proposed Method

In this section, a new feature learning method, DFL-LC, is introduced. First, the motivation of this article is presented in Section III-A. Next, the DFL-LC framework is given in Section III-B. Finally, we optimize DFL-LC in Section III-C.

A. Motivation

Traditional convolution only refers to the spectral features of the adjacent pixels but ignores the spectral features of long-range data with the sample. The GCN cannot ensure the pixels' relationships for graph construction because accurate image features cannot be obtained with only the limited iteration of the deep learning framework. Therefore, we combine MSCNN and GCN, and LCSP and LCGP are added to the objective function to keep LC. LCSP denotes that the LC is maintained by calculating the error between the predicted output label and the real label of the sample. Moreover, LCGP refers to considering long-range data by introducing a group label matrix and realizing label reuse on the basis of limited labeled data. Finally, the objective function is optimized by an iterative optimization algorithm.

B. Framework of DFL-LC

1) Multiscale Feature Extraction: In recent years, classification, detection, and recognition issues can be addressed by CNNs, which are effected by the structure of the human visual system. There are two special aspects in the CNN architecture: shared weight and local connection, which make CNNs different from other deep learning methods in architecture. Shared weight can reduce network parameters. And the CNN can make use of local connections to exploit the local correlation between the neurons of near layers.

The objects of HSI usually have different geometric appearances, so multiscale features have been proven to be useful to solve the HSI problems [39]. The multiscale structure contains plentiful contextual HSI information [40]. Deep learning can extract abundant local characteristics of image regions from different levels by using the contextual information exposed by different scales. To obtain more detailed features, we embed multiscale information into the CNN. The MSCNN can exploit both shallow features and deep features, which is better adapted to classify HSIs, and multiscale features can effectively improve the results of HSI classification. Using the MSCNN, the spectral–spatial features are introduced to describe HSIs. The MSCNN adopts three different convolutional filters to locally convolve patches $\mathbf{X}^1$, $\mathbf{X}^2$, and $\mathbf{X}^3$ with three different sizes. Then, all the features extracted from these three layers are stacked together as the input to the fully connected layer. With the MSCNN process, we can obtain the spectral–spatial features $\mathbf{Z}_1$:

$$\begin{aligned}\mathbf{Z}_1 = f(&\mathrm{ReLU}(\mathbf{W}^{(0,1)} \otimes \mathbf{X}^1 + \mathbf{b}^{(0,1)})\\ &\oplus \mathrm{ReLU}(\mathbf{W}^{(0,2)} \otimes \mathbf{X}^2 + \mathbf{b}^{(0,2)})\\ &\oplus \mathrm{ReLU}(\mathbf{W}^{(0,3)} \otimes \mathbf{X}^3 + \mathbf{b}^{(0,3)}))\end{aligned} \tag{1}$$

where *f* is the fully connected operation. $\otimes$ represents the traditional convolution operation, and $\otimes$ represents the features that are added together in the third dimension. $\mathbf{W}^{(0,i)}$ and $\mathbf{b}^{(0,i)}$ are the weight and bias for $\mathbf{X}^i$.

2) Graph Convolution Process: There are hundreds of thousands of pixels in the HSI, which makes the computational complexity for graph convolution and HSI classification difficult to accept. In order to solve this problem, the GCN is introduced by treating each sample as a node in graph instead of a pixel of the HSI. This method can significantly reduce the number of graph nodes and improve the computational efficiency. Different from the CNN, which extracts features by convolution, the GCN studies the features of the graph through the eigenvalues and eigenvectors of the Laplace matrix of the graph. The GCN can find the simple and clear neighbor connections between the nodes from a complex graph and smooth the label information via neighbor connections over the graph until achieving a global steady state.

To perform graph convolution, we first construct an undirected graph, which is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. $\mathbf{A}$ denotes the adjacency matrix of $\mathcal{G}$, which represents the connection relationship between nodes in the graph. Here, the adjacency matrix is constructed according to the spatial relationship among patches, which can be calculated as follows:

$$\mathbf{A}_{ij} = \begin{cases} e^{-\gamma \|x_i - x_j\|^2}, & x_i \in N(x_j) \mathrm{or} x_j \in N(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where γ is empirically set to 0.01 in the experiments. x_i represents the patch, and $N(x_i)$ is the set of neighbors of x_i.

The normalized Laplacian of the graph is $\mathbf{L}=\mathbf{I}_N-\mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}=\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\mathbf{D}$ denotes the degree matrix of $\mathcal{G}$, $\mathbf{U}$ is the matrix composed of the eigenvectors of $\mathbf{L}$, $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{L}$, and $\mathbf{I}$ denotes the identity matrix with the proper size. According to the graph convolution theorem, the graph convolution can be written as

$$g * \mathbf{x} = \mathbf{U}\left(\mathbf{U}^T g \mathbf{U}^T \mathbf{x}\right). \tag{3}$$

Then, we can consider that $g_\theta(\boldsymbol{\Lambda})=\mathbf{U}^T g$ is a function of the eigenvalues of $\mathbf{L}$. We can define spectral convolutions on graphs as the multiplication of a signal x with a filter in the Fourier domain

$$g_\theta * \mathbf{x} = \mathbf{U} g_{\theta'}(\boldsymbol{\Lambda})\mathbf{U}^T\mathbf{x} = \mathbf{U} g_\theta \mathbf{U}^T \mathbf{x} \tag{4}$$

where $g_\theta=\text{diag}(\theta)=g_{\theta'}(\boldsymbol{\Lambda})$ is the filter.

However, the amount of computation required to eigendecompose the $\mathbf{L}$ of large graphs is prohibitively expensive. To address this problem, Hammond *et al.* approximated $g_\theta(\boldsymbol{\Lambda})$ up to the *K*th order by a truncated expansion according to Chebyshev polynomials $T_k(\mathbf{x})$ [41]

$$g_{\boldsymbol{\theta}'}(\boldsymbol{\Lambda}) \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k\left(\tilde{\boldsymbol{\Lambda}}\right) \tag{5}$$

with a rescaled $\tilde{\boldsymbol{\Lambda}}=\frac{2}{\lambda_{\max}}\boldsymbol{\Lambda}-\mathbf{I}_N$. $\lambda_{\max}$ is the maximum eigenvalues of $\mathbf{L}$ and $\boldsymbol{\theta}'$ is a vector of Chebyshev coefficients. Therefore, the convolution can be written as

$$g_{\boldsymbol{\theta}'} * \mathbf{x} \approx \sum_{k=0}^{K} \boldsymbol{\theta}'_k T_k\left(\tilde{\mathbf{L}}\right)\mathbf{x} \tag{6}$$

where $\tilde{\mathbf{L}}=\frac{2}{\lambda_{\max}}\mathbf{L}-\mathbf{I}_N$ is the scaled Laplacian matrix. $(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T)^k=\mathbf{U}\boldsymbol{\Lambda}^k\mathbf{U}^T$ can easily verify (6). Since this formula is a *K*th-order polynomial for the Laplacian, the nodes away from the central node at most *K* steps determine the filtering.

Therefore, in the form of (6), stacking graph convolutional layers can build a graph convolution network model, in which pointwise nonlinearity is after each layer. Therefore, (6) becomes a linear function on the Laplacian spectrum of the graph considering the first-order neighborhood ($K=1$). We further approximate $\lambda_{\max}\approx 2$ in this linear formulation of a GCN. Therefore, (6) can be simplified to

$$g_{\boldsymbol{\theta}'} * \mathbf{x} \approx \theta'_0\mathbf{x} + \theta'_1(\mathbf{L}-\mathbf{I}_N)\mathbf{x} = \theta'_0\mathbf{x} + \theta'_1\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x} \tag{7}$$

where θ'_0 and θ'_1 are two free parameters. To avoid overfitting caused by many parameters, (7) is converted to

$$g_\theta * \mathbf{x} \approx \theta\left(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)\mathbf{x} \tag{8}$$

with a single parameter $\theta=\theta'_0=-\theta'_1$. Since the eigenvalues of $\mathbf{I}_N+D^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ are in the range [0, 2], numerical instabilities and explosion or vanishing of gradients will be resulted by repeatedly using this operator. To alleviate this problem, Kipf and Welling performed the renormalization trick $\mathbf{I}_N+\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\rightarrow\mathbf{I}_N+\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ with $\tilde{\mathbf{A}}=\mathbf{A}+\mathbf{I}_N$ and $\tilde{\mathbf{D}}_{ii}=\sum_j\tilde{\mathbf{A}}_{ij}$ [18].

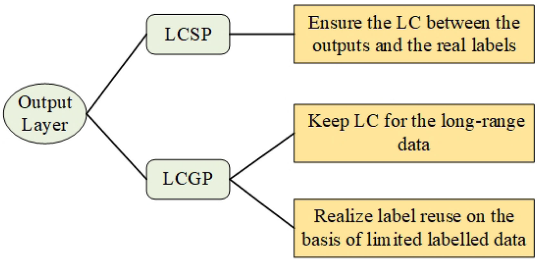

Output Layer
LCSP
LCGP
Ensure the LC between the outputs and the real labels
Keep LC for the long-range data
Realize label reuse on the basis of limited labelled data

Fig. 2. Overview of the output layer. LCSP and LCGP are proposed to keep LC.

Since the spectral–spatial features $\mathbf{Z}_1$ have been obtained, based on the graph convolution, we build a GCN, which contains two-layer graph convolution for node classification on a graph as an example. Therefore, the forward model can be simplified to

$$\mathbf{Z}=\psi(\mathbf{X},\mathbf{A})=\text{soft max}\left(\hat{\mathbf{A}}\text{ReLU}\left(\hat{\mathbf{A}}\mathbf{Z}_1\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right) \tag{9}$$

where $\hat{\mathbf{A}}=\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ and $\mathbf{W}^{(0)}\in\mathbb{R}^{C\times H}$ and $\mathbf{W}^{(1)}\in\mathbb{R}^{H\times M}$ are the weight matrixes of the input-to-hidden layer and the hidden-to-output layer, which can be updated via the backpropagation algorithm.

3) Output Layer: In the output layer, we formulate two kinds of constraints to boost the classification accuracy: LCSP and LCGP constraints, which are shown in Fig. 2. LCSP is maintained by calculating the error between the label prediction and the real label of the sample. LCGP refers to considering the long-range data by introducing a group label matrix and realizing label reuse on the basis of limited labeled data. Let *s* be the number of labeled patches, q be that of unlabeled patches, and $n=s+q$ be the number of patches. The cross-entropy loss and LC constraint train the features extracted from the MSCNN and the GCN with the labeled data.

LCSP Loss: In the GCN, the output layer is activated by the ReLU activation function to transmit the features into the probability of all class labels $\mathbf{Z}$. The LCSP loss is used to calculate the difference between the output of the network $\mathbf{Z}$ and the real label $\mathbf{Y}$

$$\Theta_{\text{LCSP}}=-\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{L}I(j)\log\mathbf{Z}_{ij} \tag{10}$$

where L is the number of classes and $\mathbf{Z}_i$ is the label prediction for the *i*th patch. The value of *I(j)* is 1 when *j* equals the desired label $\mathbf{Y}_i$ of the *i*th patch ($1\leq i\leq s$); otherwise, the value is 0.

In (10), the probability of all class labels is predicted and optimized using the cross-entropy loss.

LCGP Loss: The LCGP is achieved by introducing the group label matrix $\mathbf{G}$, which considers the long-range data and realizes label reuse. For example, assume that patches $\mathbf{X}_1$ are from class 1; $\mathbf{X}_2$ and $\mathbf{X}_3$ are from class 2; and $\mathbf{X}_4$ is from class 3. Then, G is defined as

$$\mathbf{G} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

The objective function of the LC is then described as

$$\Theta_{\text{LCGP}} = \|\mathbf{G} - \mathbf{TZ}\|_F^2 + \alpha \|\mathbf{T}\|_F^2 \tag{12}$$

where $\mathbf{T} \in \mathbb{R}^{s\times L}$ is a transformation matrix for transforming the predicted label $\mathbf{Z} \in \mathbb{R}^{L\times s}$ into the matrix of the same size as $\mathbf{G} \in \mathbb{R}^{s\times s}$, and α is the balance term.

4) Overall Objective Function of DFL-LC: Considering the constraints of (10) and (12), we formulate the joint objective function of DFL-LC as follows:

$$\begin{aligned} \Theta &= \Theta_{\text{LCSP}} + \lambda\Theta_{\text{LCGP}} \\ &= -\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{L} I(j)\log \mathbf{Z}_{ij} + \lambda\left[\|\mathbf{G} - \mathbf{TZ}\|_F^2 + \alpha\|\mathbf{T}\|_F^2\right] \end{aligned} \tag{13}$$

where λ is the balance term.

C. Optimization of DFL-LC

In this section, we propose an iterative algorithm to optimize the parameters in the DFL-LC, and Algorithm 1 summarizes the optimization procedure. Let φ_1 be the collection of weights and bias of the CNN, and φ_2 be the collection of weights of the GCN. In each iteration of the algorithm, the parameters φ_1, φ_2, and $\mathbf{T}$ are optimized.

The parameters φ_1 and φ_2 are solved when $\mathbf{T}$ is fixed, so the optimization problem defined in (13) can be rewritten as

$$\begin{aligned} \min_{\varphi_1,\varphi_2} \Theta = \min_{\varphi_1,\varphi_2} -\frac{1}{s}\sum_{i=1}^{s}\sum_{j=1}^{L} I(j)\log \mathbf{Z}_{ij} \\ + \lambda\left[\|\mathbf{G} - \mathbf{TZ}\|_F^2\right]. \end{aligned} \tag{14}$$

Then, we update the parameters on each iteration

$$\begin{aligned} \varphi_1 &\leftarrow \varphi_1 - \beta\nabla_{\varphi_1}\left[\min_{\varphi_1,\varphi_2}\Theta\right] \\ \varphi_2 &\leftarrow \varphi_2 - \beta\nabla_{\varphi_2}\left[\min_{\varphi_1,\varphi_2}\Theta\right] \end{aligned} \tag{15}$$

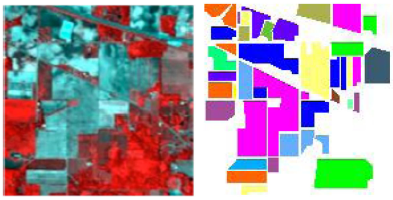
where β is the learning rate of the DFL-LC.

$\mathbf{T}$ is resolved when φ_1 and φ_2 are fixed. Therefore, the optimization problem defined in (13) is rewritten as

$$\min_{\mathbf{G}} \Theta = \lambda\left[\|\mathbf{G} - \mathbf{TZ}\|_F^2 + \alpha\|\mathbf{T}\|_F^2\right]. \tag{16}$$

Then, we update the transform matrix $\mathbf{T}$ as

$$\mathbf{T} \leftarrow \mathbf{T} - \beta\nabla_{\mathbf{G}}\left[\min_{\mathbf{G}}\Theta\right]. \tag{17}$$



(a) (b)

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

Algorithm 1: DFL-LC.

```
Input: Training samples $\mathbf{X}$, matrix $\mathbf{G}$, parameters $\lambda$, $\alpha$.
Output: Predicted class labels $\mathbf{Z}$.
Initialize: $\varphi_1$, $\varphi_2$, and $\mathbf{T}$.
for the number of iteration do
    Sample batch size of samples $\mathbf{X}$;
    Compute the features from the MSCNN using (1);
    Feed the features extracted from the MSCNN into the GCN;
    Compute the features from the GCN using (9);
    Compute the class probability $\mathbf{Z}$ using (9);
    Compute the LCSP loss using (10);
    Compute the LCGP loss using (12);
    Update $\varphi_1$ and $\varphi_2$ using (15);
    Update $\mathbf{T}$ using (17);
```

IV. Experiments

In this section, we experimentally evaluate the performance of the DFL-LC to classify HSIs. First, the datasets are introduced in Section IV-A. Next, the experimental settings of DFL-LC and approaches are given in Sections IV-B and IV-C. Finally, we give the classification results and the analysis of parameters in Sections IV-D and IV-E.

A. Dataset

1) Indian Pines: The Indian Pines dataset is over the Indian Pines test site in north-western Indiana, which collected by the AVIRIS sensor. In the Indian Pines dataset, there is one-third of forest or other natural perennial vegetation and two-thirds of agriculture. The dataset contains 145×145 pixels and 220 bands. After removing 20 bands that are water absorption and noisy, 200 bands are reserved. The Indian Pines ground truth contains 16 classes. Fig. 3 shows the false color image and ground-truth map of the dataset, and Table I lists the number of labeled and unlabeled pixels of various classes.

2) Salinas: The spatial resolution of Salinas dataset was 3.7 m, which was collected by the 224-band AVIRIS sensor over Salinas Valley, CA, USA. After removing 20 water absorption bands, the image comprises 204 bands with 512×217 pixels. It includes vegetables, bare soils, and vineyard fields. The ground truth of Salinas contains 16 classes. The false color image and ground-truth map are shown in Fig. 4, and Table II shows the numbers of samples to train and test in the Salinas dataset, respectively.

TABLE I
NUMBERS OF TRAINING AND TEST SAMPLES IN THE INDIAN PINES DATASET

ID	Class	Training(10%)	Test(90%)
1	Alfalfa	5	41
2	Corn-notill	143	1285
3	Corn-mintill	83	747
4	Corn	24	213
5	Grass-pasture	49	434
6	Grass-trees	73	657
7	Grass-pasture-mowed	3	25
8	Hay-windrowed	48	430
9	Oats	2	18
10	Soybean-notill	98	874
11	Soybean-mintill	246	2209
12	Soybean-clean	60	533
13	Wheats	21	184
14	Woods	127	1138
15	Buildings-grass-trees-drives	39	347
16	Stone-steel-towers	10	83
	Total	1031	9128

TABLE II
NUMBERS OF TRAINING AND TEST SAMPLES IN THE SALINAS DATASET

ID	Class	Training(10%)	Test(90%)
1	Brocoli_green_weeds_1	201	1808
2	Brocoli_green_weeds_2	373	3353
3	Fallow	198	1778
4	Fallow_rough_plow	140	1254
5	Fallow_smooth	268	2410
6	Stubble	396	3563
7	Celery	358	3221
8	Grapes_untrained	1128	10143
9	Soil_vinyard_develop	621	5582
10	Corn_senesced_green_weeds	328	2950
11	Lettuce_romaine_4wk	107	961
12	Lettuce_romaine_5wk	193	1734
13	Lettuce_romaine_6wk	92	824
14	Lettuce_romaine_7wk	107	963
15	Vinyard_untrained	727	6541
16	Vinyard_vertical_trellis	181	1626
	Total	5418	48711

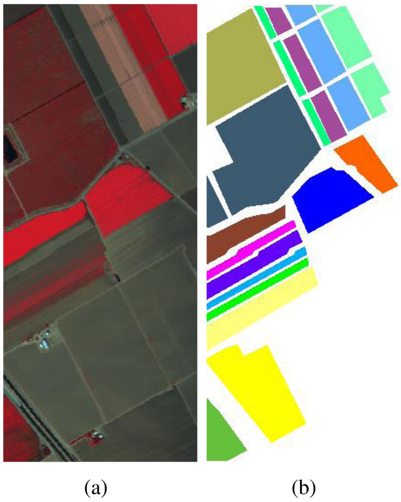


Fig. 4. Salinas. (a) False color image. (b) Ground-truth map.

3) University of Pavia: The University of Pavia dataset is with a spatial resolution of 1.3 m during a flight over Pavia in northern Italy, which was obtained by the ROSIS sensor. The scene is 610 × 340 × 103 after removing 12 noisy bands. The University of Pavia dataset contains nine classes of interest. Fig. 5 shows the false color image and the ground-truth map of the University of Pavia dataset, and Table III shows the information of training and test samples on the number.

B. Experimental Settings

The proposed method is enforced through PyTorch with the Adam optimizer, and the backpropagation algorithm is used to optimize the parameters of the whole network. For learning the network, we set the learning rate to 0.01 with 9000 epochs and a hidden layer size of 24 units. We crop each pixel and its surrounding neighboring pixels as the input of DFL-LC. The datasets we use in the experiment are Indian Pines, Salinas, and University of Pavia. Training samples are selected from 10% of the samples in each class, and other samples are used to test to evaluate the classification performance.

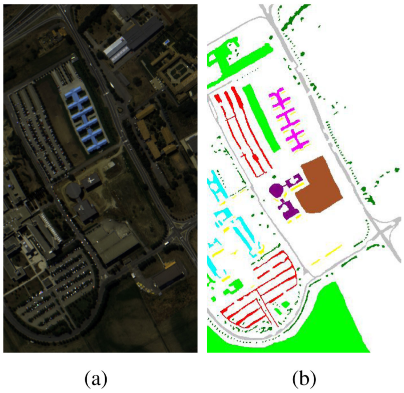


Fig. 5. University of Pavia. (a) False color image. (b) Ground-truth map.

TABLE III
NUMBERS OF TRAINING AND TEST SAMPLES IN THE UNIVERSITY OF PAVIA DATASET

ID	Class	Training(10%)	Test(90%)
1	Asphalt	664	5967
2	Meadows	1865	16784
3	Gravel	210	1889
4	Trees	307	2757
5	Painted metal sheets	135	1210
6	Bare Soil	503	4526
7	Bitumen	133	1197
8	Self-Blocking Bricks	369	3313
9	Shadows	95	852
	Total	4281	38495

TABLE IV
ACCURACY OF EACH CLASS, AND OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT CLASSIFICATION METHODS ON THE INDIAN PINES DATASET

ID	SVM	CNN	CNN-MRF	HybridSN	SSCNN	SDP	DFL-LC
1	75.61	97.22	95.12	**100.00**	87.80	82.93	90.24
2	77.82	89.85	91.83	90.17	93.70	94.86	**95.88**
3	65.33	84.09	91.97	92.77	90.23	87.01	**95.31**
4	69.01	84.73	77.93	**95.48**	82.16	75.59	87.79
5	89.40	**99.04**	98.16	96.28	91.24	95.39	97.47
6	95.74	95.44	99.09	96.83	97.56	97.41	**99.39**
7	84.00	**100.00**	56.00	**100.00**	92.00	88.00	**100.00**
8	99.30	94.42	99.53	99.31	**100.00**	**100.00**	**100.00**
9	38.89	85.71	**100.00**	77.78	33.33	38.89	**100.00**
10	81.35	86.42	84.78	87.17	91.30	**95.19**	89.36
11	82.66	86.74	96.88	94.26	93.53	95.29	**97.60**
12	70.36	90.79	82.18	**95.13**	84.05	80.86	93.06
13	96.20	95.98	**100.00**	95.34	96.20	98.91	**100.00**
14	92.27	97.85	97.19	96.58	99.38	**99.30**	99.03
15	55.91	84.57	86.74	88.27	**92.80**	91.35	92.22
16	84.34	82.35	90.36	81.05	79.52	79.52	**100.00**
OA	81.81	90.23	93.23	93.37	93.22	93.71	**96.16**
AA	78.64	90.95	90.48	92.90	87.80	87.53	**96.08**
Kappa	79.22	88.84	92.26	92.44	92.26	92.83	**95.62**

C. Comparison Approaches

To verify an evaluate the classification ability of the proposed DFL-LC, other traditional and state-of-the-art methods for HSI classification (SVM [2], CNN [11], CNN-MRF [42], HybridSN [43], SSCNN [44], and SDP [45]) are also used for comparison. We compare CNN-GCN with DFL-LC to verify the validity of MSCNN.

1) SVM [2]: Combining SVM with a feature-reduction technique is sufficient in HSI classification.

2) CNN [11]: It is a 3-D CNN model to effectively extract spectral and spatial for HSI classification.

3) CNN-MRF [42]: The CNN is used to learn the posterior class distributions, and then, Markov random field prior is used to consider the spatial information.

4) HybridSN [43]: It is a spectral–spatial 3-D CNN followed by spatial 2-D CNN. The 3-D CNN can represent spectral and spatial features, and the 2-D CNN can further learn more spatial features.

5) SSCNN [44]: It is a novel semisupervised CNN to classify HSIs, which can automatically learn features from complex data structures.

6) SDP [45]: It is a new semisupervised active learning approach to classify HSIs that improves machine generalization by using pseudo-labeled samples.

D. Classification Results

In these experiments, three objective metrics (overall accuracy (OA), average accuracy (AA), and the Kappa coefficient) adopted are used to quantitatively evaluate the capability of DFL-LC and other methods. The OA is obtained by calculating the ratio of the number of correctly classified test samples to the total number of test samples. The AA is the average of the classification accuracies of each class. The Kappa coefficient represents the robust measure of the degree of consistency, which is calculated by weighting the classification accuracies. The experiments are conducted on Indian Pines, Salinas, and University of Pavia datasets. The quantitative classification results are summarized in Tables IV–VI, and the highest accuracy in each class is highlighted in bold. And the classification maps obtained by different methods are shown in Figs. 6–8. Therefore, we can obtain the following observations.

1) Compared with other methods, the DFL-LC can achieve a higher classification accuracy and the best performance on three datasets than other methods. It demonstrates that DFL-LC can learn more representative features of HSI, which considers long-range data and keeps LC. Compared with the ground-truth map and other classification maps, the result of DFL-LC method shows fewer misclassifications and produces a smoother visual effect. This indicates that DFL-LC is very useful to classify HSIs, which can effectively construct the relationships among the samples.

2) We can observe that the CNN-based methods, including CNN, CNN-MRF, HybridSN, SSCNN, and SDP, achieve relatively low accuracy combined with DFL-LC. The reason is that they can only perform convolutions on a regular image region and cannot extract specific local spatial information. It also proves that GCN and LCGP can consider spectral features of long-range data, which play a significant role in HSI classification.

3) By contrast, we also observe that the DFL-LC methods can yield relatively good performance compared with SSCNN and SDP, which are semisupervised classification methods. It explains that LC can realize label reuse based on limited labeled data to improve the feature learning ability of DFL-LC.

E. Parameters Analysis

1) Impact of γ in the Adjacency Matrix: In the proposed method, the calculation method of the adjacency matrix in this article is shown in (2). It can be seen that different values of γ affect the classification accuracy in (2). Thus, we vary the

TABLE V
ACCURACY OF EACH CLASS, AND OA, AA (%), AND KAPPA COEFFICIENT ACHIEVED BY DIFFERENT CLASSIFICATION METHODS ON THE SALINAS DATASET

ID	SVM	CNN	CNN-MRF	HybridSN	SSCNN	SDP	DFL-LC
1	99.56	99.50	99.89	99.83	99.50	99.56	**100.00**
2	99.55	99.94	99.94	99.91	99.88	99.94	**100.00**
3	99.61	99.78	84.59	**99.83**	99.49	99.61	99.49
4	99.44	98.96	99.76	99.28	99.92	99.52	**100.00**
5	98.13	99.46	98.80	99.29	98.76	99.21	**99.83**
6	99.83	**100.00**	99.97	99.75	99.94	99.94	**100.00**
7	99.88	**100.00**	99.60	99.63	99.69	99.63	99.63
8	91.53	90.56	90.10	89.87	88.88	89.62	**96.34**
9	99.96	99.71	99.95	99.91	99.68	99.50	**99.98**
10	95.59	97.73	96.68	99.46	**98.14**	98.10	96.88
11	98.02	99.48	98.23	99.69	**99.90**	99.79	96.25
12	99.94	99.94	99.48	**100.00**	99.94	99.77	**100.00**
13	97.33	99.88	**100.00**	99.76	99.64	98.42	**100.00**
14	98.23	98.86	99.58	99.48	**99.90**	96.68	98.23
15	65.48	85.52	**92.25**	85.64	89.36	85.40	91.27
16	99.20	98.65	**99.69**	99.38	99.51	99.20	98.71
OA	92.99	95.75	95.95	95.76	95.94	95.46	**97.67**
AA	96.33	97.99	97.41	98.17	98.25	97.74	**98.54**
Kappa	92.18	95.27	95.50	95.28	95.49	94.94	**97.41**

TABLE VI
ACCURACY OF EACH CLASS, AND OA, AA (%), KAPPA COEFFICIENT ACHIEVED BY DIFFERENT CLASSIFICATION METHODS ON THE UNIVERSITY OF PAVIA DATASET

ID	SVM	CNN	CNN-MRF	HybridSN	SSCNN	SDP	DFL-LC
1	93.78	97.34	96.65	96.16	94.79	95.22	**98.41**
2	97.88	98.86	96.66	99.22	98.73	98.50	**99.74**
3	80.62	78.61	82.58	88.62	91.48	**92.22**	91.64
4	94.49	**98.04**	96.55	96.77	**98.04**	96.23	98.01
5	99.42	**100.00**	**100.00**	99.34	99.59	**100.00**	**100.00**
6	90.46	94.41	**95.67**	95.25	93.15	90.65	95.16
7	87.80	**94.49**	80.7	89.97	87.47	89.14	90.14
8	91.58	90.71	92.76	92.60	94.05	94.02	**98.94**
9	99.88	99.77	99.77	99.77	99.88	**100.00**	**100.00**
OA	94.52	96.26	95.18	96.74	96.36	96.00	**98.12**
AA	92.88	94.69	93.48	95.30	95.24	95.11	**96.89**
Kappa	92.72	95.04	93.63	95.67	95.17	94.69	**97.50**

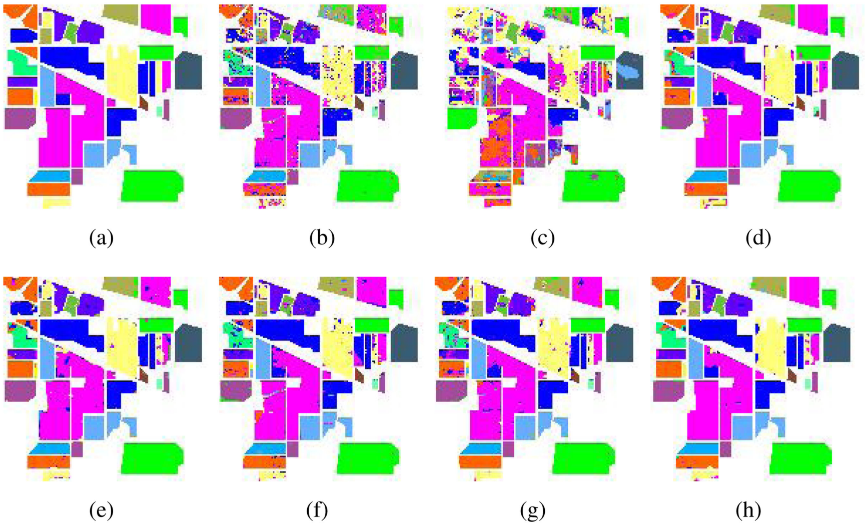


Fig. 6. Classification maps obtained by different classification methods on the Indian Pines dataset. (a) Ground-truth map. (b) SVM. (c) CNN. (d) CNN-MRF. (e) HybirdSN. (f) SSCNN. (g) SDP. (h) DFL-LC.

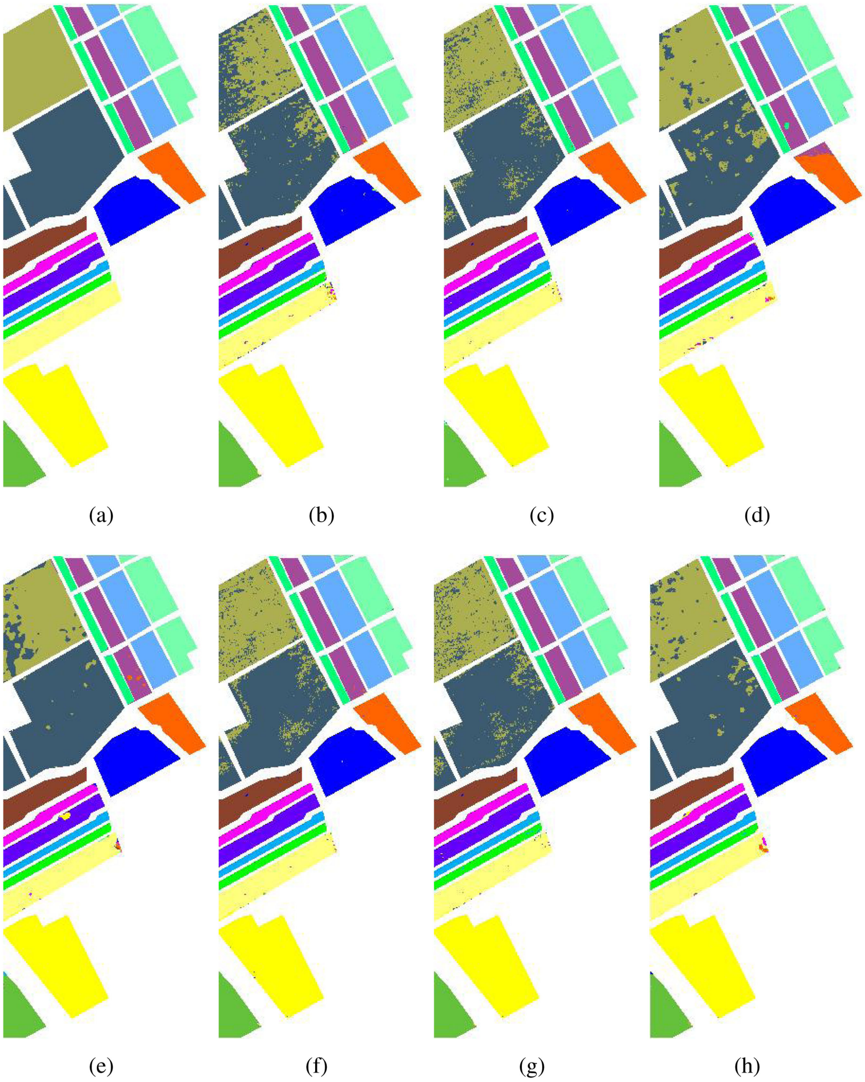


Fig. 7. Classification maps obtained by different classification methods on the Salinas dataset. (a) Ground-truth map. (b) SVM. (c) CNN. (d) CNN-MRF. (e) HybirdSN. (f) SSCNN. (g) SDP. (h) DFL-LC.

value of γ from 0.001 to 0.1 and report the OA on the Indian Pines, Salinas, and University of Pavia datasets. The results of classification are shown in Fig. 9. We can observe that the classification accuracy is improved when the maximal value of the nonzero elements in the adjacency matrix approaches 1. According to the experimental results, we can find that the classification accuracy is highest when γ is 0.01. Moreover, the capability of DFL-LC is more stable than that of CNN-GCN with a changed value of γ.

2) Impact of the Number of Hidden Nodes: The GCN learns feature by encoding features of node and graph structure in the hidden layer. There is a hidden layer in the proposed method, and the number of nodes of hidden layer also has some influence on the classification result. Therefore, we vary the number of hidden nodes in 16, 24, and 32. The OA on the Indian Pines, Salinas, and University of Pavia datasets are shown in Fig. 10. We can observe that the classification accuracy is improved when the number of hidden nodes is more than the number of categories in the dataset. Thus, we choose the number of hidden nodes according to the best accuracy in the experiment, so 16 hidden nodes are set in the method.

3) Influence of the Multiscale Filter Bank: To verify the validity of filter banks with different scales in extracting feature information, we compare the filter banks with different configurations, which are 1×1, $\sim 3 \times 3$, $\sim 5 \times 5$ and $\sim 7 \times 7$. The $\sim 7 \times 7$ denotes that the sizes of the convolutional filters are 1×1, 3×3, 5×5 and 7×7, and the others are in the same way. As shown in Table VII, the classification accuracy of multiscale filters is better than that of the method with a 1×1 convolutional filter. Multiscale convolution can exploit the spatial–spectral feature, which is better adapted for HSI classification. Additionally, since $\sim 7 \times 7$ contains more noise, the $\sim 5 \times 5$ multiscale filter shows better performance.

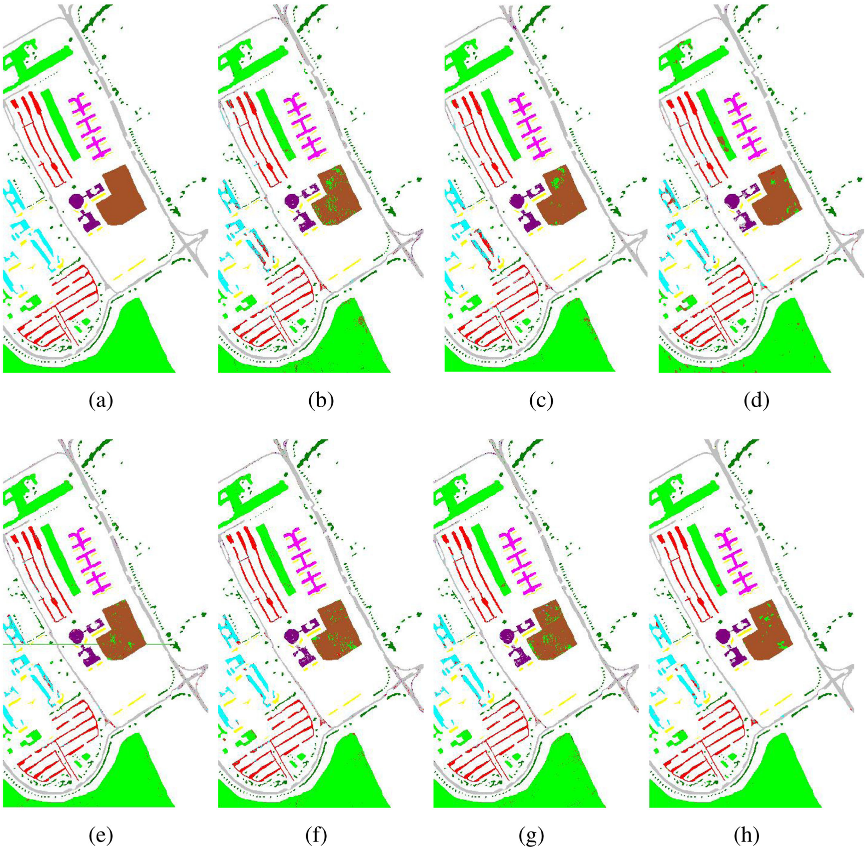


Fig. 8. Classification maps obtained by different classification methods on the University of Pavia dataset. (a) Ground-truth map. (b) SVM. (c) CNN. (d) CNN-MRF. (e) HybirdSN. (f) SSCNN. (g) SDP. (h) DFL-LC.

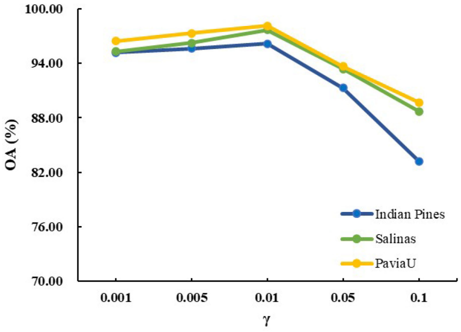


Fig. 9. Overall accuracies of DFL-LC on Indian Pines, Salinas, and University of Pavia datasets under different value of γ.

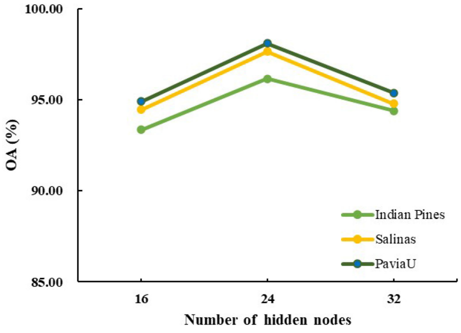


Fig. 10. Overall accuracies of DFL-LC on Indian Pines, Salinas, and University of Pavia datasets under different number of hidden nodes.

TABLE VII
OVERALL ACCURACIES (%) OF DFL-LC AND CNN-GCN ON INDIAN PINES, SALINAS, AND UNIVERSITY OF PAVIA DATASETS

Dataset	Indian Pines	Salinas	University of Pavia
11	81.57	84.59	85.20
$\tilde{3}3$	89.40	92.58	93.36
$\tilde{5}5$	96.16	97.67	98.12
$\tilde{7}7$	95.12	96.55	97.53

V. CONCLUSION

In order to effectively extract features and keep LC, we propose a novel DFL-LC to achieve HSI classification, which is based on traditional convolution and graph convolution. In DFL-LC, the MSCNN is used to obtain basic features, the GCN can capture relationships between pixels and realize HSI classification, and LCSP and LCGP are embedded in the objective function. LCSP can ensure LC between the predicted label and the real label of the sample. DFL-LC is a semisupervised method, and the method considers the truthful neighborhood information of all samples. LCGP can ensure the quality of extracted features when a small number of labeled samples are obtained, so DFL-LC can alleviate the deficiently labeled sample problem. Compared with the traditional and state-of-the-art classification methods, the experimental result demonstrates that the proposed method can yield better HSI classification performance. In future research, DFL-LC will be applied into other recognition tasks, such as high-spatial-resolution remote sensing image segmentation.

References

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

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

[3] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectralcspatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[4] B. Du and L. Zhang, "Target detection based on a dynamic subspace," *Pattern Recognit.*, vol. 47, no. 1, pp. 344–358, 2014.

[5] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogrammetry Remote Sens.*, vol. 142, pp. 344–357, 2018.

[6] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Top Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, Feb. 2017.

[7] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sens.*, vol. 2015, Jul. 2015, Art. no. 258619.

[8] 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.

[9] 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.

[10] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.

[11] 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.

[12] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[13] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[14] X. Sun, F. Zhou, J. Dong, F. Gao, Q. Mu, and X. Wang, "Encoding spectral and spatial context information for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2250–2254, Dec. 2017.

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

[16] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sens.*, vol. 8, no. 2, Jan. 2016, Art. no. 99.

[17] W. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," Jun. 2017, *arXiv:1706.02216*.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[19] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," 2016, *arXiv:1611.08402*.

[20] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," 2015, *arXiv:1511.02136*.

[21] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph LSTM," 2016, *arXiv:1603.07063*.

[22] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," 2017, *arXiv:1711.09869*.

[23] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," 2018, *arXiv:1801.07829*.

[24] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for RGBD semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5209–5218.

[25] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 20–28.

[26] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, "Rethinking knowledge graph propagation for zero-shot learning," 2018, *arXiv:1805.11724*.

[27] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang, "Multi-label zero-shot learning with structured knowledge graphs," 2017, *arXiv:1711.06526*.

[28] P. Zhong, Z. Gong, S. Li, and C. Schonlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[29] Y. Cao, Y. Wang, J. Peng, C. Qiu, L. Ding, and X. Xiang, "SDFL-FC: Semisupervised deep feature learning with feature consistency for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2020.

[30] Y. Cao *et al.*, "SLCRF: Subspace learning with conditional random field for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2020.

[31] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.

[32] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.

[33] B. Kuo, C. Li, and J. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1139–1155, Apr. 2009.

[34] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, vol. 2, pp. 729–734.

[35] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," 2017, *arXiv:1711.04043*.

[36] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," 2018, *arXiv:1803.08035*.

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

[38] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multi-scale dynamic graph convolutional network for hyperspectral image classification," 2019, *arXiv:1905.06133*.

[39] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectralcspatial classification of hyperspectral images with a superpixel-based discriminative sparse model," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4186–4201, Aug. 2015.

[40] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3904–3908.

[41] 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, 2011.

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

[43] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.

[44] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, 2017.

[45] C. Liu, J. Li, and L. He, "Superpixel-based semisupervised active learning for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 1, pp. 357–370, Jan. 2019.

Siyuan Liu is currently working toward the master's degree with the School of Land Science and Technology, China University of Geosciences, Beijing, China.

Her research interests include deep learning and remote sensing image classification.

Yun Cao is currently working toward the Ph.D. degree with the School of Land Science and Technology, China University of Geosciences, Beijing, China.

Her research interests include deep learning and remote sensing image processing.

Yuebin Wang (Member, IEEE) received the Ph.D. degree in remote sensing from the School of Geography, Beijing Normal University, Beijing, China, in 2016.

He was a Postdoctoral Researcher with the School of Mathematical Sciences, Beijing Normal University. He is currently an Associate Professor with the School of Land Science and Technology, China University of Geosciences, Beijing. His research interests include remote sensing imagery processing and 3-D urban modeling.

Junhuan Peng received the Ph.D. degree in geodesy from Wuhan University, Wuhan, China, in 2003.

He is currently a Professor with the School of Land Science and Technology, China University of Geosciences, Beijing, China. His research interests include temporal–spatial data analysis, surveying adjustment, applied statistics, and their associated application in surveying engineering, image geodesy, remote sensing, and satellite geodesy.

P. Takis Mathiopoulos (Senior Member, IEEE) received the Ph.D. degree in digital communications from the University of Ottawa, Ottawa, ON, Canada, in 1989.

From 1982 to 1986, he was with Raytheon Canada Ltd., working in the areas of air navigational and satellite communications. In 1989, he joined as an Assistant Professor the Department of Electrical and Computer Engineering, University of British Columbia (UBC), Vancouver, Canada, and where he was a faculty member until 2003, holding the rank of Professor from 2000 to 2003. From 2000 to 2014, he was the Director and then the Director of Research of the Institute for Space Applications and Remote Sensing, National Observatory of Athens. Since 2014, he has been a Professor of Telecommunications with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece. He also held visiting faculty long-term honorary academic appointments as Guest Professor with South West Jiao Tong University, Chengdu, China, and a Guest (Global) Professor with Keio University, Tokyo, Japan. He has delivered numerous invited presentations, including plenary and keynote lectures, and has taught many short courses all over the world. He has coauthored 135 journal papers published mainly in various IEEE journals, one book (edited), five book chapters, and more than 140 conference papers in the areas of his research activities and contributions, which have dealt with wireless terrestrial and satellite communication systems and network as well as in remote sensing, LiDAR systems, and information technology, including blockchain systems.

Dr. Mathiopoulos has been or is currently on the editorial board of several archival journals, including the *IET Communications* as an Area Editor, the IEEE TRANSACTIONS ON COMMUNICATIONS, the *Remote Sensing Journal*, and as a Specialty Chief Editor of the *Arial and Space Network Journal* of Frontiers. From 2001 to 2014, he was a Greek Representative to high-level committees in the European Commission and the European Space Agency. He has been a member of the Technical Program Committees (TPC) for numerous IEEE and other international conferences and was the TPC Vice-Chair of several IEEE conferences. As a faculty member UBC, he was the recipient of an Advanced Systems Institute (ASI) Fellowship as well as a Killam Research Fellowship. He was also the co-recipient of two best conference paper awards and the Satellite and Space Communication Technical Committee 2017 Distinguished Service Award for outstanding contributions in the field of Satellite and Space Communications from the IEEE Communication Society.

Yong Li received the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, Shenyang, China, in 2020.

He is currently an Assistant Professor with the School of Electrical Engineering, Guangxi University, Nanning, China. His research interests include intelligent robots, point cloud processing, computer vision, and pattern recognition.