Hyperspectral Imagery Classification via Random Multigraphs Ensemble Learning

Yanling Miao, Mulin Chen, Yuan Yuan, *Senior Member, IEEE*, Jocelyn Chanussot, *Fellow, IEEE*, and Qi Wang, *Senior Member, IEEE*

***Abstract*—Hyperspectral imagery (HSI) classification, which attempts to assign hyperspectral pixels with proper labels, has drawn significant attention in various applications. Recently, the graph-based semisupervised learning (SSL) approaches have shown the outstanding ability to handle the situation of limited labeled data for classification task. However, it is hard to construct the pairwise adjacent graph due to the high dimensionality of hyperspectral data. Besides, the graph-based SSL models are usually decided by a single classifier, which fail to effectively learn the complex structures and intrinsic properties of HSI. To address these problems, we propose a novel graph-based SSL classification model for HSI, which is based on random multigraphs construction and ensemble strategy (RMGE). Specifically, the anchor graph (AG) is constructed with spatial–spectral features, which integrates the spatial characteristics extracted by local binary pattern on each selected spectrum, preserving fine structures of local region. In order to enhance the discriminative capability of the classifier and avoid the trivial solution, the maximum entropy regularization is added into adjacent AG model. In addition, to capture the diversity of HSI data effectively, we design the ensemble framework by employing multiple AGs to learn HSI features. Experiments conducted on real hyperspectral datasets indicate that the proposed RMGE shows better performance than that of state-of-the-art approaches.**

***Index Terms*—Ensemble learning, graph-based semisupervised learning (SSL), hypersectral imagery, maximum entropy regularization.**

I. Introduction

HYPERSPECTRAL imagery (HSI) provides abundant spatial structures and spectral characteristics of the land cover classes, which is obtained by utilizing hyperspectral sensors mounted on different platforms [1], [2]. With hundreds of very narrow spectral bands, HSI usually can be used to distinguish subtle materials between the observed objects [3], [4]. Therefore, it has been applied to many fields, such as vegetation investigation [5], resource exploration [6], environmental monitoring [7], and target identification [8]. Among these applications, HSI classification (HSIC) is considered as one of active studies, which needs to predict a label for each pixel.

HSIC methods are divided into supervised, semisupervised, and unsupervised according to nature of available training samples. Lots of supervised approaches [9]–[12] have been developed for HSI task. He *et al.* [13] proposed a supervised model for HSIC with limited training samples, which extracts features exiting in and among samples and learns the label distribution. By doing this, the multiscale feature cutout is presented as the regularization technique, which relieves the overfitting problem. Feng *et al.* [14] combined collaborative learning and attention mechanism to generate high-quality samples for HSI, where the features of real multiclass samples assist the sample generation in the generator, improving the classification performance of the discriminator. These methods are usually based on pattern recognition algorithms, e.g., random forest, support vector machine (SVM) and k-nearest neighbor (kNN) [15]. The more training data is used to achieve the better classification performance of supervised approaches. However, there are some challenges to limit the effectiveness of HSIC, including the high dimensionality of HSI, limited labeled data, and large spatial variability. Therefore, the semisupervised learning (SSL) techniques have emerged and been more meaningful, where labeled and unlabeled samples are employed for classification model training. The SSL-based approaches contain generative models, co-training models, self-learning models, and graph-based models. Feng *et al.* [16] proposed a semisupervised model that is constructed by using the intraclass compactness constraint, which makes full use of both limited labeled and sufficient unlabeled samples. In the past two decades, the graph-based SSL models have obtained widespread attention [17], [18], which explore the pairwise adjacent graph between pixels and capture the data structure. Nevertheless, these approaches generally suffer from the high complexity associated with eigenvalue decomposition on graph Laplacian [19].

In order to handle this issue, the anchor graph (AG)-based methods [20]–[22] have been proposed, which chose a small part of points as anchors to encode the adjacent graph. Thus, the computational complexity is reduced effectively. Wang *et al.* [21] developed a scalable AG-based clustering method, which adds the nonnegative relaxation to AG model. With this, the clustering results are directly obtained without adopting k-means. He *et al.* [22] proposed fast SSL model with AG, which constructs a naturally sparse and scale invariant AG,

Manuscript received July 12, 2021; revised November 19, 2021; accepted November 30, 2021. Date of publication December 9, 2021; date of current version January 5, 2022. This work was supported by the National Natural Science Foundation of China under Grant U21B2041, Grant U1864204, Grant 61632018, and Grant 61825603. *(Corresponding author: Qi Wang.)*

Yanling Miao is with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: skilamiaomyl@gmail.com).

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

Mulin Chen, Yuan Yuan, and Qi Wang are with the School of Artificial Intelligence Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: chenmulin001@gmail.com; y.yuan1.ieee@gmail.com; crabwq@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3132993

alleviating the computation burden. He *et al.* [23] presented semisupervised model with bipartite graph, which calculates the labels of the data and anchors simultaneously and adopts the Woodbury matrix to handle the large matrix inverse. It obtains better accuracy in the shorter time and requires less storage spaces. However, these AG-based methods rarely utilize spatial characteristics within samples, where the discriminability of models is restricted.

A plenty of spatial–spectral combined algorithms [24]–[28] have been implemented through using spatial correlations and spectral similarity. Luo *et al.* [29] constructed an intraclass spatial–spectral hypergraph by joining the coordinate relationships and spectral similarity of neighboring pixels. The abovementioned approaches extract the feature in spatial usually according to the coordinate distance, making it unable to capture the fine features. Besides, most of them just consider the spatial structures in the HSI data preprocessing process, and fail to incorporate these inherent structures into the graph construction process [30], [31]. In addition, the aforesaid methods are constructed by a single graph structure, to lead the weak diversity learning of intrinsic property existing in the HSI data. To mitigate the phenomenon, the ensemble learning [32]–[34] causes more attention. However, there are few methods to apply the AG-based ensemble model for HSIC.

To handle the aforementioned drawbacks, we propose a novel graph-based SSL classification model combined random multigraphs construction and ensemble strategy (RMGE) for hypersectral data. We summarize the contributions as follows.

1) We employ the local binary pattern (LBP) model to encode the texture information as the spatial features, and use them in the graph construction procedure. LBP discovers the fine characteristics, which enhances the discriminability of the proposed RMGE.
2) We present the AG-based SSL scheme to learn the adjacent graph by allocating adaptive neighbors based on local distance, which effectively handles the high-dimensional problem. The maximum entropy regularization is applied for the graph learning to avoid the trivial solution.
3) We introduce the ensemble framework to obtain the optimal prediction model, which makes multiple AGs to be built in parallel. Thus, it can guarantee the diversity learning of HSI data and the efficiency of the proposed RMGE.

Compared with the conference version of the research in [35], the improvements of this article include that introducing more technical modules and providing more experiments. Specifically, Section III introduces the weighted mean filtering (WMF) module to reduce the impact of noisy pixels in HSI and guarantee the consistency of pixels within a local region. Section III-C proposes the maximum entropy regularization for the graph construction, which makes the model to avoid the trivial solution. Section IV provides the experiments on another large-scale dataset, and the major parameters analysis of RMGE is also discussed.

The rest of this article is organized as follows. Section II reviews the graph-based SSL and AG-based HSIC methods. In Section III, the specific content of the proposed RMGE is given. Section IV reports the extensive experimental results and analysis. Finally, Section V concludes this article.

II. Related Work

A. Graph-Based SSL HSIC Methods

Various graph-based SSL models have obtained success in HSIC, where the sparse representation (SR) of data and the corresponding manifold structure are considered.

1) Graph-Based SSL With Manifold Learning: Many variants of manifold learning explore the nonlinear manifold structure hidden in the high-dimensional HSI data. Ma *et al.* [36] constructed the graph by adopting the manifold learning strategy, which captures the local geometric properties of each neighborhood and improves the classification accuracy of data with multiple submanifolds. Cheng *et al.* [37] proposed the SSL framework that joints discriminant information and adaptive loss, which improves the separability between different objects by utilizing the pairwise constraints, alleviating the impact of outliers. To make better use of spatial information, Luo *et al.* [38] built adjacent graph with distance metric and local manifold structure to design the pairwise similarity, preserving the local properties of spectral–spatial neighbors. Jamshidpour *et al.* [39] constructed different graphs and fused their Laplacians to build the adjacent graph, achieving the better capability of extracting the spatial features.

2) Graph-Based SSL With SR: The SR-based graphs can obtain the adjacent relationships among pixels and weights simultaneously and automatically, which reinforce the discriminability of model naturally. Therefore, many various SR-based graph models have been proposed. Morsier *et al.* [40] encoded the representation of graph with sparse coding and low rank constraints for HSIC. This method obtains the graph that has limited connectivity and provides the smoother kernel distortion. Shao *et al.* [41] presented a graph by merging probabilistic class structure into the SR-based edge weighting model, which describes the edge weights between pixels and classes, enhancing the discriminability of graph structure. Dornaika *et al.* [42] proposed graph-SSL with LBP for holistic object categorization, which designs a two-phase regularized least square to obtain graph in a semisupervised context and utilizes LBP as descriptors to classify the detected objects in outdoor and indoor scenes. Cheng *et al.* [43] developed a spectral–spatial random patches network, which integrates shallow, deep, spectral, and spatial feature with LBP, and stacks them into high-dimensional vector. Then, the high-dimensional spectral–spatial vector is fed into a graph-based learning for HSIC. Xue *et al.* [44] fused the spatial information into SR model by total variation. Based on sparse graph representation, it propagated the predicted labels from dictionary to unknown data, which is free-parameters and robust to noise. Aydemir and Bilgin [45] designed a SSL-based method for spectral–spatial HSIC, which provides rich samples by using subtractive clustering-based model for training. By combining SVM and kernel SR classifiers, this method obtained better and more promising results. Hong *et al.* [28] proposed an end-to-end fusion framework, which develops the fusion schemes by integrating extracted features and the proposed graph convolutional networks. It learns more discriminative feature representations for HSIC.

Although graph-based SSL methods for classification have shown better accuracy, this is at the cost of higher computational complexity, which is inefficient in large-scale data. In addition,

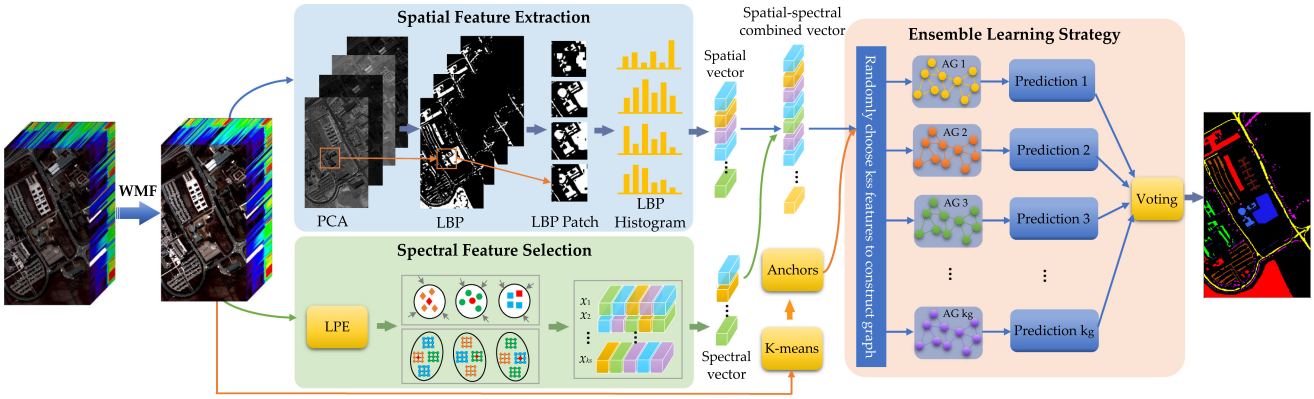


Fig. 1. Architecture of an RMGE algorithm.

the high dimensionality and limited labeled data of HSI further restrict the performance of classification.

B. AG-Based HSIC Methods

AG-based methods can address the scalability issue with a small subset of samples as anchors, which is enough to represent the entire points. Motivated by recent development in the AG construction, He *et al.* [22] proposed fast SSL model with AG, which constructs a naturally sparse and scale invariant AG, alleviating the computation burden. Wang *et al.* [21] developed a scalable AG-based clustering method, which adds the nonnegative relaxation to AG model. With this, the clustering results are directly obtained without adopting k-means. To handle the defect that AG-based methods usually ignore the spatial information, Wang *et al.* [47] presented an AG-based method with clustering for HSI data, which fuses the spatial information by using the mean of neighboring pixels to reconstruct center pixel. Wei *et al.* [48] utilized the spatial correlation by adopting WMF to filter hyperspectral pixels, which considers the local neighborhood relationship within a window. He *et al.* [23] presented a semisupervised model with bipartite graph, which calculates the labels of the data and anchors simultaneously and adopts the Woodbury matrix to handle the large matrix inverse. It obtains better accuracy in the shorter time and requires less storage spaces. Gao *et al.* [34] improved a spatial–spectral classification framework with multigraphs, which randomly selects several features to create the graph, mitigating the well-known Hughes phenomenon. Zhou *et al.* [30] proposed a local regularized embedding model with spatial–spectral information, which incorporates spatial feature into the dimensionality reduction procedure, to achieve optimal discriminative matrix by the minimization of local spatial–spectral scatter. Feng *et al.* [31] designed the discriminate margins with spatial–spectral neighborhood pixels, which effectively captures the discriminative features and learns the structures of HSI data.

III. Methodology

The details of the proposed RMGE algorithm are described from five aspects: WMF, spatial–spectral feature extraction, anchor-based adjacent graph construction, graph-based SSL framework, and AG-based SSL model. The architecture of RMGE is shown in Fig. 1.

A. Weighted Mean Filtering

To smooth the homogeneous regions and reduce the interference of noisy points, the WMF [30] is employed to preprocess the original HSI data $\mathbf{Y}$. The main idea of the WMF is to use the weighted mean of the neighboring pixels in the spatial window to reconstruct the center pixel value, where the weight coefficient of the center pixel and the neighboring pixels is determined by their spectral similarity. The higher the similarity is, the greater the weight coefficient is. On the contrary, the lower the similarity (such as the similarity between the noise point and the center pixel) is, the smaller the weight coefficient is. Therefore, although there are many filter techniques, such as edge-preserving filtering and anisotropic diffusion filtering, the WMF is relatively simple, fast, and efficient for HSI preprocessing.

Assume $\mathbf{Y} = [y_1, y_2, ..., y_n]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$ and each pixel is $y_i \in \mathbb{R}^d$ $(i = 1, 2, ..., n)$. The labels of $\{y_1, y_2, ..., y_n\}$ are denoted as $\mathbf{T}(y_i) \in \{1, 2, ..., c\}$, respectively, and c represents the number of categories. Suppose that $(\bar{p}_i, \bar{g}_i)$ denotes the corresponding spatial coordinate of y_i, then the neighborhood of y_i is denoted as

$$\Omega(y_i) = \left\{ y_i(\bar{p}, \bar{g}) \middle| \begin{matrix} \bar{p} \in [\bar{p}_i - \delta, \bar{p}_i + \delta] \\ \bar{g} \in [\bar{g}_i - \delta, \bar{g}_i + \delta] \end{matrix} \right\} \tag{1}$$

where $\delta = (s-1)/2$, s indicates the size of neighborhood window and is a positive odd number.

The reconstructed pixel $\hat{y}_i$ by WMF is defined with a weighted summation, i.e.,

$$\hat{y}_i = \frac{y_i + \sum_{k=1}^{s^2-1} v_k y_{ik}}{1 + \sum_{k=1}^{s^2-1} v_k}, \quad y_{ik} \in \Omega(y_i) \tag{2}$$

where $v_k = \exp\{-\gamma_0 \|y_i - y_{ik}\|_2^2\}$ represents the similarity between y_{ik} and y_i on the spectrum. The parameter γ_0 is empirically set to be 0.2, which reflects the degree of filtering. $\Omega(y_i)$ is the adjacent space of y_i. The WMF method can effectively decrease the influence of noise and enhance the spectral similarity between pixels from the same class by adjusting the

degree of filtering. After WMF, the consistency of pixels in the same homogeneous regions is guaranteed.

B. Spatial–Spectral Feature Extraction

1) Spatial Feature Extraction: To improve computational efficiency, the principal component analysis is applied to reduce the dimensionality of HSI preprocessed by WMF. The LBP is then adopted in each principal component for extracting texture features to refine the spatial information. The original LBP model can effectively extract rotation-invariant texture features, such as edges, corners, and spots, to describe local spatial patterns. Those texture features are learned by comparing each pixel with its local neighborhood. Specifically, LBP utilizes the center pixel value to label the pixels of a local region with a binary threshold, which does not make any assumptions about the distribution of this local region. We use the uniform pattern of LBP in this article, which reduces the feature vector and implements rotation-invariant operator.

By LBP processing, the local spatial feature of y_i is defined as

$$\mathrm{LBP}(y_i) = \sum_{h=1}^{s_l^2-1} \mathcal{L}_{\mathrm{LBP}}(y_{ih} - y_i)2^h, y_{ih} \in \Omega(y_i) \tag{3}$$

where s_l indicates the size of neighborhood window and $\mathcal{L}_{\mathrm{LBP}}(y_{ih} - y_i) = 0$ if $y_{ih} \leq y_i$, otherwise $\mathcal{L}_{\mathrm{LBP}}(y_{ih} - y_i) = 1$. As indicated in (3), a binary code is assigned to each neighbor of y_i, which reveals the texture orientation and smoothness in the adjacent space. Based on the LBP code, the corresponding histogram is obtained over the local patch centered at y_i. Then, the spatial feature is built by concatenating all the principal components of LBP histograms, which captures local structures efficiently. LBP is a nonparametric method and computed simply, and the special property of LBP is the tolerance about monotonic illumination changes, which is suitable for single band processing in HSI.

2) Spectral Feature Selection: The linear prediction error [49] is employed to select spectral bands. Given two initial bands $\mathbf{B}_1$ and $\mathbf{B}_2$, the other $\mathbf{B}$ that is the least similar to $\mathbf{B}_1$ and $\mathbf{B}_2$ can be found by the two steps: a) using $\mathbf{B}_1$ and $\mathbf{B}_2$ to predict $\mathbf{B}$ by $\mathbf{B}' = \alpha_0 + \alpha_1\mathbf{B}_1 + \alpha_2\mathbf{B}_2$; and b) learning the parameters $(\alpha_0, \alpha_1, \alpha_2)$ to minimize $\varepsilon = \|\mathbf{B} - \mathbf{B}'\|$. Assume that $\mathbf{a} = [\alpha_0, \alpha_1, \alpha_2]^{\mathrm{T}}$, and it is obtained by employing the least squares solution, i.e.,

$$\mathbf{a} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}q \tag{4}$$

where $\mathbf{M} = [1, \mathbf{b}_1, \mathbf{b}_2] \in \mathbb{R}^{n\times 3}$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the column vectors that represent all the n pixels in $\mathbf{B}_1$-band and $\mathbf{B}_2$-band, respectively. Similarly, q is the all pixels in $\mathbf{B}$-band. The band that maximizes the error $\varepsilon_{\min}$ is selected as $\mathbf{B}$. Thus, it is obvious that this similar procedure is simple to conduct when the desired number of bands is greater than two. Through abovementioned steps, the band subset that can be used to represent other bands is obtained as the spectral features.

Based on the abovementioned steps, we obtain the spatial–spectral features of a pixel by stacking the spatial features and spectral features into 1-D vector.

C. Adjacent AG Construction

Graph-based SSL models include two steps:
1) constructing the adjacent graph;
2) using the graph to predict the unknown labels.

Generally, constructing the adjacent graph has several methods, such as kNN, ϵ-neighborhood, and heat kernel. This section mainly describes the method of constructing the adjacent graph that we design.

We randomly select k_{ss} features from the spatial–spectral features obtained in the Section III-B to build a graph. To decrease the computational complexity, we use the anchor-based strategy to build the adjacent graph. Ideally, the linear combination of the anchors can be utilized to represent each sample. Furthermore, the data label is also considered as a specific representation of the sample. The defined prediction function of labels is

$$f(x_i) = \sum_{j=1}^{m} w_{ij} f(u_j) \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{m\times d}$ means the anchor set and m represents the number of anchors ($m \ll n$). $\mathbf{W} \in \mathbb{R}^{n\times m}$ denotes the affinity matrix, where w_{ij} is the similarity between y_i and u_j. Assuming that $\mathbf{F} \in \mathbb{R}^{n\times c}$ and $\mathbf{F}_u \in \mathbb{R}^{m\times c}$, (5) is expressed as $\mathbf{F} = \mathbf{W}\mathbf{F}_u$. It can be seen that the design of $\mathbf{W}$ is an important part, which contains the following two stages.

1) *Anchors generation:* There are usually two methods of anchor generation: random selection and k-means method. Random selection generates anchors by selecting samples from data points randomly, which cannot guarantee that generated anchors are always effective. The k-means method utilizes the clustering centers as anchors, which can obtain more representative anchors. Therefore, the k-means method is adopted here. The k-means method is a simple and easily implemented clustering algorithm. The details are described as follows: the method selects the initialized m samples as the initial cluster centers, and it proceeds by alternating following two steps until converge: a) assignment: each sample is assigned to the cluster with the closest distance; and b) update: recalculate the new means to become the centroid of the samples in the cluster. When the assignment is no longer changed, the algorithm is considered to have converged.
2) *Adjacent graph learning:* The matrix $\mathbf{W}$ is usually constructed by kNN approach. Motivated by [17] and [50], the nearest anchors assignment of y_i is considered as

$$\min_{w_i^{\mathrm{T}}\mathbf{1}=1, 0\leq w_{ij}\leq 1, \|w_i\|_0=k} \sum_{j=1}^{m} \|y_i - u_j\|_2^2 w_{ij} + \sum_{j=1}^{m} \gamma w_{ij} \ln w_{ij} \tag{6}$$

where w_i^{T} is the ith row vector of matrix $\mathbf{W}$.

The second term is the maximum entropy regularization, which controls the uniformity of the values of w_{ij} and avoids the trivial solution of (6). γ is the parameter of regularization term. Suppose that $e_{ij} = \|y_i - u_j\|_2^2$, the solution of problem

(6) is represented as

$$\min_{w_i^{\mathrm{T}}\mathbf{1}=1,0\le w_{ij}\le 1,\|w_i\|_0=k} \sum_{j=1}^{m} w_{ij}e_{ij} + \gamma \sum_{j=1}^{m} w_{ij}\ln w_{ij}. \quad (7)$$

It is obtained that the vector w_i only has k nonzero values based on ℓ_0-norm constraint. Assume that the index of k nonzero values in w_i^T is $\{j_1, j_2, ..., j_k\}$, then the problem (7) is reduced to

$$\min_{w_i^{\mathrm{T}}\mathbf{1}=1,0\le w_{ij}\le 1} \sum_{r=1}^{k} w_{ij_r}e_{ij_r} + \gamma \sum_{r=1}^{k} w_{ij_r}\ln w_{ij_r}. \quad (8)$$

The Lagrangian function of (8) is

$$\mathcal{L}(w_i,\gamma) = \sum_{r=1}^{k} w_{ij_r}e_{ij_r} + \gamma \sum_{r=1}^{k} w_{ij_r}\ln w_{ij_r} + \lambda(w_i^{\mathrm{T}}\mathbf{1}-1) \quad (9)$$

where λ is the Lagrangian multiplier. In order to achieve the optimal w_{ij_r}, it should satisfy that the derivative of (9) with respect to w_{ij_r} is equal to zero, i.e.,

$$w_{ij_r} = \exp\left(\frac{-e_{ij_r}}{\gamma}\right)\cdot \exp\left(\frac{\lambda_{ij_r}-\gamma}{\gamma}\right). \quad (10)$$

Considering $w_i^{\mathrm{T}}\mathbf{1}=1$ and (10), we obtain

$$\exp\left(\frac{\lambda_{ij_r}-\gamma}{\gamma}\right) = \frac{1}{\sum_{r=1}^{k}\exp(-e_{ij_r}/\gamma)}. \quad (11)$$

Then, according to (10) and (11), we obtain

$$w_{ij_r} = \frac{\exp(-e_{ij_r}/\gamma)}{\sum_{r=1}^{k}\exp(-e_{ij_r}/\gamma)}. \quad (12)$$

Subsequently, substituting (12) into (7) and minimizing the corresponding objective function. Then, the optimal solution for problem (6) is

$$w_{ij} = \begin{cases} \frac{\exp(-\|y_i-u_{j_r}\|_2^2/\gamma)}{\sum_{r=1}^{k}\exp(-\|y_i-u_{j_r}\|_2^2/\gamma)}, & j\in\{r_1,r_2,...,r_k\} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where $\{r_1, r_2, ..., r_k\}$ are the indexes of k nearest neighbors of pixel y_i. Then, we design the normalized adjacent graph $\mathbf{Z}$ by using the nonnegative matrix $\mathbf{W}$, and it is defined as $\mathbf{Z} = \mathbf{W}\mathbf{\Lambda}^{-1}\mathbf{W}^{\mathrm{T}}$. The jth element in diagonal matrix $\mathbf{\Lambda}\in\mathbb{R}^{m\times m}$ is expressed as $\Lambda_{jj} = \sum_{i=1}^{n} w_{ij}$. Intuitively, the z_{ij} is represented as $z_{ij} = w_i^{\mathrm{T}}\Lambda^{-1}w_j$ and $z_{ij} = z_{ji}$. Furthermore, the required $\mathbf{Z}\ge 0$ ensure that the resulting graph Laplacian is positive semidefinite. This property of $\mathbf{Z}$ is crucial to obtain global optimum of AG-based SSL model, which will be considered later.

D. Graph-Based SSL Framework

As mentioned above, the HSI data is represented as $\mathbf{Y}\in\mathbb{R}^{n\times d}$. Suppose $\mathbf{Y} = \mathbf{Y}_l \cup \mathbf{Y}_v (n = l+v)$, $\mathbf{Y}_l = \{y_1, y_2, ..., y_l\}$ denotes the labeled pixels, and $\mathbf{Y}_v = \{y_{l+1}, y_{l+2}, ..., y_{l+v}\}$ is the unlabeled pixels. Given the adjacent graph $\mathbf{Z}$, the graph-based SSL approaches are regarded as solving the following function:

$$\min \mathrm{Tr}((\mathbf{F}-\mathbf{T})^{\mathrm{T}}\mathbf{X}(\mathbf{F}-\mathbf{T})) + \mathrm{Tr}(\mathbf{F}^{\mathrm{T}}\mathbf{L}\mathbf{F}) \quad (14)$$

where $\mathbf{F}$ denotes the predicted values on labeled pixels. $\mathbf{T}$ is a label matrix and $\mathbf{T} = [t_1, ..., t_l, 0, ..., 0]^{\mathrm{T}} \in \mathbb{R}^{n\times c}$. $\mathbf{X}\in\mathbb{R}^{n\times n}$ is a diagonal matrix, where $x_i = \beta_l$ if $i\le l$ and $x_i = \beta_v$ if $l+1\le i\le n$. β_l and β_v are empirically set to $\beta_l = 0.01$ and $\beta_v = 10^{-6}$, respectively. $\mathbf{L}$ represents a Laplacian matrix defined as $\mathbf{L} = \mathbf{D}-\mathbf{Z}$, where $\mathbf{D}$ is the degree matrix.

E. AG-based SSL Model

As mentioned in the Section III-C, $\mathbf{F} = \mathbf{W}\mathbf{F}_u$. The label propagation for the AG-based SSL model can be represented as

$$\mathcal{L}(\mathbf{F}_u) = \mathrm{Tr}\left((\mathbf{F}_l-\mathbf{T}_l)^{\mathrm{T}}(\mathbf{F}_l-\mathbf{T}_l)\right) + \eta\mathrm{Tr}(\mathbf{F}^{\mathrm{T}}\mathbf{L}\mathbf{F}) \quad (15)$$

where $\mathbf{F}_l = \mathbf{W}_l\mathbf{F}_u$, $\mathbf{T}_l$ is the labels of labeled pixels, and η is the parameter of regularization term. Equation (15) is transformed into the following form:

$$\begin{aligned}\mathcal{L}(\mathbf{F}_u) &= \|\mathbf{W}_l\mathbf{F}_u-\mathbf{T}_l\|_F^2 + \eta\mathrm{Tr}((\mathbf{W}\mathbf{F}_u)^{\mathrm{T}}\mathbf{L}(\mathbf{W}\mathbf{F}_\mathbf{u})) \\ &= \|\mathbf{W}_l\mathbf{F}_u-\mathbf{T}_l\|_F^2 + \eta\mathrm{Tr}({\mathbf{F}_u}^{\mathrm{T}}\mathbf{L}_A\mathbf{F}_u). \end{aligned} \quad (16)$$

Let us define $\mathbf{L}_A = \mathbf{W}^{\mathrm{T}}\mathbf{L}\mathbf{W}$ where $\mathbf{L} = \mathbf{D}-\mathbf{Z}$. The ith element of $\mathbf{D}$ is

$$d_{ii} = \sum_{j=1}^{n} z_{ij} = \sum_{j=1}^{n} w_i^{\mathrm{T}}\Lambda^{-1}w_j = w_i^{\mathrm{T}}\sum_{j=1}^{n}\Lambda^{-1}w_j = w_i^{\mathrm{T}}\mathbf{1} = 1. \quad (17)$$

Therefore, $\mathbf{L} = \mathbf{D}-\mathbf{Z} = \mathbf{I}-\mathbf{W}\mathbf{\Lambda}^{-1}\mathbf{W}^{\mathrm{T}}$, we have

$$\begin{aligned}\mathbf{L}_A &= \mathbf{W}^{\mathrm{T}}(\mathbf{I}-\mathbf{W}\mathbf{\Lambda}^{-1}\mathbf{W}^{\mathrm{T}})\mathbf{W} \\ &= \mathbf{W}^{\mathrm{T}}\mathbf{W} - (\mathbf{W}^{\mathrm{T}}\mathbf{W})\mathbf{\Lambda}^{-1}(\mathbf{W}^{\mathrm{T}}\mathbf{W}). \end{aligned} \quad (18)$$

Taking the derivative of (16) with regard to $\mathbf{F}_u$, we have

$$\mathbf{F}_u^* = (\mathbf{W}_l^{\mathrm{T}}\mathbf{W}_l + \eta\mathbf{L}_A)^{-1}\mathbf{W}_l^{\mathrm{T}}\mathbf{T}_l. \quad (19)$$

According to $\mathbf{F}_u^*$, we calculate the predicted label of y_i directly as follows:

$$t_i = \arg\max_{j\in\{1,...,c\}} \mathbf{W}_{i\cdot}\mathbf{F}_{u_j}^*. \quad (20)$$

The whole framework of ensemble strategy in this article is described as follows.

1) k_{ss} features are selected randomly from all the high-dimensional features of each pixel.
2) Graph construction: m anchors that covered the data manifold are generated, and then the weighted matrix $\mathbf{W}$ is calculated to represent the rest of pixels by the selected anchors.
3) The semisupervised inference on this graph is conducted by employing the graph Laplacian regularization.
4) The abovementioned steps are repeated to obtain k_g graphs in parallel.
5) The unknown labels of pixels are acquired by processing the prediction results of k_g graphs using majority voting rules. The solution process of RMGE is summarized in Algorithm 1.

Algorithm 1: RMGE.

Input: Hyperspectral data $\mathbf{Y}$, window size s_l, anchor number m, the number of selected features k_{ss}, graph number k_g, $\beta_l = 0.01$, $\beta_u = 10^{-6}$.

Use WMF to preprocess the HSI data $\mathbf{Y}$;
Use LPE to select spectral features y_{s1};
Utilize PCA to complete the dimensionality reduction of HSI data, then adopt the LBP extractor to obtain spatial features y_{s2};
Stack y_{s1} and y_{s2} into 1-D vector y_{ss};
repeat
  Randomly choose k_{ss} features from y_{ss};
  Employ k-means to generate m anchors, obtaining the anchor set $\mathbf{U} \in \mathbb{R}^{m \times d}$;
  Obtain the adjacent graph $\mathbf{W}$ by (13);
  Design the normalized adjacent graph $\mathbf{Z} = \mathbf{W}\mathbf{\Lambda}^{-1}\mathbf{W}^{\mathrm{T}}$;
  Calculate $\mathbf{F}_u^*$ by (19), and get the label of each pixel by (20);
until the construction of k_g graphs is finished;
Obtain the final label t_i by the results from k_g graphs with majority voting;

Output: The label matrix $\mathbf{T}$.

IV. Experiments

We verify the effectiveness of the proposed RMGE on public hyperspectral datasets in this section. To have a fair comparison, we make all competitors adopt their respective optimal parameters. The results of experiments and parameter analysis are also discussed.

A. Datasets

The false color images and ground truths of three hyperspectral datasets are shown in Fig. 2. The distribution of training samples and test samples of each dataset is given in the Tables I and II.

1) Indian Pines: The hyperspectral image is acquired via AVIRIS device, as shown in Fig. 2(a). It covers the northwestern Indiana with 145 × 145 pixels and 200 bands. The Indian Pines contain 16 land-cover types whose distribution is shown in Fig. 2(b).

2) Pavia University (PaviaU): The dataset is captured by the German ROSIS sensor as shown in Fig. 2(c). It covers the Pavia and contains 610 × 340 pixels. The 103 bands are selected with discarding 12 bands. The PaviaU contains nine categories, as shown in Fig. 2(d).

3) Salinas: The HSI data is also taken by AVIRIS, as shown in Fig. 2(e). It covers the Salinas Valley, 1998, having 512 × 217 pixels. The 204 bands are left after excluding 20 bands. All pixels in dataset are divided into 16 categories, as shown in Fig. 2(f).

B. Experimental Setup

1) Comparison Algorithms: Some HSIC methods are compared with the proposed method.

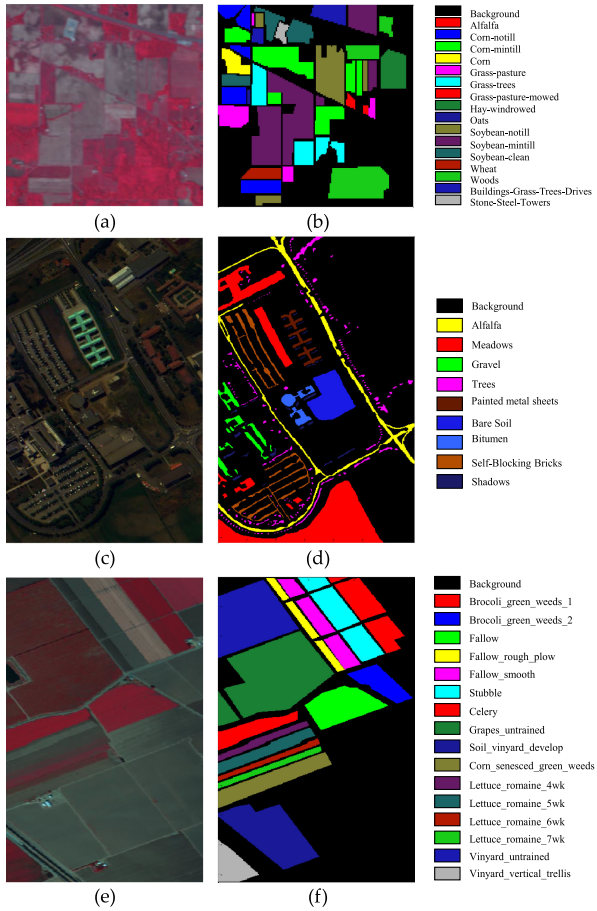


Fig. 2. RGB false color images and ground truth maps of three HSI datasets.

R-VCANet [11]: This method combines the spatial structures and spectral features by utilizing rolling guidance filter and vertex component analysis networks, which extracts deep and representative feature expression for HSI data.

GCN-CNN [28]: This method proposes an end-to-end fusion framework, which develops the fusion schemes by integrating features extracted from CNNs and the proposed minibatch graph convolutional networks. It learns more discriminative feature representations for HSIC.

LSLRR [12]: It exploits the similarity metric that joints spectral–spatial features to capture local structure. Besides, the locality constraint criterion and structure-preserving strategy are introduced to low-rank representation to construct the graph.

SS-RMG [34]: This method explores the structure with multiple graphs using random spectral–spatial information, where the addition of randomness alleviates the Hughes phenomenon.

SSHGDA [29]: The complex spatial–spectral characteristics are learned by the proposed hypergraph, which uses the priori knowledge fully and extracts the multiple intrinsic properties of HSI.

FSSLAG [22]: This method presents the AG-based SSL model, which is parameter-free and naturally sparse in the adjacent graph construction.

GLLR-SKSVM [27]: Based on low-rank representation, the graph-based SSL algorithm is designed to consider the similarity of mixed pixels in spectral–spatial space.

TABLE I
TRAIN/TEST DISTRIBUTION OF SAMPLES FOR INDIAN AND SALINAS

No.	Indian Pines			No.	Salinas		
	Class	Train	Test		Class	Train	Test
1	Alfalfa	3	43	**1**	Brocoli_green_weeds_1	20	1989
2	Corn-notill	72	1356	**2**	Brocoli_green_weeds_2	37	3689
3	Corn-mintill	42	788	**3**	Fallow	19	1957
4	Corn	12	225	**4**	Fallow_rough_plow	14	1380
5	Grass-pasture	24	459	**5**	Fallow_smooth	27	2741
6	Grass-trees	37	693	**6**	Stubble	40	3919
7	Grass-pasture-mowed	2	26	**7**	Celery	35	3544
8	Hay-windrowed	24	454	**8**	Grapes_untrained	112	11159
9	Oats	2	18	**9**	Soil_vinyard_develop	62	6141
10	Soybean-notill	49	1023	**10**	Corn_senesced_green_weeds	32	3246
11	Soybean-mintill	120	2335	**11**	Lettuce_romaine_4wk	10	1058
12	Soybean-clean	30	563	**12**	Lettuce_romaine_5wk	19	1908
13	Wheat	10	195	**13**	Lettuce_romaine_6wk	9	907
14	Woods	64	1201	**14**	Lettuce_romaine_7wk	10	1060
15	Buildings-Grass-Trees-Drives	20	366	**15**	Vinyard_untrained	72	7196
16	Stone-Steel-Towers	5	88	**16**	Vinyard_vertical_trellis	18	1789
	Total	**516**	**9833**		**Total**	**536**	**53593**

TABLE II
TRAIN/TEST DISTRIBUTION OF SAMPLES FOR PAVIAU

No.	PaviaU		
	Class	Train	Test
1	Asphalt	66	6565
2	Meadows	186	18463
3	Gravel	21	2078
4	Trees	31	3033
5	Painted metal sheets	13	1332
6	Bare Soil	50	4979
7	Bitumen	13	1317
8	Self-Blocking Bricks	37	3645
9	Shadows	9	938
	Total	**426**	**42350**

2) Evaluation Indices: Three quantitative criteria are taken in this article. The first is overall accuracy (OA) that represents the proportion of correctly classified pixels in HSI. The second is average accuracy (AA), which denotes the average of the classification accuracy of each category. The value of OA and AA ranges from 0 to 1. The third is Kappa coefficient that is a metric combined commission error and omission error, which can evaluate the overall consistency. Its value range is [0,1]. The larger Kappa values indicate better consistency.

3) Parameters Analysis: The training samples sizes of Indian Pines, PaviaU, and Salinas are 5%, 1%, and 1% respectively, and the rest of pixels in the every dataset are selected for testing. To reduce parameters adjustment, we set the anchors count equal to the number of training samples. We here mainly discuss the three parameters involved in the proposed RMGE. The first parameter k_g denotes the number of graphs, which is a crucial parameter in our method. The quantitative results of different k_g on three hyperspectral datasets as given in the Table III. Fig. 3(a) is provided to show the changing trend of OAs with varying k_g on three datasets. It is worth noting that there is an increase on OA, AA, and Kappa when $k_g \leq 4$ for three datasets. When $k_g > 4$, the accuracy tends to be stable on the Indian Pines and Salinas, while it has a sharp decline on the PaviaU dataset. For the experiments, it can be seen that the final results are not always better with the increase of k_g. Since each graph is constructed by randomly choosing k_{ss} features, the corresponding classifier is different from each other. The greater the difference between each other is, the more diversity the classifier learns, and the final result is relatively better. From the Table III, although the better results are obtained by setting $k_g = 9$ for the datasets of Indian and Salinas, the time taken is exponentially increased for the slight improvement. Therefore, to ensure good accuracy and speed up the running rate of RMGE, we choose $k_g = 4$ for three datasets.

The number of bands selected in the presented RMGE is also close to the performance of HSIC. We set the number of bands from {1, 4, 5, 6, 7, 8, 9, 10, 15, 20} to explore the changes of accuracy on the three datasets, as given in Table IV. Fig. 3(b) is provided to show the changing trend of OAs with varying bands count on three datasets. It reveals that if the number of bands is below 5, the values of OA, AA, and Kappa present a slow decline on Indian dataset. Similarly, if the number of bands is above 4, the accuracy has a sharp drop. However, according to the Fig. 3(b), the fluctuation range of accuracy is small on Salinas dataset the number of bands is above 4. It is relatively stable. Hence, the number of bands is set to 4 in our experiments. For the accuracy, although it is not the best excellent choice for all the datasets, we select the relatively number to prepare for a better analysis within the allowable hardware resources.

As given in Table V, the window size s_l of LBP module is another parameter that is related to the final classification results. We choose $s_l = \{1, 3, 7, 10, 15, 20, 30\}$ for observing the influence on the three datasets. It can be noticed that there is a larger increase on OA, AA, and Kappa when s_l ranges from 1 to 7. Then, the values of accuracy present a steady tendency when $s_l > 7$ on three datasets. Because the extracted features by the smaller window size may not represent the spatial characteristics of the center pixel. Therefore, we set $s_l = 7$ in the subsequent experiments. Meanwhile, to reflect the influence of parameters

TABLE III
QUANTITATIVE RESULTS OF DIFFERENT GRAPH NUMBERS k_g ON THREE DATASETS

k_g	Indian				PaviaU				Salinas			
	OA	AA	Kappa	Time(s)	OA	AA	Kappa	Time(s)	OA	AA	Kappa	Time(s)
1	0.9633	0.9530	0.9582	16.6	0.9027	0.8357	0.8682	113.9	0.9860	0.9786	0.9844	187.6
2	0.9736	0.9504	0.9699	18.7	0.9318	0.8988	0.9077	138.7	0.9919	0.9825	0.9910	238.9
3	0.9803	0.9794	0.9776	22.8	0.9308	0.8922	0.9059	170.1	0.9917	0.9827	0.9920	295.1
4	**0.9834**	**0.9820**	**0.9810**	**24.9**	**0.9872**	**0.9877**	**0.9830**	**174.5**	**0.9925**	**0.9820**	**0.9916**	**358.8**
5	0.9798	0.9798	0.9770	35.8	0.9384	0.9027	0.9166	252.6	0.9904	0.9763	0.9763	446.1
6	0.9731	0.9722	0.9693	41.5	0.9307	0.9023	0.9058	318.8	0.9939	0.9835	0.9932	527.6
7	0.9782	0.9542	0.9751	48.4	0.9281	0.8918	0.9023	541.3	0.9943	0.9889	0.9936	622.5
8	0.9802	0.9496	0.9775	53.3	0.9397	0.9009	0.9182	572.4	0.9952	0.9929	0.9947	738.3
9	**0.9843**	**0.9864**	**0.9821**	64.2	0.9395	0.9053	0.9181	708.6	**0.9958**	**0.9895**	**0.9954**	801.2
10	0.9770	0.9718	0.9739	68.2	0.9410	0.9070	0.9200	768.4	0.9935	0.9815	0.9928	910.9

The optimal value of each column is highlighted in bold.

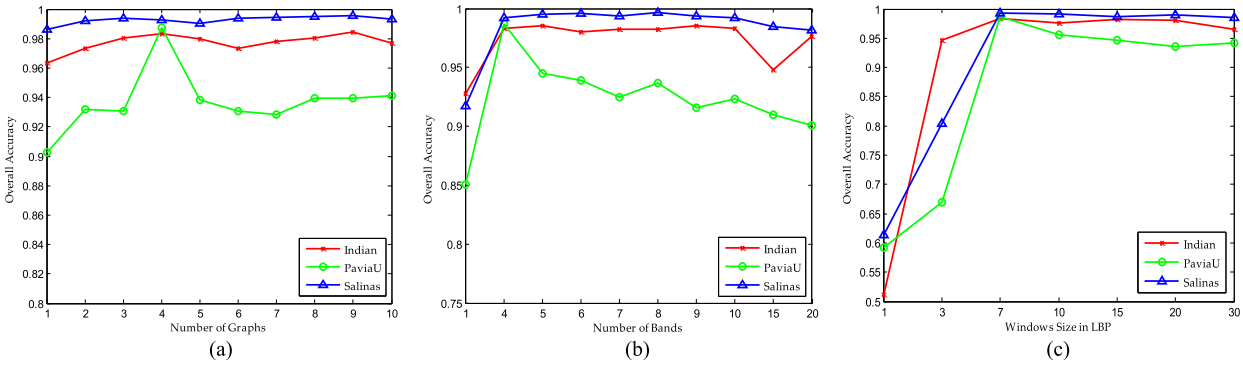


Fig. 3. Influence of different parameters on three datasets. (a) Number of graphs (k_g). (b) Number of bands. (c) Windows size in LBP (s_l).

TABLE IV
QUANTITATIVE RESULTS FOR ANALYZING THE IMPACT OF NUMBERS OF BANDS ON THREE DATASETS

Number of Bands	Indian				PaviaU				Salinas			
	OA	AA	Kappa	Time(s)	OA	AA	Kappa	Time(s)	OA	AA	Kappa	Time(s)
1	0.9274	0.9200	0.9173	35.6	0.8508	0.7803	0.7928	420.5	0.9171	0.9145	0.9072	464.3
4	**0.9834**	**0.9820**	**0.9810**	**24.9**	**0.9872**	**0.9877**	**0.9830**	**174.5**	**0.9925**	**0.9820**	**0.9916**	**358.8**
5	**0.9857**	0.9718	**0.9837**	30.2	0.9451	0.8973	0.9259	287.5	0.9955	0.9924	0.9950	368.7
6	0.9799	0.9792	0.9771	41.9	0.9387	0.8953	0.9172	369.6	0.9957	0.9899	0.9952	377.9
7	0.9826	0.9741	0.9802	38.2	0.9250	0.8752	0.8981	437.9	0.9937	0.9831	0.9929	388.6
8	0.9824	0.9651	0.9800	42.1	0.9371	0.8880	0.9147	453.2	**0.9964**	**0.9922**	**0.9960**	437.9
9	0.9855	0.9765	0.9835	35.0	0.9157	0.8233	0.8849	452.0	0.9936	0.9857	0.9929	430.3
10	0.9835	0.9787	0.9812	37.6	0.9232	0.8687	0.8960	458.3	0.9924	0.9819	0.9916	454.6
15	0.9483	0.9420	0.9412	42.1	0.9099	0.8087	0.8767	424.6	0.9849	0.9596	0.9832	585.9
20	0.9762	0.9474	0.9728	55.3	0.9007	0.7908	0.8634	737.3	0.9819	0.9583	0.9798	724.7

The optimal value of each column is highlighted in bold.

clearly, Fig. 3(c) is provided to show the changing trend of OAs with varying s_l on three datasets. Fig. 4 is provided to show the influence of multiparameter combinations for classification accuracy on the Indian dataset. According to Fig. 4, the fluctuation range of OAs is relatively small. We can see that the OAs of the proposed RMGE are relatively stable with changing the parameters.

As shown in Fig. 5, it indicates the OAs of three HSI datasets with varying k_{ss}. In the ensemble learning strategy, k_{ss} features are chosen randomly to construct graph, which can also affect the final classification results of RMGE. We set k_{ss} from {20, 50, 80, 100, 120, 140, 160, 180, 200} to analyze its influence on Indian and Salinas, and {10, 20, 30, 40, 50, 60, 70, 80, 90, 100} on PaviaU. It is worth noting that there is an increase on OA when $k_{ss} < 80$ for Indian and Salinas datasets. When $k_{ss} > 80$, the accuracy tends to be stable. For the PaviaU dataset, the OAs increase when $k_{ss} < 40$, and the fluctuation range of accuracy is small for $k_{ss} > 40$. It is relatively stable. Therefore, in this article, we set $k_{ss} = 150$ on Indian and Salinas and $k_{ss} = 80$ on PaviaU, respectively, in term of Fig. 5.

TABLE V
QUANTITATIVE RESULTS FOR ANALYZING THE IMPACT OF DIFFERENT WINDOW SIZES s_l ON THREE DATASETS

s_l	Indian				PaviaU				Salinas			
	OA	AA	Kappa	Time(s)	OA	AA	Kappa	Time(s)	OA	AA	Kappa	Time(s)
1	0.5114	0.5485	0.4421	44.6	0.5921	0.3844	0.3572	381.2	0.6139	0.5044	0.5596	423.2
3	0.9473	0.9378	0.9609	30.3	0.6700	0.5180	0.4868	598.07	0.8045	0.7457	0.7787	453.0
7	**0.9834**	**0.9820**	**0.9810**	**24.9**	**0.9872**	**0.9877**	**0.9830**	**174.5**	**0.9925**	**0.9820**	**0.9916**	**358.8**
10	0.9757	0.9731	0.9723	31.4	0.9562	0.9252	0.9416	553	0.9913	0.9748	0.9903	386.9
15	0.9818	0.9742	0.9792	31.8	0.9463	0.9056	0.9285	625	0.9861	0.9590	0.9845	397.9
20	0.9812	0.9745	0.9786	32.3	0.9350	0.8991	0.9137	643.9	0.9893	0.9893	0.9881	435.7
30	0.9657	0.9697	0.9609	37.8	0.9425	0.9006	0.9235	371.5	0.9859	0.9831	0.9854	479.4

The optimal value of each column is highlighted in bold.

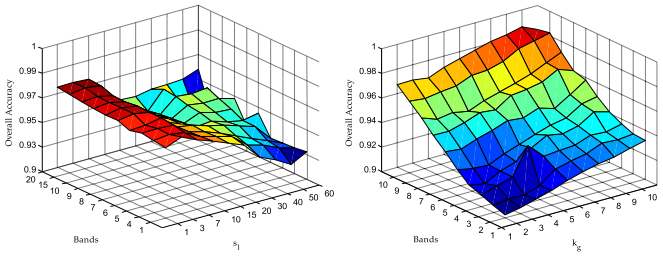


Fig. 4. Influence of multiparameters on Indian datasets.

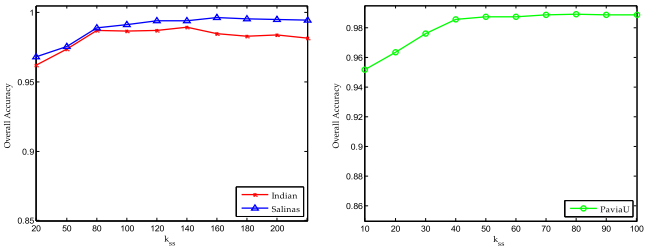


Fig. 5. Influence of k_{ss} on three datasets.

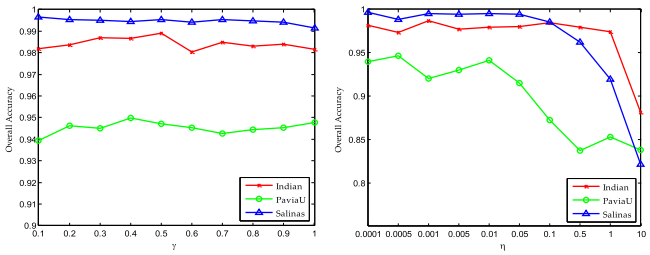


Fig. 6. Influence of γ and η on three datasets.

Fig. 6 shows the OAs of three HSI datasets with varying γ and η, respectively. γ is the maximum entropy regularization parameter in RMGE. From Fig. 6, note that, the value of classification accuracy remains unchanged basically when γ turns from [0.1:0.1:1]. This phenomenon indicates that the performance of RMGE is insensitive to parameter γ. In other words, the experimental results are not stochastic in this article, which is authentic and reliable. η is a regularization parameter, and the result demonstrates that the optimal value of η is 0.001, 0.01, and 0.001 for Indian Pines, PaviaU, and Salinas datasets, respectively.

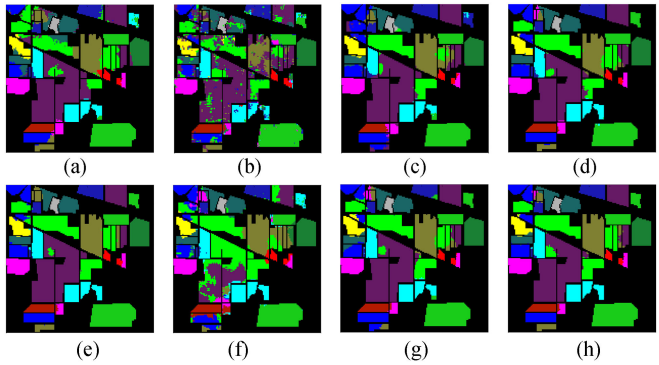


Fig. 7. Visualization results of different methods on Indian.

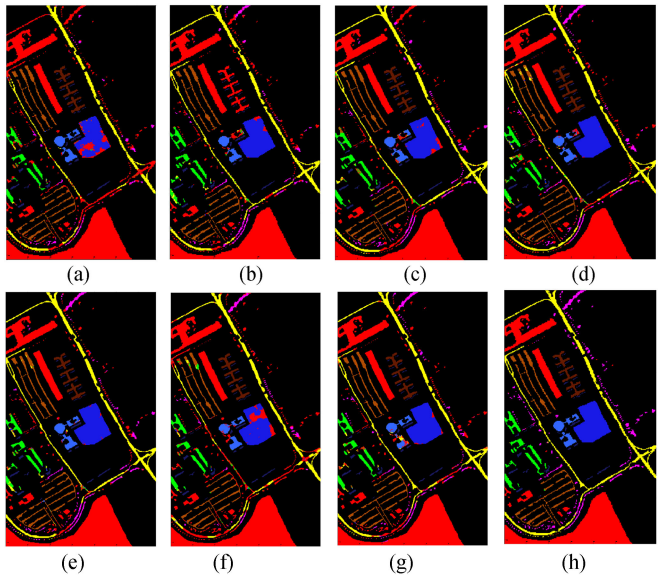


Fig. 8. Visualization results of different methods on PaviaU.

C. Comparison of Classification Performance

The experiments and parameter analysis of the proposed approach are discussed in this part. All the results about the proposed method are obtained by averaging the accuracy results of 30 times.

1) Performance on Indian Pines Dataset: 5% labeled pixels of each land cover class are randomly chosen for training

TABLE VI
QUANTITATIVE RESULTS OF COMPARISON ALGORITHMS AND OUR RMGE ON INDIAN DATASET

Class	R-VCANet	GCN-CNN	LSLRR	SS-RMG	SSHGDA	FSSLAG	GLLR-SKSVM	RMGE
Alfalfa	**1.0000**	0.9565	0.9815	0.9815	0.9814	0.9998	0.9630	0.9837±0.0025
Corn-notill	0.7085	0.6026	0.7120	0.9324	0.9595	0.8096	0.9386	**0.9838±0.0014**
Corn-mintill	0.6007	0.7218	0.9329	0.9724	0.9508	0.7158	0.8885	**0.9679±0.0117**
Corn	0.8760	0.9209	**0.9915**	0.9829	0.9529	0.8504	0.9444	0.9448±0.0124
Grass-pasture	0.9517	0.9196	0.9175	**0.9839**	0.9014	0.8330	0.9537	0.9595±0.0004
Grass-trees	0.9477	0.9859	0.8929	0.9612	**0.9959**	0.9813	0.9652	0.9801±0.0027
Grass-pasture-mowed	1.0000	0.9714	0.9231	1.0000	0.9615	0.9231	1.0000	**1.0000±0.0000**
Hay-windrowed	0.9468	0.9986	0.9959	1.0000	1.0000	1.0000	1.0000	**1.0000±0.0000**
Oats	1.0000	1.0000	0.9000	1.0000	0.7500	1.0000	0.9000	**1.0000±0.0000**
Soybean-notill	0.8626	0.7095	0.8285	0.8781	0.9907	0.8306	0.8461	**0.9906±0.0048**
Soybean-mintill	0.9712	0.7374	0.9899	0.9895	**0.9983**	0.4607	0.9846	0.9921±0.0063
Soybean-clean	0.7768	0.7772	0.9104	**0.9870**	0.9478	0.9495	0.9446	0.8790±0.0412
Wheat	0.9811	0.9987	0.9009	1.0000	**1.0000**	1.0000	0.9953	0.9927±0.0032
Woods	0.9327	0.8629	0.9080	0.9961	0.9992	0.8918	1.0000	**1.0000±0.0000**
Buildings-Grass-Trees-Drives	0.8894	0.8310	0.9553	0.9947	0.9973	0.9632	0.9395	**0.9975±0.0006**
Stone-Steel-Towers	0.9578	**0.9954**	0.9789	0.9684	0.9473	0.9895	0.9789	0.9816±0.0109
OA	0.8700	0.7825	0.9031	0.9687	0.9786	0.7757	0.9523	**0.9824±0.0021**
AA	0.9002	0.8735	0.9199	0.9768	0.9584	0.8874	0.9527	**0.9819±0.0039**
Kappa	0.8513	0.7519	0.8896	0.9644	0.9756	0.7494	0.9457	**0.9816±0.0017**

The optimal value of each row is highlighted in bold.

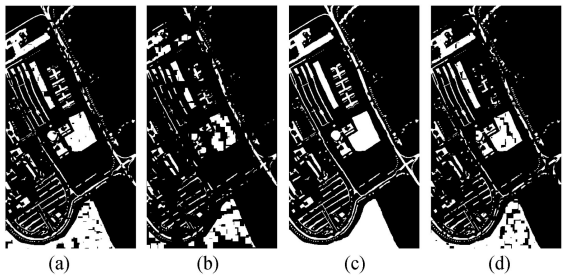


Fig. 9. Spatial features extracted by LBP on different bands.

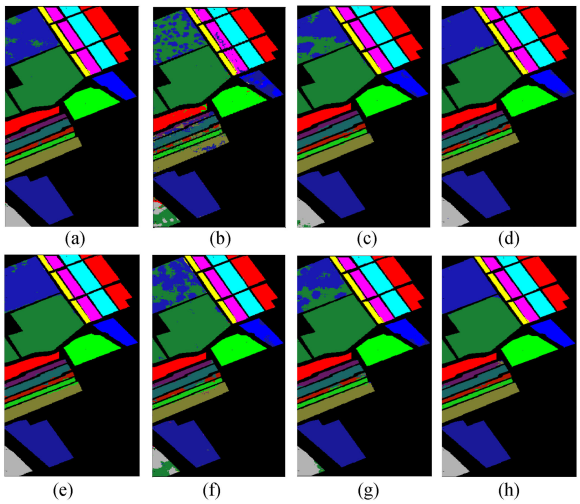


Fig. 10. Visualization results of different methods on Salinas.

model. The quantitative results of the approaches are given in Table VI. It can be revealed that RMGE obtains better results than competitors on OA, AA, and Kappa. Moreover, RMGE gains the higher accuracy in most classes, as shown Table VI, which indicates that the ensemble framework is constructed by using multiple AGs can learn the diversity of HSI data well. From Table VI, the graph-based SSL methods (SS-RMG, GLLR-SKSVM, and RMGE) outperform the traditional graph-based methods (LSLRR). Among these competitors, the proposed method achieve the best accuracy, especially for several surface objects (e.g, Corn-notill, Corn-mintill, and Soybean-notill). This illustrates that the designed AG-based SSL model captures the effective relationships in the adjacent graph learning. Furthermore, compared with FSSLAG, the proposed RMGE improves over 20% in terms of OA. It shows that the maximum entropy regularization in the graph construction procedure takes the better performance.

The final classification maps obtained with each algorithm are also shown in Fig. 7. The proposed RMGE obtains much smoother maps than other methods. Especially, RMGE has a superior performance than other spatial–spectral combined methods, which indicates that LBP-based spatial features that apply in the graph learning significantly provides an accurate description of HSI local areas. It improves the discriminability of the proposed model.

2) Performance on PaviaU Dataset: 1% labeled pixels of each land cover class are randomly chosen for training model, and the rest of pixels are selected for testing. To observe the experimental results visually and quantitatively, Fig. 8 and Table VII display the visual classification maps and the corresponding accuracy, respectively. As given in Table VII, the presented approach gives superior results than other methods, especially for the land-cover types (Asphalt, Gravel, Trees, and Bitumen), whose accuracy achieve 0.9547 ± 0.0051, 1, 0.9446 ± 0.0325, and 1. After incorporating the LBP-based spatial characteristics into the AG model, RMGE yields better accuracy in almost all classes compared with FSSLAG. This indicates that LBP module obtains more representative spatial features existing in local regions, which refines the local structures of HSI. By employing the LBP extractor, the distinctive characteristics in spatial are captured, as shown in Fig. 9. The module discovers the subtle differences of different bands, and preserves the intrinsic properties of spatial structure. By this way, the discriminability of classifier is enhanced, and the performance of RMGE is improved effectively. As is shown in Fig. 8, RMGE gains

TABLE VII
QUANTITATIVE RESULTS OF COMPARISON ALGORITHMS AND OUR RMGE ON PAVIAU DATASET

Class	R-VCANet	GCN-CNN	LSLRR	SS-RMG	SSHGDA	FSSLAG	GLLR-SKSVM	RMGE
Asphalt	0.7133	0.8132	0.8758	0.9096	0.9215	0.7133	0.8839	**0.9547±0.0051**
Meadows	0.9990	0.9490	**1.0000**	0.9998	0.9998	0.9999	0.8090	0.9939±0.0027
Gravel	0.8380	0.8553	0.8604	0.7979	0.8018	0.7861	0.5089	**1.0000±0.0000**
Trees	0.2007	0.9475	0.3769	0.4624	0.4944	0.2725	0.7313	**0.9446±0.0325**
Painted metal sheets	0.9992	0.9988	0.9992	0.9985	0.9977	0.9985	0.9548	**1.0000±0.0000**
Bare Soil	0.8841	0.8962	0.9365	**0.9968**	0.9858	0.8240	0.4007	0.9990±0.0005
Bitumen	0.7804	0.9042	0.8211	0.8804	0.9548	0.8744	0.5774	**1.0000±0.0000**
Self-Blocking Bricks	0.9885	0.7895	0.9698	0.9462	0.9693	0.8769	0.6648	**0.9929±0.0025**
Shadows	0.9493	**1.0000**	0.9556	0.9672	0.9514	0.9778	**1.0000**	0.9833±0.0059
OA	0.8673	0.9046	0.9126	0.9280	0.9350	0.8572	0.9179	**0.9868±0.0017**
AA	0.8169	0.9060	0.8662	0.8844	0.8974	0.8137	0.8668	**0.9854±0.0026**
Kappa	0.8148	0.8741	0.8805	0.9021	0.9120	0.7995	0.8877	**0.9820±0.0015**

The optimal value of each row is highlighted in bold.

TABLE VIII
QUANTITATIVE RESULTS OF COMPARISON ALGORITHMS AND OUR RMGE ON SALINAS DATASET

Class	R-VCANet	GCN-CNN	LSLRR	SS-RMG	SSHGDA	FSSLAG	GLLR-SKSVM	RMGE
Brocoli_green_weeds_1	1.0000	0.9920	1.0000	1.0000	1.0000	1.0000	1.0000	**1.0000±0.0000**
Brocoli_green_weeds_2	0.9994	0.9995	0.9994	0.9994	**0.9997**	0.9986	0.9994	0.9991±0.0004
Fallow	0.9696	0.9738	0.9149	0.8243	1.0000	0.8785	0.8836	**1.0000±0.0000**
Fallow_rough_plow	0.9956	0.9951	0.9791	0.9713	0.9763	0.9942	0.9813	**0.9946±0.0022**
Fallow_smooth	0.9895	0.9855	**0.9958**	0.9932	0.9936	0.9936	0.9869	0.9813±0.0115
Stubble	0.9992	0.9998	0.9987	0.9982	0.9992	0.9984	0.9987	**0.9993±0.0005**
Celery	**0.9987**	0.9981	0.9986	0.9983	0.9977	0.9958	0.9977	0.9973±0.0021
Grapes_untrained	**0.9998**	0.6684	0.9979	0.9991	0.9958	0.9871	0.9996	0.9906±0.0057
Soil_vinyard_develop	1.0000	0.9985	1.0000	1.0000	1.0000	0.9967	1.0000	**1.0000±0.0000**
Corn_senesced_green_weeds	0.9832	0.9371	0.9835	0.9704	0.9450	0.9456	0.9792	**0.9964±0.0026**
Lettuce_romaine_4wk	0.9260	0.9841	0.9213	0.8717	**0.9616**	0.9353	0.9466	0.8697±0.0121
Lettuce_romaine_5wk	0.9880	0.9994	0.9766	0.9896	0.9730	0.9891	**0.9901**	0.9873±0.0046
Lettuce_romaine_6wk	**0.9443**	0.9988	0.8657	0.6037	0.9312	0.6135	0.6855	0.9028±0.0107
Lettuce_romaine_7wk	0.9841	0.9837	0.9448	0.8962	**0.9906**	0.9775	0.9700	0.9618±0.0074
Vinyard_untrained	0.8999	0.7382	0.6512	0.9775	0.9741	0.6055	0.6487	**0.9979±0.0007**
Vinyard_vertical_trellis	0.9950	0.9794	0.9335	0.9994	0.9994	0.5130	0.9263	**1.0000±0.0000**
OA	0.9802	0.9072	0.9397	0.9755	0.9881	0.9104	0.9361	**0.9913±0.0018**
AA	0.9795	0.9520	0.9476	0.9433	**0.9836**	0.9014	0.9371	0.9814±0.0024
Kappa	0.9780	0.9051	0.9325	0.9727	0.9867	0.8995	0.9284	**0.9899±0.0021**

The optimal value of each row is highlighted in bold.

more smoother map than other approaches, where the visualized results accord with the Table VII.

3) Performance on Salinas Dataset: 1% labeled pixels of each land cover class are randomly chosen for training model, and the rest of pixels are selected for testing. To visualize the experimental results, Fig. 10 shows the classification maps. The quantitative accuracy of comparison algorithms and RMGE are given in Table VIII. According to the visual results, it reveals that the pixels in the Grapes_untrained class and Vinyard_untrained class regions are most wrongly assigned. That is because the classes of Vinyard_untrained and Grapes_untrained have highly similar feature information, and their spectral curves are very close. From Fig. 10, the Vinyard_untrained class is discriminated well, and its accuracy reaches to 0.9979±0.0007 in Table VIII, which is higher than that of other methods. This phenomenon also justifies the effectiveness of RMGE. As given in Table VIII, the results illustrate that RMGE yields the best classification accuracies, especially the OA and Kappa coefficient are up to 0.9913 ± 0.0018 and 0.9899 ± 0.0021, respectively.

D. Computational Time Comparison

We display the running time of all the algorithms, as shown in Table IX. All the experimental results are implemented with MATLAB R2014a on a PC of Intel Core i7-9700F 3.00GHz CPU with 16 GB RAM. Table IX gives the time consumed by different algorithms on hyperspectral datasets. As given in Table IX, R-VCANet, GCN-CNN, and SSHGDA produce larger execution time on different datasets, while the other methods have less processing time. Among these methods, the time consuming of FSSLAG and RMGE are the least, which indicates the AG-based SSL methods can process HSI data more efficiently than graph-based methods (i.e., LSLRR, SS-RMG, SSHGDA, and GLLR-SKSVM). Obviously, the proposed RMGE obtains best results about running time. It illustrates that the runtime efficiency can be ensured by constructing multiple AGs in parallel. To sum up, the RMGE algorithm proposed in this article not only executes faster, but also outperforms other algorithms in classification performance on three hyperspectral datasets.

TABLE IX
TIME CONSUMED BY DIFFERENT ALGORITHMS ON HYPERSPECTRAL DATASETS

Datasets	R-VCANet	GCN-CNN	LSLRR	SS-RMG	SSHGDA	FSSLAG	GLLR-SKSVM	RMGE
Indian Pines	1378.5s	2684.8s	336.2s	63.9s	1029.7s	37.46s	311s	**24.9s**
Pavia University	2789.7s	6898.6s	1257.3s	380.9s	13938.1s	254.7s	1498.4s	**174.5s**
Salinas	6274.9s	15759.3s	5729.7s	446.1s	20382.9s	386.4s	5894.5s	**358.8s**

The optimal value of each row is highlighted in bold.

V. CONCLUSION

In this article, an efficient graph-based SSL classification model with spatial–spectral features was proposed for HSI, which was based on random multigraphs and ensemble strategy. In order to discover the local structures and subtle changes within a region, the fine spatial features captured by LBP module were integrated into the AG learning, which contains more discriminant information. In addition, the multigraphs were constructed by selecting the spatial–spectral features randomly, to alleviate the high dimensionality of HSI data. The maximum entropy regularization was introduced in the AG construction to avoid the trivial solution. The label representation was adopted in the prediction function to capture the intrinsic properties of data. Moreover, the ensemble framework makes AGs to be built in parallel, which learns the diversity of HSI data and improves the efficiency of the proposed RMGE. Experiments conducted on HSI datasets show the satisfying performance of RMGE, and verify its advantages over the excellent competitors.

REFERENCES

[1] Q. Zhu *et al.*, "A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification," *IEEE Trans. Cybern.*, early access, May 25, 2021, doi: 10.1109/TCYB.2021.3070577.

[2] Q. Wang, Q. Li, and X. Li, "A fast neighborhood grouping method for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5028–5039, Jun. 2021.

[3] P. Sellars, A. I. Aviles-Rivero, and C.-B. Schönlieb, "Superpixel contracted graph-based learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4180–4193, Jun. 2020.

[4] C. Deng, X. Liu, C. Li, and D. Tao, "Active multi-kernel domain adaptation for hyperspectral image classification," *Pattern Recognit.*, vol. 77, pp. 306–315, 2018.

[5] S. Susan, S. Shinde, and S. Batra, "Vegetation-specific hyperspectral band selection for binary-to-multiclass classification," in *Proc. IEEE 16th India Council Int. Conf.*, 2019, pp. 1–4.

[6] F. Gan, S. Liang, P. Du, F. Dang, K. Tan, H. Su, and Z. Xue, "CHESRE: A comprehensive public hyperspectral experimental site and data set for resources exploration," in *Proc. 7th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens.*, 2015, pp. 1–4.

[7] Y. Wan, X. Hu, Y. Zhong, A. Ma, L. Wei, and L. Zhang, "Tailings reservoir disaster and environmental monitoring using the UAV-ground hyperspectral joint observation and processing: A case of study in Xinjiang, the belt and road," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 9713–9716.

[8] Q. Wang, Z. Yuan, Q. Du, and X. Li, "GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2019.

[9] G. Camps-Valls, N. Shervashidze, and K. M. Borgwardt, "Spatio-spectral remote sensing image classification with graph kernels," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 741–745, Oct. 2010.

[10] J. Peng, Y. Zhou, and C. L. P. Chen, "Region-kernel-based support vector machines for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4810–4824, Sep. 2015.

[11] 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.*, 2015, pp. 4959–4962.

[12] Q. Wang, X. He, and X. Li, "Locality and structure regularized low rank representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 911–923, Feb. 2019.

[13] X. He, Y. Chen, and P. Ghamisi, "Dual graph convolutional network for hyperspectral image classification with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5502418, doi: 10.1109/TGRS.2021.3061088.

[14] J. Feng *et al.*, "Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 7, 2020, Art. no. 1149.

[15] H. Wang, Y. Cheng, C. L. P. Chen, and X. Wang, "Semisupervised classification of hyperspectral image based on graph convolutional broad network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2995–3005, Mar. 2021.

[16] J. Feng *et al.*, "Deep reinforcement learning for semisupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art no. 5501719, doi: 10.1109/TGRS.2021.3049372.

[17] F. Nie, S. Shi, and X. Li, "Semi-supervised learning with auto-weighting feature and adaptive graph," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1167–1178, Jun. 2020.

[18] X. Li, M. Chen, F. Nie, and Q. Wang, "A multiview-based parameter free framework for group detection," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4147–4153.

[19] A. Anis, A. El Gamal, A. S. Avestimehr, and A. Ortega, "A sampling theory perspective of graph-based semi-supervised learning," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2322–2342, Apr. 2019.

[20] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 679–686.

[21] R. Wang, F. Nie, Z. Wang, F. He, and X. Li, "Scalable graph-based clustering with nonnegative relaxation for large hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7352–7364, Oct. 2019.

[22] F. He, R. Wang, and W. Jia, "Fast semi-supervised learning with anchor graph for large hyperspectral images," *Pattern Recognit. Lett.*, vol. 130, pp. 319–326, 2020.

[23] F. He, F. Nie, R. Wang, X. Li, and W. Jia, "Fast semisupervised learning with bipartite graph for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 626–638, Feb. 2020.

[24] W. Liao, M. D. Mura, J. Chanussot, and A. Piurica, "Fusion of spectral and spatial information for classification of hyperspectral remote-sensed imagery by local graph," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 583–594, Feb. 2016.

[25] H. Huang, G. Shi, H. He, Y. Duan, and F. Luo, "Dimensionality reduction of hyperspectral imagery based on spatial-spectral manifold learning," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2604–2616, Jun. 2020.

[26] H. Huang, Y. Duan, H. He, and G. Shi, "Local linear spatial spectral probabilistic distribution for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1259–1272, Feb. 2020.

[27] Z. Feng, S. Yang, M. Wang, and L. Jiao, "Learning dual geometric low-rank structure for semisupervised hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 346–358, Jan. 2021.

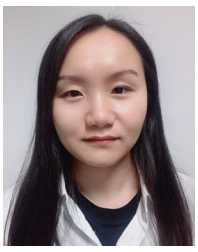
[28] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[29] F. Luo, B. Du, L. Zhang, L. Zhang, and D. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2406–2419, Jul. 2019.

[30] Y. Zhou, J. Peng, and C. P. Chen, "Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1082–1095, Feb. 2015.

[31] Z. Feng, S. Yang, S. Wang, and L. Jiao, "Discriminative spectral-spatial margin-based semisupervised dimensionality reduction of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 224–228, Feb. 2015.

[32] E. Zhang, X. Zhang, L. Jiao, L. Li, and B. Hou, "Spectral spatial hyperspectral image ensemble classification via joint sparse representation," *Pattern Recognit.*, vol. 59, pp. 42–54, 2016.
[33] Y. Zhang, G. Cao, A. Shafique, and P. Fu, "Label propagation ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3623–3636, Sep. 2019.
[34] F. Gao, Q. Wang, Y. Dong, and Q. Xu, "Spectral and spatial classification of hyperspectral images based on random multi-graphs," *Remote Sens.*, vol. 10, no. 8, 2018, Art. no. 1271.
[35] Y.Miao, Q.Wang, M.Chen, and X.Li, "Spatial-spectral hyperspectral image classification via multiple random anchor graphs ensemble learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 3641–3644.
[36] L. Ma, M. M. Crawford, X. Yang, and Y. Guo, "Local-manifold-learning-based graph construction for semisupervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2832–2844, May 2015.
[37] G. Cheng, F. Zhu, S. Xiang, Y. Wang, and C. Pan, "Semisupervised hyperspectral image classification via discriminant analysis and robust regression," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 595–608, Feb. 2016.
[38] R. Luo, W. Liao, X. Huang, Y. Pi, and W. Philips, "Feature extraction of hyperspectral images with semisupervised graph learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4389–4399, Sep. 2016.
[39] N. Jamshidpour, A. Safari, and S. Homayouni, "Spectral spatial semisupervised hyperspectral classification using adaptive neighborhood," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4183–4197, Sep. 2017.
[40] F. de Morsier, M. Borgeaud, V. Gass, J.-P. Thiran, and D. Tuia, "Kernel low-rank and sparse graph for unsupervised and semi-supervised classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3410–3420, Jun. 2016.
[41] Y. Shao, N. Sang, C. Gao, and L. Ma, "Probabilistic class structure regularized sparse representation graph for semi-supervised hyperspectral image classification," *Pattern Recognit.*, vol. 63, pp. 102–114, 2017.
[42] F. Dornaika, A. Bosaghzadeh, H. Salmane, and Y. Ruichek, "Graph-based semi-supervised learning with local binary patterns for holistic object categorization," *Expert Syst. Appl.*, vol. 41, no. 17, pp. 7744–7753, 2014.
[43] C. Cheng, H. Li, J. Peng, W. Cui, and L. Zhang, "Hyperspectral image classification via spectral-spatial random patches network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4753–4764, Apr. 2021.
[44] Z. Xue, P. Du, J. Li, and H. Su, "Sparse graph regularization for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 2351–2366, Apr. 2017.
[45] M. S. Aydemir and G. Bilgin, "Semisupervised hyperspectral image classification using small sample sizes," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 621–625, May 2017.
[46] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.
[47] Y. Wei, C. Niu, Y. Wang, H. Wang, and D. Liu, "The fast spectral clustering based on spatial information for large scale hyperspectral image," *IEEE Access*, vol. 7, pp. 141045–141054, 2019.
[48] Q. Du and H. Yang, "Similarity-based unsupervised band selection for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 564–568, Oct. 2008.
[49] Z. Wang, F. Nie, R. Wang, H. Yang, and X. Li, "Local structured feature learning with dynamic maximum entropy graph," *Pattern Recognit.*, vol. 111, 2021, Art. no. 107673.



Yanling Miao received the B.E. degree in communication engineering and the M.S. degree in computer application technology from Henan Polytechnic University, Jiaozuo, China, in 2015 and 2019, respectively. She is currently working toward the Ph.D. degree in computer science and technology with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an, China.

Her research interests include hyperspectral image processing and computer vision.

Mulin Chen received the B.E. degree in software engineering and the Ph.D. degree in computer application technology from Northwestern Polytechnical University, Xi'an, China, in 2014 and 2019, respectively.

He is currently a Researcher with the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University. His research interests include computer vision and machine learning.

Yuan Yuan (Senior Member, IEEE) is currently a Full Professor with the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an, China. She has authored or coauthored more than 150 papers, including about 100 in reputable journals, such as the IEEE TRANSACTIONS AND PATTERN RECOGNITION, as well as the conference papers in CVPR, BMVC, ICIP, and ICASSP. Her research interests include visual information processing and image/video content analysis.

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. Since 2013, he has been an Adjunct Professor of the University of Iceland. In 2015–2017, he was a visiting professor at the University of California, Los Angeles (UCLA). He holds the AXA chair in remote sensing and is an Adjunct professor at the Chinese Academy of Sciences, Aerospace Information research Institute, Beijing. His research interests include image analysis, hyperspectral remote sensing, data fusion, machine learning and artificial intelligence. He has been a visiting scholar at Stanford University (USA), KTH (Sweden) and NUS (Singapore).

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

Qi Wang (Senior Member, IEEE) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively.

He is currently a Professor with the School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, machine learning, pattern recognition, and remote sensing.