

Semisupervised Classification of Hyperspectral Image Based on Graph Convolutional Broad Network

Haoyu Wang, Yuhu Cheng , *Member, IEEE*, C. L. Philip Chen , *Fellow, IEEE*,
and Xuesong Wang , *Member, IEEE*

Abstract—Hyperspectral image (HSI) classification has attracted much attention in the field of remote sensing. However, the lack of sufficient labeled training samples is a huge challenge for HSI classification. To face this challenge, we propose a semisupervised HSI classification method based on graph convolutional broad network (GCBN). First, to avoid the underfitting problem caused by the insufficient linear sparse feature representation ability of broad learning system (BLS), graph convolution operation is applied to extract nonlinear and discriminative spectral-spatial features from the original HSI to replace the linear mapping features in the traditional BLS. Second, to solve the problem of insufficient model classification ability caused by limited labeled samples, the combinatorial average method (CAM) is proposed to use valuable paired samples to generate sample expansion set for GCBN model training. Third, BLS is used to perform broad expansion on spectral-spatial features extracted by GCN and extended by CAM, which further enhances the feature representation ability. Finally, the output weights can be easily calculated by the ridge regression theory. Experimental results on three real HSI datasets demonstrate the effectiveness of our proposed GCBN.

Index Terms—Broad learning, classification, hyperspectral image (HSI), sample expansion, semisupervised learning.

I. INTRODUCTION

HYPERSPECTRAL images (HSI) contain rich spectral and spatial information, which makes them widely used in crop monitoring, environmental monitoring, mineral exploration, and other fields [1]–[4]. HSI classification is one of the basic and key technologies of remote sensing for earth surface observation. It aims to infer the class of each pixel based on the spectral and spatial information of the HSI [5]–[7]. The early staged methods for HSI classification are mostly based

on conventional pattern recognition methods, such as K-nearest neighbor [8] and support vector machine (SVM) [9], random forest [10], and decision tree [11]. In addition, extreme learning machines [12], sparse representation [13], and graph embedding methods [14] are also used for HSI classification. However, most early staged HSI classification methods only focused on exploring the role of the spectral information for classification, and therefore high classification accuracy could not be obtained [15]. Since the neighboring pixels in HSI usually carry rich spatial information, many spectral-spatial classification methods have been proposed and the spatial information of HSI was used to obtain higher classification accuracy therein. For instance, some researchers applied the spatial information to HSI classification via the extended morphological profiles, and thus the satisfactory classification accuracy could be achieved [16], [17]. Spectral and spatial information contained in the neighborhood region of the pixels were merged and added into the sparse representation model in [18] and [19]. Tu *et al.* [20] proposed a spectral-spatial HSI classification method, which exploited the comprehensive contextual information of HSI by considering a weak assumption that the pixels in a superpixel belong to the same class, and achieved an excellent classification performance. Sellami *et al.* [21] proposed an HSI classification approach, which made full use of the spectral-spatial information by automatically selecting relevant spectral bands.

Compared with traditional machine learning algorithms, deep learning techniques can automatically extract high-level and compact features from input data. In recent years, deep learning techniques have been successfully applied to HSI classification tasks. Chen *et al.* [22] used stacked autoencoders to extract the features of HSI, and entered them into the logistic regression model for classification. Liu *et al.* [23] first used deep belief network to extract deep spectral features, and then repeatedly selected good-quality labeled samples as training samples with active learning algorithms. Zhang *et al.* [24] proposed an HSI classification algorithm based on the convolutional neural network (CNN), which utilized diverse region-based inputs to learn discriminative spectral-spatial features. Chen *et al.* [25] used 1-D, 2-D, and 3-D CNN to extract features of HSI, respectively. Kong *et al.* [26] extracted the spectral features of HSI by constructing intra-class and inter-class hypergraphs, and extracted spatial features by CNN. Zhu *et al.* [27] adopted generative adversarial networks to construct a semisupervised feature learning framework for HSI classification. Mou *et al.* [28] applied recurrent neural network (RNN) to HSI classification for the

Manuscript received December 30, 2020; revised February 17, 2021; accepted February 22, 2021. Date of publication March 2, 2021; date of current version March 18, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61976215, Grant 61772532, Grant 61751202, and Grant U1813203. (Corresponding author: Xuesong Wang.)

Haoyu Wang, Yuhu Cheng, and Xuesong Wang are with the Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China, with the Xuzhou Key Laboratory of Artificial Intelligence and Big Data, China University of Mining and Technology, Xuzhou 221116, China, and also with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China (e-mail: 2540854800@qq.com; chengyuhu@163.com; wangxuesongcumt@163.com).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: Philip.Chen@ieee.org).

Digital Object Identifier 10.1109/JSTARS.2021.3062642

first time, and proposed a parameter modified tanh activation function to replace the traditional activation function.

The impressive feature representation capability of deep learning is based on abundant labeled samples. However, collecting the labeled HSI data is difficult and expensive [29]. Therefore, how to learn a strong generalization classifier at a low labeling cost has become a research hotspot in the field of HSI analysis. To address this concern, many methods have been proposed, which contain four categories. The first one is data augmentation, which synthesizes new examples following the original data distribution [30]. Li *et al.* [31] constructed a new training set for CNN by using pairwise labeled samples and exploited it to improve the model classification accuracy. Wang *et al.* [32] established a data mixture model to augment the labeled training set quadratically, and exploited this set to train the CNN. The second category is named domain adaptation, which uses sufficient samples from different but similar domains to solve the problems for another domain [33]. Zhou and Prasad [34] first used deep convolutional RNNs to extract the discriminative features for two domains, then aligned the features with each other layer-by-layer in the common subspaces, and thus realized the HSI classification of different distributions by exploiting only part of labeled samples in the source domain. The third one is active learning, which can exploit a small number of labeled samples to train a classifier, making the classifier actively select representative unlabeled samples [35]. The semisupervised method utilizes abundant unlabeled data and limited labeled samples for classification. Wu and Prasad [36] proposed a semisupervised deep learning network, which effectively alleviated the shortage of labeled samples by combining limited labeled samples with abundant unlabeled samples for HSI classification.

Broad learning system (BLS) is a random vector functional link neural network (RVFLNN) consisting of only three parts [mapped feature (MF), enhancement node (EN), and output layer] [37]. Compared with the deep learning, BLS has the following advantages [37]: 1) BLS can nonlinearly expand the feature. 2) BLS has a simple and flexible structure with only three layers. 3) Gradient descending is used in deep learning methods, which requires more times of iterations. For BLS, the ridge regression is exploited to directly calculate the network weights of BLS, so the network training speed is fast. 4) It is easy to integrate BLS with other models. Feng and Chen [38] proposed a fuzzy BLS by combining the Takagi–Sugeno fuzzy system with BLS, which achieved an ideal accuracy in regression and classification. Chu *et al.* [39] proposed a weighted BLS, in which the contribution of each input sample to the BLS was constrained by exploiting penalty factors. Kong *et al.* [40] proposed a semisupervised model by merging the class-probability structure into BLS and achieved good classification performance in HSI classification. Kong *et al.* [41] proposed a HSI clustering algorithm based on BLS, and exploited the graph-regularized sparse autoencoder to fine-tune the weights of MF and EN.

As the latest research achievement of deep learning, the graph convolutional network (GCN) can aggregate and transform the neighbor feature information from each node. Besides, GCN is

able to encode features of graph nodes and local graph structure by convolutional layers, so as to exhaustively exploit the graph features and flexibly preserve the class boundaries [42]. However, the original GCN only utilizes the spectral information when classifying HSI, that is, only constructs the spectral adjacency matrix. Qin *et al.* [43] consider the data structure characteristic of HSI and the advantages of GCN, and completed the HSI classification by constructing a spectral-spatial adjacency matrix while using spectral and spatial information. Therefore, according to [43], we first use GCN to extract the spectral-spatial features of HSI. Then, CAM is used to expand the data with spectral-spatial features extracted by GCN, while considering the flexible network structure and the ability of feature broad expansion of BLS, a semisupervised graph convolutional broad network (GCBN) is proposed. The main contributions of our work are summarized as follows.

- 1) We replace the linear mapping features used in the traditional BLS with the spectral-spatial features extracted from the original HSI by GCN, which can achieve accurate HSI classification at low labeling cost by means of exploiting limited labeled samples and abundant unlabeled samples.
- 2) In the proposed combinatorial average method (CAM), some valuable paired samples are selected in a targeted manner, and averaged in pairs to generate a sample expansion set much larger than the original training set. Thus, the problem of the lack of labeled samples to support high-precision classification model training can be solved.
- 3) We exploit the BLS to perform broad expansion on spectral-spatial features extracted by GCN and extended by CAM, which is helpful to further enhance the representation ability of features and thus improve the classification accuracy of HSI.

The rest of this article is organized as follows. We elaborate the semisupervised classification method of HSI based on GCBN in Section II. We present experimental results on three real HSI datasets and analyze them in Section III-A followed by a conclusion in Section IV.

II. SEMISUPERVISED CLASSIFICATION OF HSI BASED ON GCBN

A. Flowchart of GCBN for HSI Classification

The flowchart of the proposed GCBN for HSI classification is shown in Fig. 1, which mainly contains the following five steps:

- 1) The principal component analysis (PCA) is applied to the original HSI to reduce dimensionality;
- 2) The spectral-spatial graph of GCBN constructed based on the spectral and spatial information of limited labeled samples and abundant unlabeled samples is used for graph convolution operation. Then the discriminative spectral-spatial features of HSI are extracted by the trained GCN;
- 3) In our proposed CAM, some valuable paired samples are selected in a targeted manner, and averaged in pairs to generate a sample expansion set for GCBN training;
- 4) BLS is used to expand the width of spectral-spatial features extracted by GCN and extended by CAM;

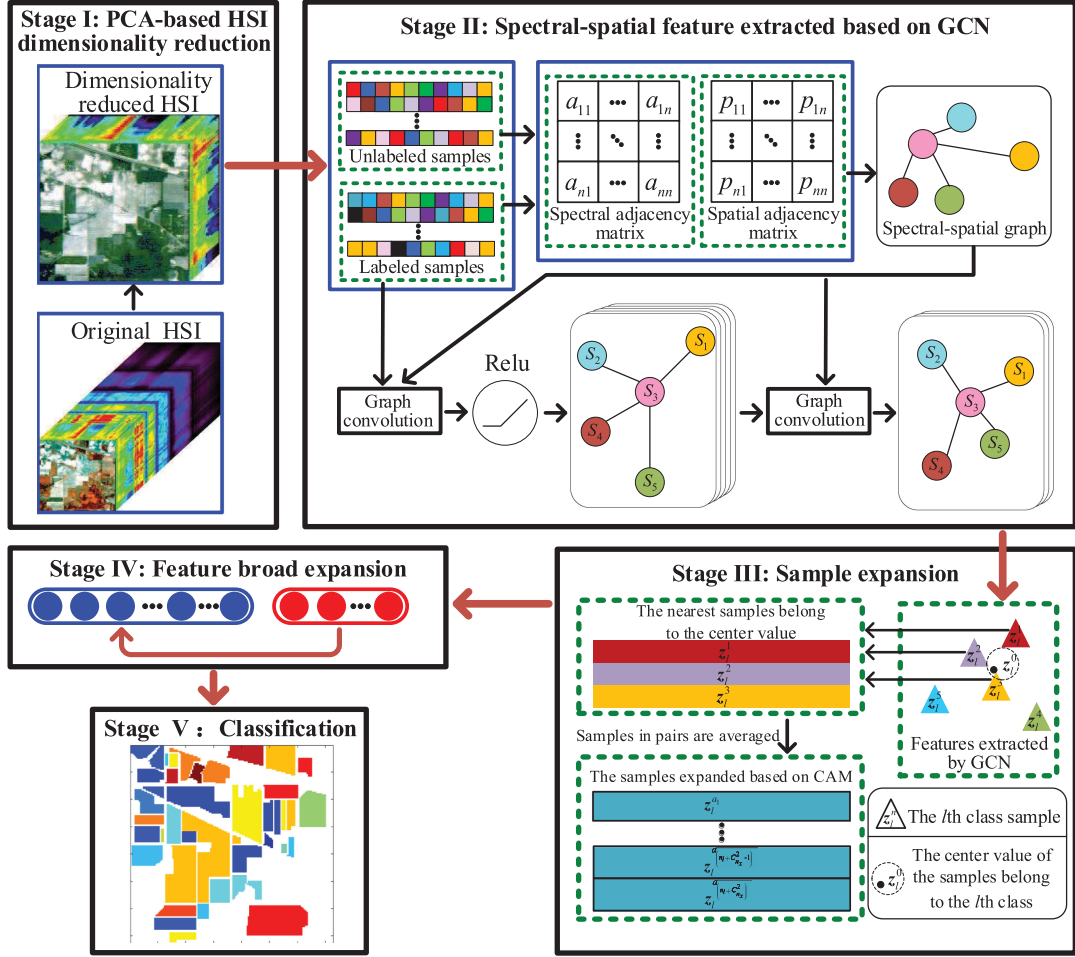


Fig. 1. Flowchart of GCBN for HSI classification.

- 5) The output layer weights can be calculated with the ridge regression theory.

B. Feature Extraction Based on GCN

Since there is redundant information in the original HSI band, directly entering the original HSI into the GCN will cause a dramatic increase in the network parameters and affect the classification performance of GCN. Therefore, PCA is used to reduce the dimensionality of the original HSI data \mathbf{X}_0 . Define $\mathbf{X} \in \mathbb{R}^{n \times d}$, then $\mathbf{x} \in \mathbb{R}^d$ is the signal after dimension reduction by PCA, $\theta \in \mathbb{R}^d$ is the Fourier coefficient. The graph convolution operation in the spectral domain can be expressed as each frequency of the signal x multiplied by the filter g_θ parameterized by θ in the Fourier domain

$$g_\theta \star x = \mathbf{U} g_\theta \mathbf{U}^T x \quad (1)$$

where \mathbf{U} is the matrix composed of the eigenvectors of the normalized graph Laplacian, $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$. $\mathbf{\Lambda}$ is the diagonal matrix which contains the eigenvalues of \mathbf{L} , \mathbf{D} is the degree matrix, $D_{ii} = \sum_j A_{ij}$, \mathbf{I} is the identity matrix. Then, we can regard g_θ as a function of the eigenvalues of \mathbf{L} , $g_\theta(\mathbf{\Lambda})$. In order to reduce computational consumption,

Hammond *et al.* [44] tried to approximate $g_\theta(\mathbf{\Lambda})$ by simplifying the Chebyshev polynomials $T_k(x)$ to the K th-order

$$g_\theta(\mathbf{\Lambda}) \approx \sum_{k=0}^K \theta'_k T_k(\tilde{\mathbf{\Lambda}}) \quad (2)$$

where θ' is a vector of Chebyshev coefficients, $\tilde{\mathbf{\Lambda}} = \frac{2}{\lambda_{\max}} \mathbf{L} - \mathbf{I}_N$, λ_{\max} is the largest eigenvalue of \mathbf{L} . According to [44], the Chebyshev polynomial is defined as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, $T_0(x) = 1$ and $T_1(x) = x$. Then define the expression of the convolutional filter g_θ' on signal x as

$$g_\theta' \star x \approx \sum_{k=0}^K \theta'_k T_k(\tilde{\mathbf{L}}) x \quad (3)$$

where $\tilde{\mathbf{L}} = \frac{2}{\lambda_{\max}} \mathbf{L} - \mathbf{I}_N$ represents the scaled Laplacian matrix. Equation (3) can be easily verified by exploiting the fact $(\mathbf{U} \mathbf{\Lambda} \mathbf{U}^T)^k = \mathbf{U} \mathbf{\Lambda}^k \mathbf{U}^T$. It can be seen that this expression is a K th-order polynomial regarding to the Laplacian, that is, the final filtering result only depends on the nodes at most K steps away from the center point. In this article, we consider the first-order neighborhood, i.e., $K = 1$. This means that filtering of the graph signal x only relies on the nearest node of the current node. We

further approximate $\lambda_{\max} \approx 2$, as the parameters of the neural network, can adapt to changes during large-scale training [45]. Thus, (3) can be simplified to

$$\begin{aligned} g_{\theta'} \star x &\approx \theta'_0 x + \theta'_1 (\mathbf{I}_N - \mathbf{I}_N) x \\ &= \theta'_0 x - \theta'_1 \mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mu \mathbf{P}) \mathbf{D}^{-\frac{1}{2}} x \end{aligned} \quad (4)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of spectral signatures, $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of spatial signatures, and μ is the spatial coefficient

$$a_{ij} = \begin{cases} \|x_i - x_j\|_2, & i \neq j \\ 0, & i = j \end{cases} \quad (5)$$

$$p_{ij} = \begin{cases} \|d_i - d_j\|_2, & i \neq j \\ 0, & i = j \end{cases} \quad (6)$$

where x represents the spectral feature vector of the sample, and d represents the spatial position coordinates of x .

Since reducing the number of parameters is helpful to solve overfitting problem, we set $\theta = \theta_0 = -\theta'_1$, then (4) can be converted to

$$g_{\theta'} \star x \approx \theta \left(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mu \mathbf{P}) \mathbf{D}^{-\frac{1}{2}} \right) x. \quad (7)$$

Since the eigenvalues of $\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mu \mathbf{P}) \mathbf{D}^{-\frac{1}{2}}$ are within the range $[0, 2]$, repeatedly using this operator in a deep neural network will lead to numerical instabilities and vanishing/exploding gradients [45]. To solve this problem, according to [43], we performed the renormalization trick $\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mu \mathbf{P}) \mathbf{D}^{-\frac{1}{2}} \rightarrow \tilde{\mathbf{D}}^{-\frac{1}{2}} (\mathbf{I}_N + \mathbf{A} + \mu \mathbf{P}) \tilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\mathbf{D}_{ii} = \sum_j (\mathbf{I}_N + \mathbf{A} + \mu \mathbf{P})_{ij}$. The GCN can be expressed as

$$\mathbf{S}^{(l)} = \text{Relu} \left(\tilde{\mathbf{A}} \mathbf{S}^{(l-1)} \mathbf{W}^{(l)} \right) \quad (8)$$

where $\mathbf{S}^{(l)}$ is the output of the l th layer, $\text{Relu}(\cdot) = \max(0, \cdot)$ was selected as the activation function, $\mathbf{W}^{(l)}$ denotes the trainable weight matrix contained in l th layer, and $\tilde{\mathbf{A}}$ can be calculated by

$$\tilde{a}_{ij} = \begin{cases} e^{-\frac{(\|x_i - x_j\|^2 + \mu \|d_i - d_j\|^2)}{\sigma}}, & \text{if } x_i \in \text{Nei}(x_j) \\ & \text{or } x_j \in \text{Nei}(x_i) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where σ is the spectral-spatial coefficient.

Only the three-layer graph CNN is selected. The propagation rule of the first two layers is shown in (8), and propagation rules of the last layer is as follows:

$$\mathbf{S}^{(3)} = \text{softmax} \left(\tilde{\mathbf{A}} \mathbf{S}^{(2)} \mathbf{W}^{(2)} \right) \quad (10)$$

where $\text{softmax}(z_i) = \exp(z_i) / \sum_i \exp(z_i)$ is selected as the activation function of the output. The loss function is

$$L = - \sum_{k \in \mathbf{Y}_L} \sum_{c=1}^C \mathbf{Y}_{kc} \ln \mathbf{S}_{kc}^{(3)} \quad (11)$$

where \mathbf{Y}_L denotes the set of vertex indices corresponding to the labeled samples, C is the total number of categories, and \mathbf{Y} is the category matrix. Similar to [45], we employed the gradient descent method to learn the weight parameters.

C. Sample Expansion Based on CAM

When the number of input labeled samples is insufficient, the BLS is prone to the problems of insufficient network training and overfitting. Therefore, we propose the CAM to expand the samples after graph convolution operation. First, limited labeled samples $\mathbf{X} \in \mathbb{R}^{n_l \times d_0}$ are fed into the trained GCN to obtain \mathbf{Z} with discriminative spectral-spatial features

$$\mathbf{Z} = \tilde{\mathbf{A}} \text{Relu} \left(\tilde{\mathbf{A}} \mathbf{X} \mathbf{W}^{(1)} \right) \mathbf{W}^{(2)} = \begin{bmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_l \\ \vdots \\ \mathbf{Z}_C \end{bmatrix} \in \mathbb{R}^{(C \times n_l) \times d_1} \quad (12)$$

where n_l is the number of labeled samples selected for each class, $l \in [1, C]$.

Second, the center value of the selected samples belonging to the l th class is defined as z_l^0 , which can be calculated by

$$z_l^0 = \frac{z_l^1 + z_l^2 + \dots + z_l^{n_l}}{n_l}. \quad (13)$$

Third, the n_x samples nearest to the center value are averaged in pairs to obtain $C_{n_x}^2$ samples. The expanded samples belonging to the l th class is defined as $\mathbf{Z}_l^a \in \mathbb{R}^{C_{n_x}^2 \times d_1}$

$$\mathbf{Z}_l^a = \begin{bmatrix} z_l^{a_1} \\ \vdots \\ z_l^{a_{C_{n_x}^2}} \end{bmatrix}. \quad (14)$$

Finally, the \mathbf{Z} is expanded and used as the training set \mathbf{Z}^K for GBCN, $\mathbf{Z}_l^K = [\mathbf{Z}_l; \mathbf{Z}_l^a] \in \mathbb{R}^{(n_l + C_{n_x}^2) \times d_1}$ is defined as all samples of l th class of \mathbf{Z}^K .

$$\mathbf{Z}^K = \begin{bmatrix} \mathbf{Z}_1^K \\ \vdots \\ \mathbf{Z}_l^K \end{bmatrix} \in \mathbb{R}^{C(n_l + C_{n_x}^2) \times d_1}. \quad (15)$$

The CAM can be used to extend the sample size of the data with discriminative spectral-spatial features extracted by GCN, which will provide more valuable samples for GBCN training. CAM is only used for model training, and n_x can be set according to the specific situation.

D. Spectral-Spatial Feature Broad Expansion Based on BLS

BLS is a new type of flat network designed based on the idea of RVFLNN [37]. Although the lack of linear sparse representation ability of BLS could lead to an underfitting problem, it still has such advantages as simple structure, fast calculation speed, and feature broad expansion. Therefore, BLS can be used to expand the width of the nonlinear features extracted by the GCN to further enhance the feature representation ability.

The original input is mapped to feature nodes via random weights, d^M is denoted as the number of feature node groups, and G^M is denoted as the feature dimension of each group. The

Algorithm 1: GCBN

Inputs: PCA-based HSI representation \mathbf{X} , labels of samples \mathbf{Y}_L , unlabeled samples \mathbf{X}_U , spatial coefficient μ , spectral-spatial coefficient σ , number of expanded samples belong to the per class n_x , regularization coefficient δ , feature dimensions of each group G^M , number of nodes in MF per group d^M , and number of nodes in EN d^E .

- Step 1. Initialize GCBN network parameter.
- Step 2. Calculate the spectral-spatial adjacency matrix $\tilde{\mathbf{A}}$ according to (9).
- Step 3. Pretrain GCN with labeled samples \mathbf{X}_L and unlabeled samples \mathbf{X}_U .
- Step 4. Extract features \mathbf{Z} according to (12). Calculate expanded samples \mathbf{Z}^K according to (12)~(15) and take \mathbf{Z}^K as the training set for GCBN.
- Step 5. Calculate the network weights \mathbf{W}^O of GCBN according to (16)~(20).
- Step 6. Calculate the predictive labels \mathbf{Y} according to (21).

Outputs: Predictive labels \mathbf{Y} .

i th group MFs is

$$\mathbf{M}_i = \mathbf{Z}^K \mathbf{W}_{ei} + \beta_{ei}, i = 1, \dots, d^M \quad (16)$$

where \mathbf{W}_{ei} and β_{ei} are connection weights and bias from \mathbf{Z}^K to MF. Sparse autoencoder is used to fine-tune the initial \mathbf{W}_{ei} , $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_1, \dots, \mathbf{M}_{d^M}]$. To further enhance the feature representation capability, \mathbf{M} is randomly mapped to EN to achieve feature broad expansion by

$$\mathbf{H}_j = \varphi(\mathbf{M} \mathbf{W}_{hj} + \beta_{hj}), j = 1, \dots, G^E \quad (17)$$

where \mathbf{W}_{hj} and β_{hj} are connection weights and bias from MF to EN, respectively, and $\varphi(\cdot)$ is tansig function here.

Finally, MF and EN are simultaneously mapped to the output layer, and the output of the GCBN is

$$\mathbf{O} = [\mathbf{M} \mid \mathbf{H}] \mathbf{W}^O. \quad (18)$$

The objective function of the GCBN is as

$$\operatorname{argmin}_{\mathbf{W}^O} \|\mathbf{O} - \mathbf{Y}^K\|_2^2 + \delta \|\mathbf{W}^O\|_2^2 \quad (19)$$

where δ is the regularization parameter. Then the network weights of GCBN can be calculated with the ridge regression as

$$\mathbf{W}^O = \frac{[\mathbf{M} \mid \mathbf{H}]^T \mathbf{Y}^K}{\delta \mathbf{I} + [\mathbf{M} \mid \mathbf{H}]^T [\mathbf{M} \mid \mathbf{H}]} \quad (20)$$

where \mathbf{Y}^K is the label corresponding to \mathbf{Z}^K . The predicted result can be calculated by the following formula

$$\mathbf{Y} = [\mathbf{M} \mid \mathbf{H}] \mathbf{W}^O. \quad (21)$$

In summary, the steps of semisupervised HSI classification based on GCBN are summarized as follows.

III. EXPERIMENTS

A. HSI Datasets

Three real HSI datasets were selected in our experiments.

Indian Pines dataset was acquired by AVIRIS sensor over the Indian Pines test site in North-western Indiana, containing 145×145 pixels and 224 bands. This image is mainly used for agricultural related research with two-third of agricultural land, one-third of forests, and other natural perennial vegetation, including 16 classes.

Botswana dataset was acquired by Hyperion sensor over the Okavango Delta, Botswana, containing 1476×256 pixels and 242 bands and including 14 classes. After removing noise, atmospheric and water absorption, and overlapping bands, the remaining 145 bands are used for the experiment.

Kennedy space center (KSC) dataset was acquired by AVIRIS sensor over Florida, containing 614×512 pixels and 224 bands and including 13 classes. After removing the water absorption and noise bands, 176 bands of the image are reserved for the experiment.

B. Experimental Result

To verify the validity and superiority of the proposed GCBN, the following 11 classifiers are selected for comparison:

- 1) traditional classification method: SVM [9];
- 2) deep learning methods: 2D-CNN [24], GCN [45], SS-GCN [43], MDGCN [42];
- 3) broad learning methods: BLS [37], SBLS [40];
- 4) GCBN without CAM: GB;
- 5) replacing CAM of GCBN with the data augmentation methods in [26] and [32] respectively: GZB, GMB; and
- 6) replacing GCN of GCBN with Graphsage [46]: GSCB.

The experimental settings are as follows.

- 1) Since Wan *et al.* [42] also selected the Indian Pines and KSC datasets to test the performance of MDGCN, here we directly refer to [42] to select the hyperparameters of MDGCN. The hyperparameters of the remaining 6 comparison classifiers were set via grid search method;
- 2) A three-layer GCN with 40 hidden nodes is used in GCBN. The epoch is 200 and the learning rate is 0.01, $\mu = 30$, $\sigma = 6$, $\delta = 0.01$, $n_x = n_c - 2$, where n_c is the number of labeled samples. The feature dimensions of each group $G^M=30$, number of nodes in MF per group $d^M=15$, and number of nodes in EN $d^E=600$ are set via grid search method;
- 3) All eight classifying methods are implemented in PyTorch and MATLAB R2017a using a computer with a 3.60 GHz Intel Core i5-6500 CPU and 8 GB of RAM;
- 4) We select 4 evaluating indexes to evaluate the experimental results, including per-class accuracy (%), overall accuracy (OA, %), Kappa coefficient, and consumed time (Time, s), where the consumed time here means the training and testing time of the classifier. To eliminate the influence of random factors, each experiment is conducted ten times to get the average value of all indexes;

TABLE I
COMPARISON OF CLASSIFICATION PERFORMANCE ON INDIAN PINES DATASET

Surface object	SVM [9]	2D-CNN [24]	GCN [45]	BLS [37]	SBLS [40]	SSGCN [43]	MDGCN [42]	GB	GZB	GMB	GSCB	GCBN
I1 (%)	69.76	82.93	100	100	100	100	91.30	90.24	100	100	100	97.56
I2 (%)	32.26	2.62	35.28	57.39	36.54	55.38	72.13	91.29	73.23	81.31	91.27	90.16
I3 (%)	37.72	47.76	73.39	42.22	64.36	45.21	73.49	57.94	76.73	57.94	64.60	66.06
I4 (%)	46.91	25.43	90.52	80.18	100	72.41	88.61	95.69	96.12	79.31	100	100
I5 (%)	62.49	37.24	50.42	78.62	74.90	69.24	90.89	84.31	83.89	65.48	85.68	89.96
I6 (%)	72.86	50.62	87.72	96.06	78.76	50.96	78.63	98.48	93.79	100	92.52	74.76
I7 (%)	86.52	43.48	100	100	100	100	100	100	100	100	100	100
I8 (%)	63.23	56.45	99.15	91.70	95.14	100	100	100	100	100	100	100
I9 (%)	92.00	80.02	100	90.00	100	100	100	100	100	100	100	100
I10 (%)	48.15	20.89	63.81	61.34	83.66	70.22	83.64	76.53	80.14	84.90	80.71	84.59
I11 (%)	35.86	40.04	22.61	39.75	69.27	76.73	74.54	75.47	73.35	77.59	75.97	76.98
I12 (%)	33.35	23.30	78.40	78.01	54.42	88.61	63.41	70.41	72.79	90.14	90.60	90.31
I13 (%)	91.80	97.01	100	100	100	100	95.61	100	99.50	100	99.49	100
I14 (%)	61.02	84.76	65.48	91.49	83.33	84.52	94.23	81.35	94.84	87.99	71.52	82.62
I15 (%)	31.39	12.34	99.21	69.67	84.78	80.11	83.68	91.08	97.90	82.15	97.62	91.86
I16 (%)	89.09	90.91	100	87.67	98.86	100	100	100	94.32	100	91.76	100
OA (%)	46.79	39.99	58.17	65.15	71.07	75.94	80.53	81.89	82.42	82.31	82.80	83.56
Kappa	0.4093	0.3250	0.5480	0.6107	0.6807	0.7309	0.7814	0.7951	0.8020	0.8049	0.8070	0.8152
Time (s)	3.25	2055.60	500.43	2.41	480.61	200.40	368.47	203.31	203.49	206.22	97.23	203.44

TABLE II
COMPARISON OF CLASSIFICATION PERFORMANCE ON BOTSWANA DATASET

Surface object	SVM [9]	2D-CNN [24]	GCN [45]	BLS [37]	SBLS [40]	SSGCN [43]	MDGCN [42]	GB	GZB	GMB	GSCB	GCBN
B1 (%)	87.85	95.17	46.79	99.25	38.87	66.79	47.92	83.02	74.34	81.89	87.17	65.28
B2 (%)	91.04	17.02	81.25	97.92	97.92	96.88	100	81.25	100	100	100	100
B3 (%)	67.85	28.40	58.54	59.76	100	80.89	80.08	86.99	94.72	99.19	76.83	90.24
B4 (%)	82.29	87.38	97.14	94.29	100	80.04	91.90	93.33	93.81	90.00	69.05	93.33
B5 (%)	57.08	57.46	46.21	71.59	77.27	94.31	98.86	67.42	83.33	98.48	79.55	96.21
B6 (%)	38.52	51.87	43.18	61.74	68.94	83.33	73.48	87.12	73.11	72.73	86.36	86.36
B7 (%)	83.94	79.84	71.26	98.03	98.43	84.25	100	99.21	100	100	83.07	100
B8 (%)	61.82	51.49	76.26	92.42	97.47	96.97	89.39	89.90	89.90	59.60	88.89	89.40
B9 (%)	58.45	38.66	61.17	68.93	99.68	98.71	99.68	91.91	99.35	88.03	100	91.59
B10 (%)	55.06	40.49	93.83	60.08	65.02	95.47	100	100	90.12	99.18	97.94	100
B11 (%)	55.80	52.30	83.67	83.67	100	89.67	89.00	76.00	89.67	89.67	100	100
B12 (%)	65.00	94.44	85.80	84.09	70.45	98.86	89.77	98.86	94.32	96.59	100	85.80
B13 (%)	50.34	7.12	97.34	39.54	98.10	58.94	77.95	95.92	76.81	94.92	100	79.47
B14 (%)	77.56	97.87	72.22	84.44	72.22	71.11	95.56	100	100	81.11	71.11	86.67
OA (%)	64.39	55.50	71.05	76.27	84.80	85.31	87.04	88.64	88.77	89.50	89.24	90.12
Kappa	0.6141	0.5218	0.6871	0.7430	0.8352	0.8475	0.8595	0.8769	0.8783	0.8843	0.8833	0.8930
Time (s)	1.43	1088.02	141.95	1.02	100.28	20.66	26.70	21.88	21.95	22.66	20.64	21.91

- We randomly select five samples from each class of the ground objects in the HSI dataset as labeled samples for experiments.
- In the Indian Pines dataset, the surface objects represented by I1-I16 are: Alfalfa, Corn-notill, Corn-mintill, Corn, Grass-pasture, Grass-trees, Grass-pasture-mowed, Hay-windrowed, Oats, Soybean-notill, Soybean-mintill, Soybean-clean, Wheat, Woods, Buildings-Grass-Tree-Drives, and Sybtone-Steel-Towers. In the Botswana dataset, B1-B14 represent: Water, Hippo grass, Floodplain grasses1, Floodplain grasses2, Reeds1, Riparian, Firescar2, Island interior, Acacia woodlands, Acacia shrublands, Acacia grasslands, Short mopane, Mixed mopane, and Exposed soils. In the KSC dataset, the surface objects represented by K1-K14 are: Srub, Willow swamp, CP hammock, Slash pine, Oak, Hardwood, Swamp, Graminoid, Spartina marsh, Cattail marsh, Salt marsh, Mud flats, and Water.

Tables I–III and Figs. 2–4 shows the performance comparison results of different classifiers.

It can be observed from Tables I–III and Figs. 2–4 that

- Among the 12 methods, 2-D CNN obtains the lowest OAs and Kappa coefficients on all three HSI datasets, and consumes the longest time. The reason is that the impressive performance of the deep learning network requires abundant labeled samples to ensure. When the number of labeled samples is insufficient, 2-D CNN cannot be adequately trained, resulting in low classification accuracy of HSI, even lower than that of conventional SVM. In addition, 2-D CNN has a large number of network layers and the gradient descent which needs repeated iteration training is used to learn the model. So it consumes a long time.
- BLS has the shortest time-consuming and high classification accuracy among the 12 methods. The reason is that the structure is simple and the nonlinear mapping from MF to EN in BLS achieves the broad expansion of MF and enhances the classification ability of BLS. Compared with BLS, SBLS achieves higher OAs and Kappa coefficients on all three datasets because SBLS additionally utilizes a large amount of unlabeled sample information.

TABLE III
COMPARISON OF CLASSIFICATION PERFORMANCE ON KSC DATASET

Surface object	SVM [9]	2D-CNN [24]	GCN [45]	BLS [37]	SBLS [40]	SSGCN [43]	MDGCN [42]	GB	GZB	GMB	GSCB	GCBN
K1 (%)	59.46	20.26	99.73	88.49	95.46	94.72	100	97.75	100	100	99.74	85.58
K2 (%)	66.01	36.78	92.58	37.39	67.10	92.08	100	95.80	100	100	97.06	100
K3 (%)	69.00	26.67	90.91	98.01	90.16	75.89	91.24	85.26	90.84	73.31	76.10	100
K4 (%)	45.38	32.27	53.78	42.11	40.00	65.46	67.21	83.40	72.06	89.47	62.75	100
K5 (%)	58.85	46.25	86.60	73.08	47.65	78.48	100	81.41	100	96.15	98.72	100
K6 (%)	48.53	4.82	29.30	88.30	32.26	98.67	97.77	98.21	99.55	99.55	98.66	100
K7 (%)	73.40	25.00	61.54	98.00	97.85	90.20	100	100	100	100	99.00	100
K8 (%)	73.05	35.58	91.37	97.42	82.34	97.19	57.28	91.78	100	57.75	75.12	91.78
K9 (%)	74.39	63.58	99.60	93.79	93.11	99.49	99.42	83.69	83.50	100	99.81	83.50
K10 (%)	92.26	9.43	61.28	34.59	95.66	100	100	100	100	100	100	100
K11 (%)	64.69	62.20	84.81	93.24	94.59	100	97.10	91.06	100	94.52	100	95.89
K12 (%)	67.77	53.19	34.91	75.50	83.10	35.80	100	83.53	72.89	99.40	84.74	95.38
K13 (%)	97.68	86.18	98.14	97.94	99.02	100	98.48	100	100	98.70	100	100
OA (%)	72.56	45.19	79.42	81.99	85.26	88.61	93.80	92.69	93.92	93.83	93.20	94.77
Kappa	0.6963	0.3978	0.7699	0.7998	0.8354	0.8735	0.9309	0.9187	0.9323	0.9311	0.9242	0.9421
Time (s)	2.83	1556.80	200.17	1.14	167.75	58.45	71.46	60.75	60.79	62.13	28.49	60.77

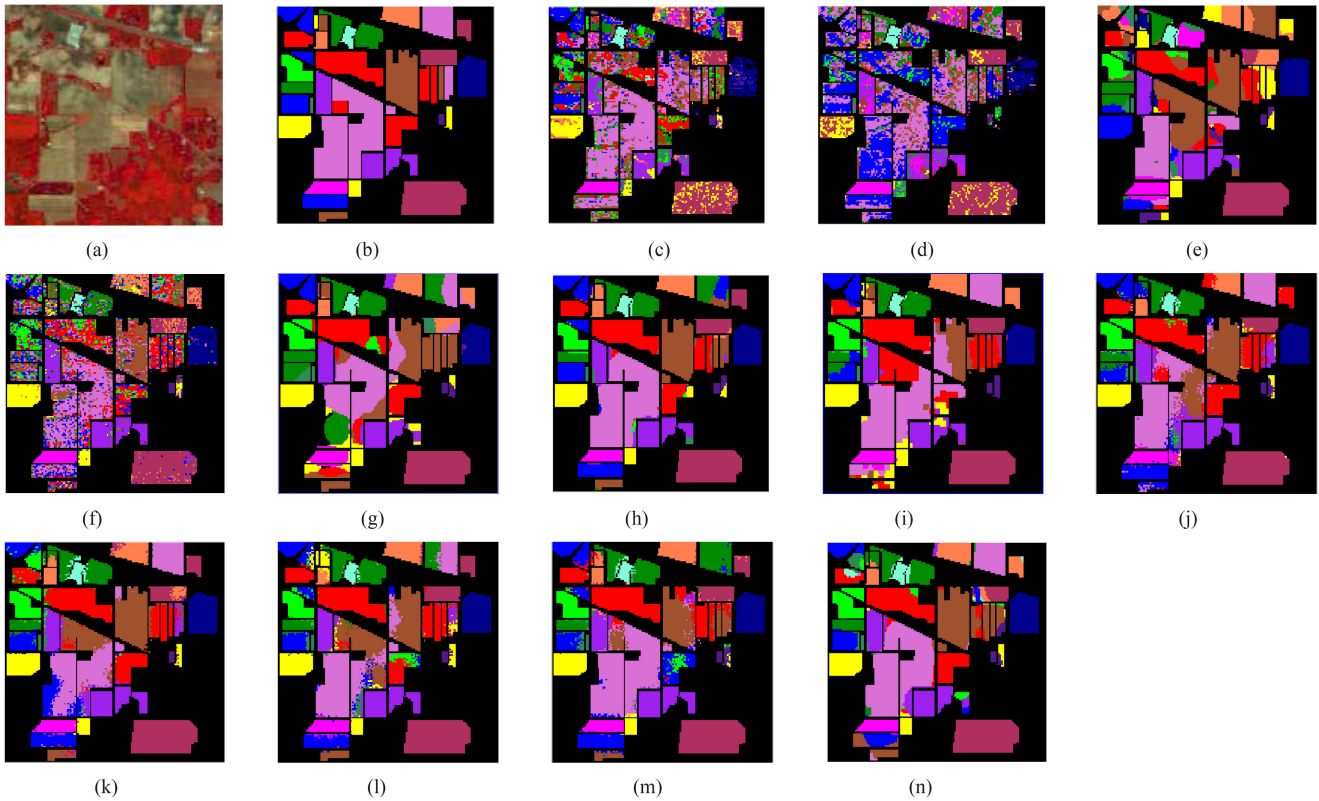


Fig. 2. Classification maps on Indian Pines dataset. (a) False-color image. (b) Ground-truth map. (c) SVM. (d) 2D-CNN. (e) GCN. (f) BLS. (g) SBLS. (h) SSGCN. (i) MDGCN. (j) GB. (k) GZB. (l) GMB. (m) GSCB. (n) GCBN.

- 3) GCN, GCBN, and MDGCN are all GCN methods, in which GCBN has the highest classification accuracy, followed by MDGCN. The reason is that both GCBN and MDGCN use spectral and spatial information of HSI, while GCN only considers spectral information. In addition, compared with GCBN and MDGCN takes the spectral and spatial information of different scales into account, and dynamically updates the constructed graph during training.
- 4) GCBN achieves the highest OAs and Kappa coefficients and the lowest time-consuming on all three datasets. The reason is as follows. First, GCBN is a semisupervised classification method that uses limited labeled samples and abundant unlabeled samples. Second, GCN helps extract more discriminant spectral-spatial features from the original HSI. Third, combinatorial average expansion of spectral-spatial features provides a great quantity of

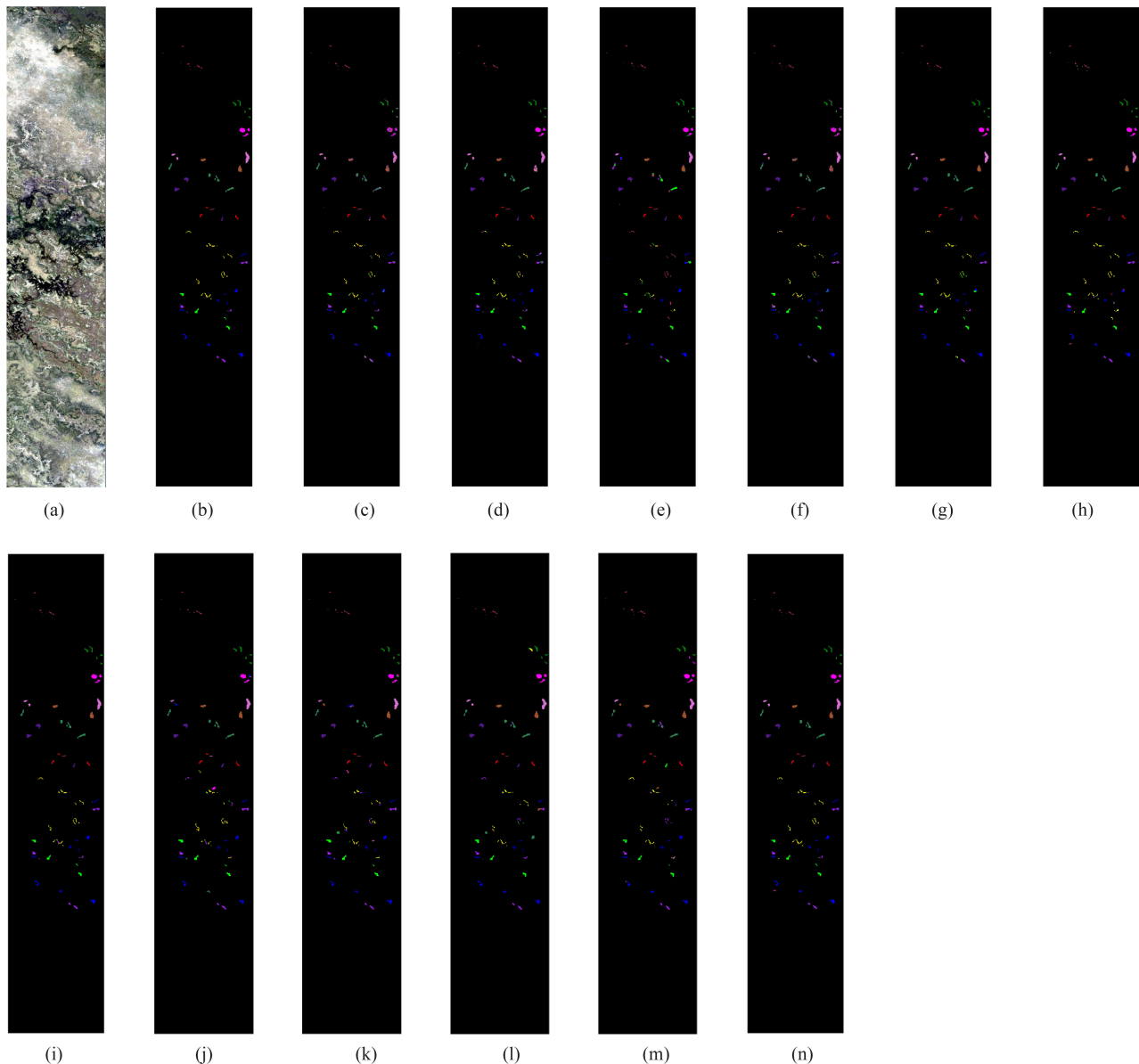


Fig. 3. Classification maps on Botswana dataset. (a) False-color image. (b) Ground-truth map. (c) SVM. (d) 2D-CNN. (e) GCN. (f) BLS. (g) SBLs. (h) SSGCN. (i) MDGCN. (j) GB. (k) GZB. (l) GMB. (m) GSCB. (n) GCBN.

valuable samples for GCBN training. Fourth, the spectral-spatial feature broad expansion further enhances the feature representation ability of GCBN. GCBN only uses one layer of graph convolutional and the structure of BLS is simple, so the learning speed of GCBN is fast.

- 5) Among the three HSI datasets, the OAs and Kappa coefficients of the 12 methods are the lowest on Indian Pines. This is because the similarity of the features in the Indian Pines dataset is relatively large. For instance, the corn-notill, corn-mintill, and corn belong to the same class in essence, so it is difficult to classify them. All the classification models have the lowest time-consuming on the Botswana dataset. This is because the Botswana dataset has the smallest sample size with only 3268 samples, while the Indian Pines dataset contains 10 249 samples.

- 6) Among the three data augmentation methods (GZB, GMB, and GCBN), GCBN achieves the highest OAs and Kappa coefficients. This is because CAM can increase the number of training samples without losing key information.
- 7) Compared with GSCB, GCBN obtains higher OAs and Kappa coefficients. The reason is that GCN integrates the global contextual information of the graph by constructing a spectral-spatial matrix of the entire graph.

Then the influence of different labeled sample sizes on the classification accuracy of HSI is studied. It can be seen from Fig. 5 that: 1) with the increase of the number of labeled samples, the OAs of all classification models show an increasing trend; 2) when the number of labeled samples is small (5 or 10 per class), the classification accuracy of 2-D CNN on the three datasets is the lowest among all models. As the number of

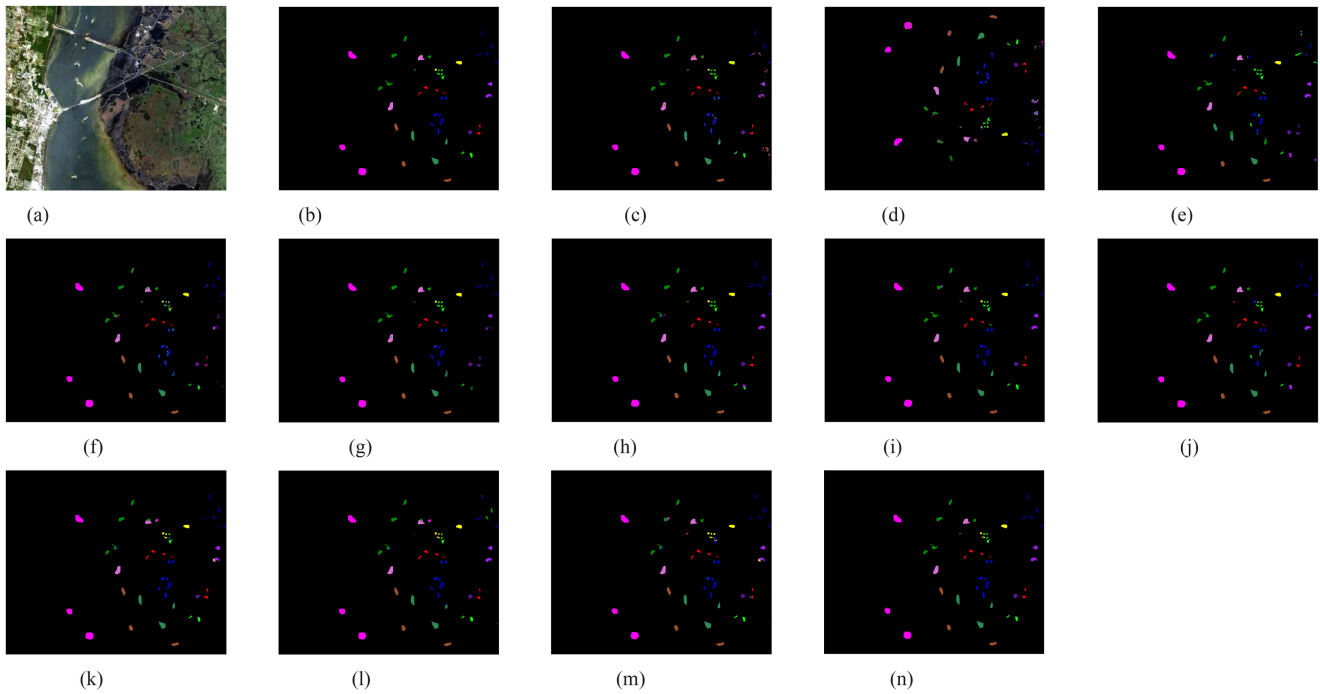


Fig. 4. Classification maps on the KSC dataset. (a) False-color image. (b) Ground-truth map. (c) SVM. (d) 2D-CNN. (e) GCN. (f) BLS. (g) SBLS. (h) SSGCN. (i) MDGCN. (j) GB. (k) GZB. (l) GMB. (m) GSCB. (n) GCBN.

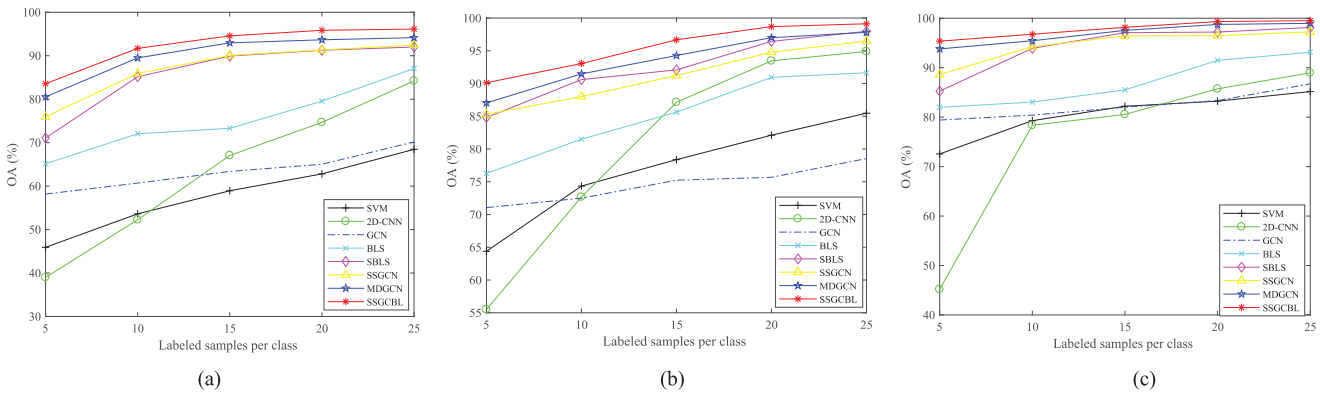


Fig. 5. OAs of various methods under different numbers of labeled samples per class. (a) Indian Pines. (b) Botswana. (c) KSC.

labeled samples gradually increases, the 2-D CNN classification accuracy increases the most.

IV. CONCLUSION

An HSI classification method, named GCBN, is proposed in this article. First, the deep and nonlinear spectral-spatial features extracted by GCN are used to replace the linear mapping features in traditional BLS, which is helpful to avoid the underfitting problem caused by the insufficient linear sparse feature representation ability of BLS. Then we propose CAM to select valuable paired samples so as to generate a sample expansion set for GCBN, which can alleviate the problem of poor classification ability caused by the limited labeled samples. Furthermore,

we use BLS to expand the width of spectral-spatial features extracted by GCN and CAM, which is able to enhance the representation ability of features and improve the classification ability of GCBN. Finally, the objective function can be easily obtained with the ridge regression theory. Experimental results on three real HSI datasets demonstrate the proposed GCBN can obtain higher classification accuracy than several other methods.

REFERENCES

- [1] L. Liang *et al.*, "Estimation of crop LAI using hyperspectral vegetation indices and a hybrid inversion method," *Remote Sens. Environ.*, vol. 165, pp. 123–134, Aug. 2015.
- [2] S. Li, R. Dian, L. Fang, and J. M. Bioucas-Dias, "Fusing hyperspectral and multispectral images via coupled sparse tensor factorization," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4118–4130, Aug. 2018.

- [3] S. Zhang, J. Li, Z. Wu, and A. Plaza, "Spatial discontinuity-weighted sparse unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5767–5779, Oct. 2018.
- [4] X. Wang, Y. Kong, Y. Gao, and Y. Cheng, "Dimensionality reduction for hyperspectral data based on pairwise constraint discriminative analysis and nonnegative sparse divergence," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1552–1562, May 2017.
- [5] B. Tu, C. Zhou, D. He, S. Huang, and A. Plaza, "Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4116–4131, Jan. 2020.
- [6] X. D. Kang, C. C. Li, S. T. Li, and H. Lin, "Classification of hyperspectral images by gabor filtering based deep network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1166–1178, Nov. 2017.
- [7] 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.
- [8] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.
- [9] 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.
- [10] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.
- [11] S. Delalieux, B. Somers, B. Haest, T. Spanhove, J. V. Borre, and C. A. Mcher, "Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers," *Remote Sens. of Environ.*, vol. 126, pp. 222–231, Nov. 2012.
- [12] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
- [13] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
- [14] D. Lungu, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 55–66, Jan. 2014.
- [15] 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.
- [16] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.
- [17] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.
- [18] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "A spatial-spectral kernel-based approach for the classification of remote-sensing images," *Pattern Recognit.*, vol. 45, no. 1, pp. 381–392, Jan. 2012.
- [19] L. Fang, S. Li, W. Duan, J. Ren, and J. A. Benediktsson, "Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6663–6674, Dec. 2015.
- [20] B. Tu, C. Zhou, X. Liao, G. Zhang, and Y. Peng, "Spectral-spatial hyperspectral classification via structural-kernel collaborative representation," *IEEE Trans. Geosci. Remote Sens. Lett.*, to be published, doi: [10.1109/LGRS.2020.2988124](https://doi.org/10.1109/LGRS.2020.2988124).
- [21] A. Sellami, A. B. Abbas, V. Barra, and I. R. Farah, "Fused 3-D spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification," *Pattern Recognit. Lett.*, vol. 138, pp. 594–600, Aug. 2020.
- [22] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
- [23] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, Feb. 2017.
- [24] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.
- [25] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [26] Y. Kong, X. Wang, and Y. Cheng, "Spectral-spatial feature extraction for HSI classification based on supervised hypergraph and sample expanded CNN," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4128–4140, Nov. 2018.
- [27] 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.
- [28] L. Mou, P. Ghamisi, and X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
- [29] H. Wang, X. Wang, C. L. P. Chen, and Y. Cheng, "Hyperspectral image classification based on domain adaptation broad learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3006–3018, Jun. 2020.
- [30] J. Nalepa, M. Myller, and M. Kawulok, "Training- and test-time data augmentation for hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 292–296, Feb. 2019.
- [31] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
- [32] C. Wang, L. Zhang, W. Wei, and Y. Zhang, "Hyperspectral image classification with data augmentation and classifier fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1420–1424, Aug. 2020.
- [33] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.
- [34] X. Zhou and S. Prasad, "Deep feature alignment neural networks for domain adaptation of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5863–5872, Oct. 2018.
- [35] C. Liu, J. Li, and L. He, "Superpixel-based semisupervised active learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 1, pp. 357–370, Dec. 2019.
- [36] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.
- [37] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jul. 2017.
- [38] S. Feng and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2020.
- [39] F. Chu, T. Liang, C. L. P. Chen, X. Wang, and X. Ma, "Weighted broad learning system and its application in nonlinear industrial process modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3017–3031, Aug. 2020.
- [40] Y. Kong, X. Wang, Y. Cheng, and C. L. P. Chen, "Hyperspectral imagery classification based on semi-supervised broad learning system," *Remote Sens.*, vol. 10, no. 5, May 2018, Art. no. 685.
- [41] Y. Kong, Y. Cheng, C. L. P. Chen, and X. Wang, "Hyperspectral image clustering based on unsupervised broad learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1741–1745, Nov. 2019.
- [42] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May 2020.
- [43] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral-spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.
- [44] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
- [45] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.
- [46] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 28th Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.



Haoyu Wang received the M.S. degree in control theory and control engineering from the China University of Mining and Technology, Xuzhou, China, in 2017. He is currently working toward the Ph.D. degree in control theory and control engineering at the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China.

His main research interests include hyperspectral image analysis and transfer learning.



Yuhu Cheng (Member, IEEE) received the Ph.D. degree in control science and technology from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His main research interests include machine learning and intelligent system.



C. L. Philip Chen (Fellow, IEEE) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently the Dean at the School of Computer Science and Engineering with South China University of Technology, Guangzhou, China and the Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Zhuhai, China. His current research interests include computational intelligence, systems, and cybernetics.

Dr. Chen is a Fellow of AAAS, the Chinese Association of Automation, and HKIE. He is also the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control (IFAC). He is also an Accreditation Board of Engineering and Technology Education (ABET) Program Evaluator for Computer Engineering, Electrical Engineering, and Software Engineering programs. He has been the Editor-in-Chief of the IEEE TRANSACTIONS ON CYBERNETICS since 2020 and an Associate Editor for several IEEE TRANSACTIONS. He was the President of the IEEE Systems, Man, and Cybernetics Society from 2012 to 2013.



Xuesong Wang (Member, IEEE) received the Ph.D. degree in control science and technology from China University of Mining and Technology, Xuzhou, China, in 2002.

She is currently the Dean of the Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, the Dean of Xuzhou Key Laboratory of Artificial Intelligence and Big Data, and the Professor with the School of Information and Control Engineering, China University of Mining and Technology. Her main research interests include machine learning, bioinformatics, and artificial intelligence. Currently, she is the Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, INTERNATIONAL JOURNAL OF MACHINE LEARNING AND CYBERNETICS, *Acta Automatica Sinica*, and *Acta Electronica Sinica*.