Hybrid CNN-GCN Network for Hyperspectral Image Classification

Diling Liao[1], Cuiping Shi [1,*], Haiyang Wu [1]; Liguo Wang [2]

1 College of Communication and Electronic Engineering, Qiqihar University, Qiqihar 161000, China; 2020910228@qqhru.edu.cn; shicuiping@qqhru.edu.cn; 2021920323@qqhru.edu.cn.

2 College of Information and Communication Engineering, Dalian Nationalities University, Dalian 116000, China; wangliguo@hrbeu.edu.cn.

* Correspondence author: shicuiping@qqhru.edu.cn

Abstract—In recent years, convolutional neural networks (CNNs) have been impressive due to their excellent feature representation abilities, but it is difficult to learn long-distance spatial structures information. Unlike CNN, graph convolutional networks (GCNs) can well handle the intrinsic manifold structures of hyperspectral images (HSIs). However, the existing GCN-based classification methods do not fully utilize the edge relationship, which makes their performance is limited. In addition, a small number of training samples is also a reason for hindering high-performance hyperspectral image classification. Therefore, this paper proposes a hybrid CNN-GCN network (HCGN) for hyperspectral image classification. Firstly, a graph edge enhanced module (GEEM) is designed to enhance the superpixel-level features of graph edge nodes and improve the spatial discrimination ability of ground objects. In particular, considering multiscale information is complementary, a multiscale graph edge enhanced module (MS-GEEM) based on GEEM is proposed to fully utilize texture structures of different sizes. Then, in order to enhance the pixel-level multi hierarchical fine feature representation of images, a multiscale cross fusion module (MS-CFM) based on the CNN framework is proposed. Finally, the extracted pixel-level features and superpixel-level features are cascaded. Through a series of experiments, it has been proved that compared with some state-of-the-art methods, HCGN combines the advantages of CNN and GCN frameworks, can provide superior classification performance under limited training samples, and demonstrates the advantages and great potential of HCGN.

Keywords – graph convolutional network, edge enhanced, cross fusion, hyperspectral image.

1 INTRODUCTION

Usually, hyperspectral images (HSIs) have hundreds of wavebands, each of which contains information about spectral features [1]. Based on the material composition and structural characteristics reflected by HSIs, spectral information extraction of HSIs can more effectively distinguish objects of interest, with strong material recognition abilities. Currently, HSIs has been widely used in agriculture [2], forestry [3], urban planning [4], geological exploration [5] and other fields. In these applications, hyperspectral image classification is a common technique. However, hyperspectral mixing, spectral variability, and complex noise effects make it difficult to extract image discrimination information. In addition, samples of HSIs markers are scarce. Therefore, hyperspectral image classification is still a challenging topic in the field of HSIs.

In recent years, thanks to advances in equipment and the expansion of accessible image data, deep learning (DL) has made breakthroughs in the fields of computer vision [6] - [12] and natural language processing [13] - [18]. DL can automatically extract more abstract and distinctive features, avoiding artificial engineering and the need for prior knowledge. Similarly, DL is also very popular in hyperspectral image classification tasks. In hyperspectral image classification, DL frameworks mainly include automatic encoders (AEs), convolutional neural networks (CNNs), generative adversarial networks (GANs), recurrent neural networks (RNNs), capsule networks (CapsNets), Transformer networks, and graph convolutional networks (GCNs). Chen et al. [19] first used principal component analysis (PCA) [20] to reduce HSI dimensions, and then a stacked automatic encoder was designed for image feature extraction. In [21], Hang et al. designed a cascaded RNN hyperspectral image classification network based on the sequence of adjacent spectral bands. The GAN network mainly consists of two main parts [22], a generator and a discriminator. The generator generates fake images by learning the distribution of real images, and the discriminator is used to distinguish between real images and fake images generated by the generator. In [23], Odena et al. designed an auxiliary classifier using GAN and used it to handle multiple classification tasks. In addition, Radford et al. [24] combined the GAN and CNN frameworks to construct a deep convolutional GAN (DCGAN), which has been widely used.

It is worth noting that CNN is one of the mainstream frameworks for DL, and there has been a lot of outstanding work. According to the characteristics of image feature extraction, this series of CNN-based methods can be divided into three categories: spectral feature based methods, spatial feature based methods, and spectral-spatial based joint methods [25]. In [26], Hu et al. used 1-D CNN to directly extract features from the HSI spectral domain, achieving better performance than support vector machines and traditional deep learning methods. Chen et al. [27] used 2-D CNN to extract nonlinear, discriminant, and invariant features of images. In addition, they also proposed a 3-D CNN based finite element model combined with regularization to extract effective spectral-spatial features of HSIs, providing competitive results. Hamida et al. [28] segmented HSI into multiple 3D cubes and constructed 3D CNN to extract spectral-spatial joint features of images. Roy et al. [29] designed a hybrid CNN classification network combining 3-D CNN and 2-D CNN. In [29], Yu et al. proposed a spatial-spectral dense CNN framework for feedback attention, which solves the problems of high complexity, information redundancy, and inefficient description of traditional networks. However, with the gradual deepening of the CNN, the network has become more complex. Although higher-level semantic features can be obtained through deeper CNN, they also pose more challenges. For example, gradient disappearance or explosion [30], over fitting [31], computational complexity [32], and interpretability [33]. To solve the above problems, Zhong et al. [34] designed a spatial-spectral residual network (SSRN), which used a residual structure to model and extract spatial-spectral features of images. Although the above work has achieved good performance, the training of the network relies on sufficient labeled samples. However, HSI labeling is time-consuming and labor-intensive [35]. Therefore, it is necessary to design a classification network with small training samples. Recently, some classification networks based on small training samples have been proposed, and their classification performance is relatively satisfactory. For example, Ma et al. [36] proposed a dual branch multi attention mechanism (DBMA) network, and verified that the model can perform well under limited training samples. In [37], DBMA was improved by designing a dual branch dual attention mechanism (DBDA) network, and verifying the effectiveness of the attention mechanism. In addition, Roy et al. [38] proposed an attention-based adaptive spectral-spatial kernel improved residual network (A^2S^2KResNet) from the perspective of the receptive field, and achieved good classification performance under small training samples. Similarly, we also proposed a feedback expansion convolution network (FECNet) [39], which further improves classification accuracy.

In the past two years, classification methods based on the Transformer framework have emerged in endlessly. Vision Transformer (ViT) [40] is a pioneering work in the field of computer vision. In [41], He et al. combined transfer learning and ViT to propose a spatial-spectral Transformer (SST) network, which has been successfully applied to hyperspectral image classification tasks. In addition, Qing et al. [42] used attention mechanisms and ViT modeling to effectively capture continuous spectral relationships. Hong et al. [43] proposed a spectral Transformer (SF) network from the perspective of spectral sequence. In order to overcome the constraints of fixed geometric structure characteristics of convolutional kernels, Zhong et al. [44] proposed a new spectral-spatial Transformer network (SSTN). In [45], Sun et al. proposed a spectral-spatial feature tokenization Transformer (SSFTT) that captures rich spectral-spatial features and high-level semantic features. In order to solve the dependency relationship that CNN is difficult to describe over long distances, Song et al. [46] proposed a bottleneck spatial-spectral Transformer (BS2T) network. Although Transformer-based representations have strong long-distance dependency abilities, classification performance still needs to be further improved.

In contrast, graph convolution, one of the deep learning methods, is not as popular as CNN and Transformer, but it does not lack excellent work. Generally, HSIs contain complex features and are irregularly distributed. Unlike CNN, GCN can aggregate the features of nodes in non-Euclidean domain [47]. Kipf and Welling et al. [48] proposed the properties of graph convolution (GConv) through graph learning theory. According to the definition of GConv, Qin et al. [49] proposed a graph convolution classification network for extracting spatial-spectral features of HSIs. In [50], Mou et al. proposed a non-local GCN, which takes the entire image data as input to the network, but this inevitably leads to a large number of computational parameters. In order to alleviate the computational explosion, Hong et al. [51] proposed a new minibatch GCN (mini GCN). In addition, Yang et al. [52] used GraphSAGE to limit the size of the input graph. In order to reduce memory consumption, Wan et al. [53] applied superpixels to the GCN structure as graph nodes. In [54], Liu et al. proposed a dual branch semi-supervised classification network that can simultaneously extract pixel-level and superpixel-level features. However, using superpixel technology to segment HSI will occupy a large amount of memory, which to some extent limits its application. Therefore, a fast dynamic graph convolutional network and CNN (FDGC) parallel network [55] has been proposed to alleviate the problem of high complexity of the model. However, the existing models still have the following issues:

1) The spatial structure GCN based on superpixels cannot consider the fine features of pixels, and the

existing GCN-based classification methods do not fully utilize the edge relationship

2) More pixel-level spatial-spectral features are extracted through CNN square window sliding, but it is difficult to learn long-distance spatial structures information.

3) The classification performance of the above network is still limited by insufficient labeled samples.

To solve the above problem, we propose a hybrid CNN-GCN network (HCGN) for HSI classification, which is used to capture pixel-level fine information and superpixel-level long distance structural information of images. Firstly, LDA dimensionality reduction technology and superpixel segmentation are introduced. Then, a multiscale graph edge enhanced module (MS-GEEM) is proposed, and the multi hierarchical structure features of the graph are obtained through the long distance spatial relationships of the graph. Secondly, a multiscale cross fusion module (MS-CFM) is proposed to extract multi hierarchical fine features. Finally, the extracted multi hierarchical superpixel-level features and multi hierarchical pixel-level features are cascaded.

The main contributions of this article are as follows:

1) A graph edge enhanced module is proposed to enhance the representation ability of edge sets and acquire rich multi hierarchical features of the graph using a multiscale graph structure.

2) A multiscale cross fusion module is designed to acquire multi hierarchical fine features at the pixel-level of an image by learning small scale regular regions.

3) The proposed hybrid CNN-GCN network combines CNN and GCN frameworks. By learning the features of small scale regular regions and large scale irregular regions respectively, multi hierarchical pixel-level features and superpixel-level features are extracted. Extensive experiments have been conducted on four common datasets and prove the effectiveness of the proposed HCGN method with limited samples.

The rest of this article is organized as follows. Section II introduces the overall framework and main modules of HCGN in detail. Section III presents the results and analysis of a series of experiments in detail. Section IV provides a conclusion of this article and prospects for future work.

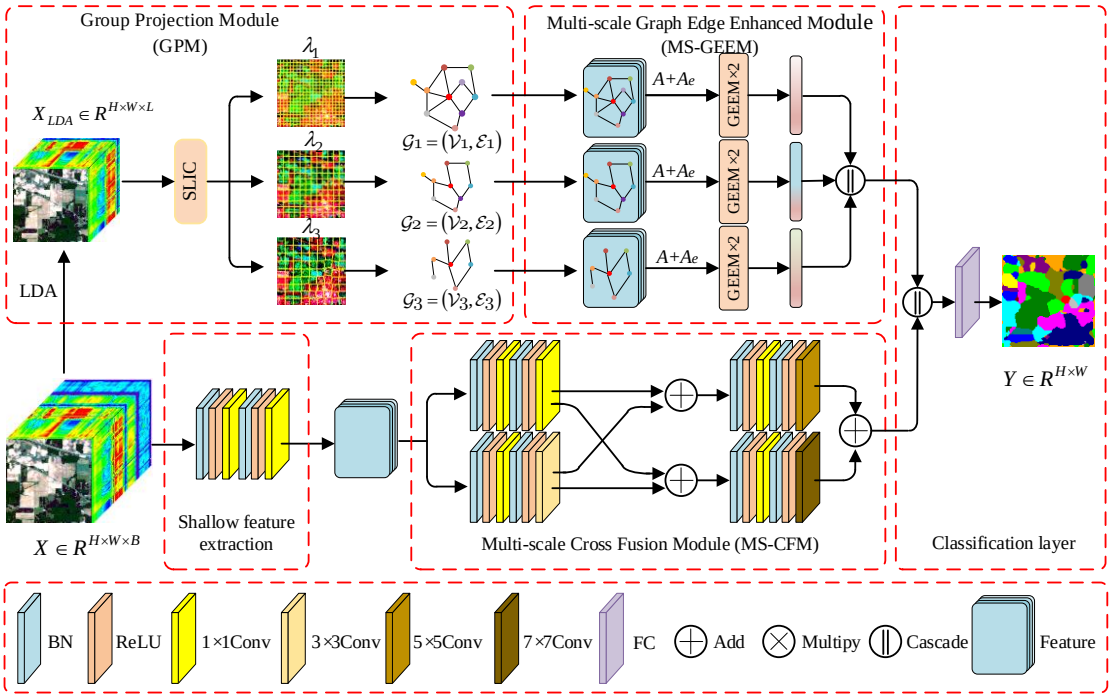


Fig. 1 The overall framework of HCGN. The model consists of five parts: a superpixel segmentation module, a multiscale graph edge enhanced module, a shallow feature extraction module, a multiscale cross fusion module, and a classification layer. X represents the input image of the model and Y represents the predicted results.

2 METHODOLOGY

The overall framework of HCGN is shown in Fig. 1. As can be seen from the figure, HCGN includes five parts: a map mapping module, a multiscale image edge enhancement module, shallow feature extraction, a multiscale cross fusion module, and a classification layer. We represent the original HSI data as $X \in \mathbb{R}^{H\times W\times B}$. Where H and W represents the spatial size of the HSI, and B represents the spectral band. Firstly, the super-pixel level features are obtained through the super-pixel segmentation and multiscale image edge enhancement module. Secondly, pixel-level features are

obtained through shallow feature extraction and multiscale cross fusion modules. Finally, cascade the above two features and send them to the classification layer for classification to obtain the label set $Y \in \mathbb{R}^{H\times W} = \{y_i \mid y_1, y_2, ...y_N\}$ of the image, and N is the maximum label value of a pixel. Next, we will describe the main modules of HCGN in detail.

2.1 Graph Projection Module (GPM)

For GCN model structures, an accurate graph plays an important role in the expressiveness of features. However, using image pixels as graph nodes will bring a huge computational burden. Fortunately, dividing pixels into different regions to construct graph nodes has proven to be effective in reducing computational costs in GCN-based models [53]. Therefore, in order to make the constructed graph nodes as accurate as possible and alleviate the computational burden, this article uses a simple linear iterative clustering (SLIC) [56] for superpixel segmentation to complete the mapping of the image from pixel to superpixel.

The GPM structure is shown in the upper left portion of Fig. 1. Specifically, in order to make the dimensionality reduced data more discriminative and contain more information, the input data $X \in \mathbb{R}^{H\times W\times B}$ is first subjected to linear discriminant analysis (LDA) for supervised dimensionality reduction to obtain the dimensionality reduced data $X_{LDA} \in \mathbb{R}^{H\times W\times L}$. Then, in order to complete the conversion of the image from pixel to superpixel, SLIC is used to divide the image into many spatial connected and spectral similar superpixels. Finally, HSI is transformed into an undirected graph $\mathcal{G} = \mathcal{V}, \mathcal{E}$ by constructing adjacent relationships between superpixels. Where $\mathcal{V}$ and $\mathcal{E}$ represent nodes and edges of a graph, respectively. In addition, in order to divide different sizes of superpixel regions, a segmentation scale factor λ is introduced, and the total number of superpixels obtained by different λ is different, which can be expressed as follows

$$k = (H \times W) / \lambda, \quad 1 \le \lambda \tag{1}$$

where $H \times W$ represents the total number of image pixels. Through superpixel division, the superpixel set $S = \{S_i\}_{i=1}^{k}$ is obtained through SLIC. S_i represents the i-th superpixel, and $S_i = \{x_j^i\}_{j=1}^{Z_i}$. x_j^i represents the j-th pixel in S_i, and Z_i represents the total number of pixels in S_i. It is worth noting that all areas divided should meet the following conditions

$$\begin{cases} S_i \cap S_j = \varnothing, & \forall i \ne j \\ H \times W = \sum_{i=1}^{k} Z_i \end{cases} \tag{2}$$

That is, each pixel exists only in one of the divided superpixels, and the total number of pixels within all superpixels is the same as the total number of pixels in HSI.

Next, using the centroid of each superpixel as a node, the graph features can be represented as follows

$$V = [V_1, V_2, ..., V_k]^T \tag{3}$$

In the above formula, V_i represents the eigenvector of the i-th node, and V is the matrix form of node $\mathcal{V}$.

It is worth noting that depending on the segmentation scale factor λ, the total number of super pixels divided by an image is different, which means that the image information contained is different. Therefore, in order to construct graphs on different neighborhood scales to capture multiscale feature information, HSI can be converted into three corresponding undirected graphs $\mathcal{G} = \{\mathcal{G}_i \mid (\mathcal{V}_1, \mathcal{E}_1), (\mathcal{V}_2, \mathcal{E}_2), (\mathcal{V}_3, \mathcal{E}_3)\}$ based on the scale factor $\lambda = \{\lambda_i \mid n_1, n_2, n_3\}$.

2.2 Multiscale Graph Edge Enhanced Module （MS-GEEM）

The effectiveness of multiscale information in feature extraction tasks for hyperspectral image classification has been widely proven [57] [58]. This is because HSI contains complex terrain structures, and contextual information extracted at different scales can enrich the regional features of an image. In the method designed in this article, we use the three undirected graphs provided in Section A to capture multiscale spectral-spatial information of images. In addition, considering the different contributions of

different nodes in the graph to feature extraction, in order to enhance the representation ability of effective edge nodes, inspired by edge convolution (EdgeConv) [59], we proposed a Multiscale graph edge enhanced module (MS-GEEM). Next, we will introduce the proposed GEEM and MS-GEEM in detail.

Generally, graph $\mathcal{G} = \mathcal{V}, \mathcal{E}$ is encoded using a node matrix H and an adjacency matrix A. The i-th row of h represents the i-th node, and $A_{i,j}$ represents the edge weight values of the i-th and j-th nodes. The Laplace operator [60] can be defined as $L = D - A$. By further standardizing it, graph Laplace can be defined as

$$\hat{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{4}$$

In the above formula, I is the identity matrix. Assuming that the graph signal of each node is $x \in R^N$, graph convolution is defined as

$$g_\theta * x \approx \sum_{k=0}^{K} \theta_k T_k \left(\frac{2}{\lambda_{\max}} \hat{L} - I_N \right) x \tag{5}$$

where, $x \in R^N$ represents the node signal of the graph, g_θ represents the spectral filter, $*$ represents the spectral convolution operation, K represents the order of the spectral filter g_θ, T_k represents the Chebyshev polynomial, $\lambda_{\max}$ represents the maximum eigenvalue of $\hat{L}$, and $\theta \in R^K$ represents the K order vector of the Chebyshev coefficients.

For ease of application, Kipf and Welling [48] take K of formula (5) as 1 and $\lambda_{\max}$ as 2. Formula (5) is simplified as

$$g_\theta * x = \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x \tag{6}$$

Formula (6) is normalized and the convolution is calculated as

$$g_\theta * x = \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \right) x \tag{7}$$

In the above equation, $\tilde{A} = A + I_N$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. In multi-dimensional image processing, graph signals can be expanded to $X \in R^{N \times C}$, and C is the channel dimension of the image. Therefore, the propagation rules of the graph convolution model are defined as

$$H^{l+1} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^{(l)} \right) \tag{8}$$

where H^l and H^{l+1} represent the input and output of the first layer, σ represents the activation function, and $W^{(l)}$ represents the weight of the first layer, respectively. Generally, the GCN used consists of two layers, and the output of the second layer uses a softmax classifier. Finally, the convolution of GCN is defined as

$$H^{l+1} = f\left(H^l, A\right) = soft\max\left(\hat{A} \operatorname{Re} LU \left(\hat{A} H^l W^{(0)} \right) W^{(1)} \right) \tag{9}$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, and $f(\cdot, \cdot)$ is a graph convolution function. The structure of GCN is shown in Fig. 2 (a).

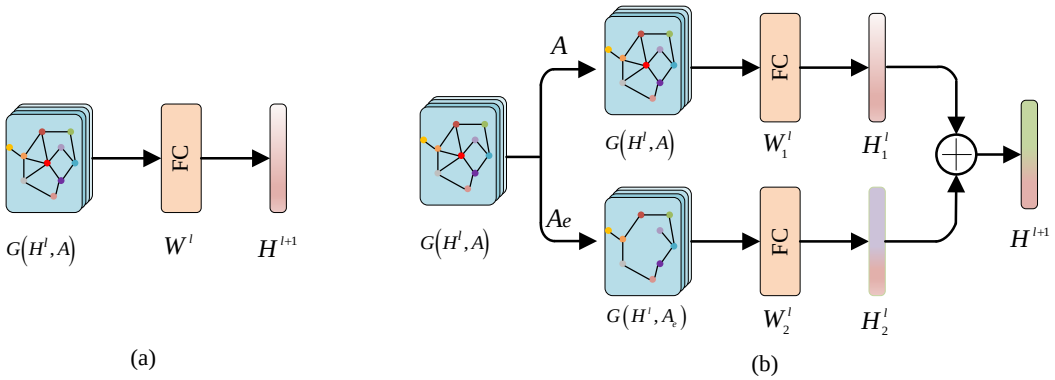


Fig. 2. Comparison of graph convolution structure and graph edge enhanced structure. (a) Graph convolution structure. (b) Graph edge enhanced structure.

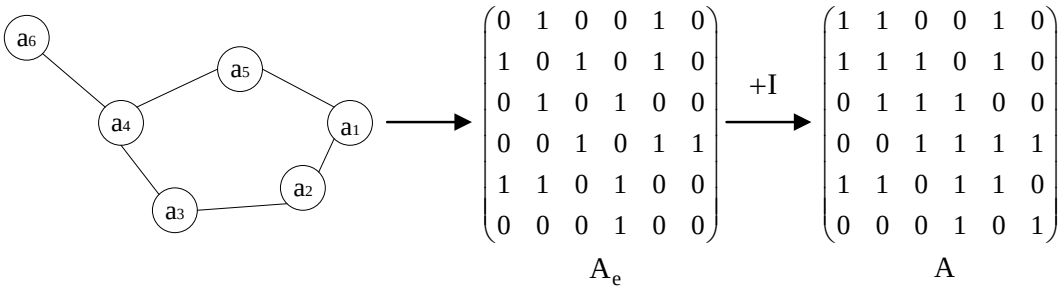


Fig. 3 Acquisition of adjacency matrix A. The relationship between matrix A and matrix A_e is $A = A_e + I$.

Through the above analysis, it can be found that GCN uses the connection relationship between graph nodes to transfer information from the current layer and adjacent nodes to the next layer. However, this connection relationship is obtained by constructing an adjacency matrix. In other words, in addition to its own nodes, the information about the surrounding nodes is also very important. Taking a 6-node graph as an example, the acquisition process of adjacency matrix A is shown in Fig. 3. Matrix A_e is the matrix from which adjacency matrix A removes its own information. Therefore, considering the domain information of surrounding nodes, we propose a graph edge enhanced module (GEEM), which enhances the transfer of self and neighborhood information. Its structure is shown in Fig. 2 (b). Specifically, it is assumed that H^l and H^{l+1} represents the inputs and outputs of l-th layer, and matrices A and A_e serve as adjacency matrices for the two branches, respectively. Similarly, using a two-layer GEEM for image feature extraction, the entire process can be represented as

$$\begin{aligned} H^{l+1} &= f\left(H^l, A, A_e\right) \\ &= soft\max\left(\left(A \oplus A_e\right) \mathrm{Re}LU\left(\left(A \oplus A_e\right) H^l W^{(0)}\right) W^{(1)}\right) \end{aligned} \tag{10}$$

where $\oplus$ represents the fusion of two features. W^i represents the weight of the i-th layer graph.

It is worth noting that HSI contains complex terrain structures, and contextual information extracted at different scales can enrich the regional features of an image. Therefore, in order to further capture multiscale image spectral-spatial information and enhance the representation ability of features, we use three different superpixel scale factors to obtain three different graph structures, and apply them to the proposed GEEM to obtain a new module, MS-GEEM. Finally, the multiscale graph features obtained from three different graph structures are cascaded, and the structure is shown in Fig. 4. The whole can be represented as

$$F = f_1\left(G_1, A, A_e\right) \| \, f_3\left(G_1, A, A_e\right) f_3 \| \left(G_3, A, A_e\right) \tag{11}$$

where G_1, G_2, and G_3 represent the inputs to the module, and F represents the outputs. $f_i(\cdot,\cdot,\cdot)$ represents the graph convolution function of MS-GEEM.

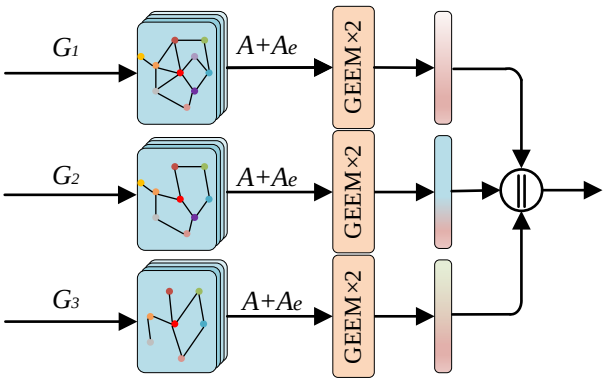


Fig. 4 The structure of MS-GEEM. (|| Operation represents the cascade of features)

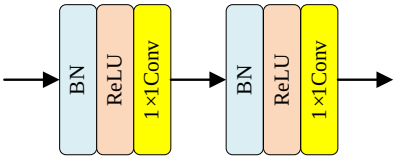


Fig. 5. Shallow feature extraction CNN module.

2.3 Multiscale Cross Fusion Module （MS-CFM）

In recent years, convolutional neural networks (CNNs) have been widely used in hyperspectral image classification tasks due to their strong feature learning abilities. Typically, CNN are used to reduce redundant information contained in HSIs and enhance the robustness of the modules. Similarly, before the multiscale cross fusion module (MS-CFM) proposed in this article, we used the CNN module to conduct shallow feature learning and reduce the redundant information contained in HSI. The structure is shown in Fig. 5. The calculation formula for two-dimensional convolution (2-D CNN) module is

$$v_{i,j}^{x,y} = f_{cnn}\left(\sum_{d}\sum_{\alpha=0}^{H_i-1}\sum_{\beta=0}^{W_i-1} K_{i,j,d}^{\alpha,\beta} v_{(i-1),d}^{(x+\alpha),(y+\beta)} + b_{i,j}\right) \tag{12}$$

where $f_{cnn}(\cdot)$ represents the 2-D CNN function (including the normalized BN layer, the activation layer ReLU, and the convolution layer). H and W represent the height and width of the input image, respectively. $K_{i,j,d}^{\alpha,\beta}$ represents the weight parameter of the d -th feature map at position (i,j), and $b_{i,j}$ represents the offset term. Therefore, the entire process of shallow feature extraction CNN module can be represented as

$$y = f_{cnn}\left(f_{cnn}(x)\right) \tag{13}$$

In the above formula, x and y represent the input and output of the image, respectively.

Next, we will introduce the MS-CFM proposed in this article in detail, and its structure is shown in Fig. 6. $\oplus$ represents the fusion of features. As can be seen from the figure, MS-CFM is mainly composed of multiple BN, ReLU, and convolution. It is worth noting that the convolutional kernels used in the convolutional layer differ in size and feature extraction branches are cross connected. This multiscale structure can effectively extract multi hierarchical features of images and improve the diversity of spatial spectral features. The entire calculation process of MS-CFM is represented as

$$y = f_{cnn3}\left(f_{cnn1}(x) \oplus f_{cnn2}(x)\right) \oplus f_{cnn4}\left(f_{cnn1}(x) \oplus f_{cnn2}(x)\right) \tag{14}$$

where x and y represent the input and output of the image, f_{cnni} represents the convolution function of the i -th convolution block, and $\oplus$ represents the fusion of features.

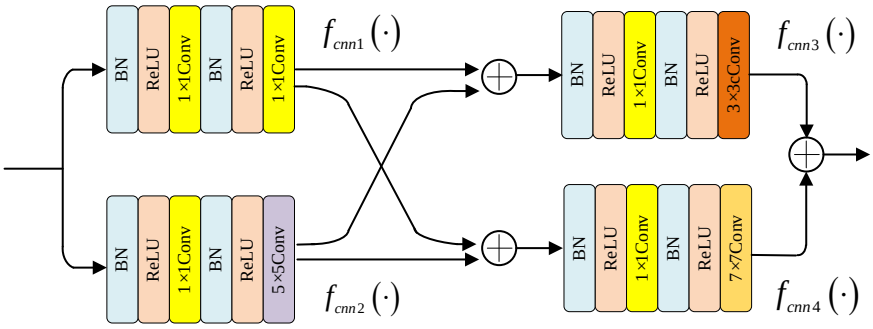


Fig. 6. The structure of MS-CFM.

2.4 Implementation Process

In order to better explain the hyperspectral image classification network HCGN, we provide an implementation of this network in this section. Taking the Indian Pines dataset as an example, the input image is $X \in \mathbb{R}^{145\times145\times200}$ and the label is $Y \in \mathbb{R}^{145\times145}$. First, the LDA dimensionality reduction output is flattened to $X_{LDA} \in \mathbb{R}^{N\times L}$, and the SLIC operation is used to complete the pixel to superpixel mapping to obtain the graph $\mathcal{G} = \mathcal{V}, \mathcal{E}$ and the correlation matrix $A \in \mathbb{R}^{k\times N}$. The number of superpixel $k = [H \times W]/\lambda$. Taking $\lambda = 10,100,150$, three types of graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{E}_3)$, and three adjacency matrices are obtained. Then, the features V of the three graphs are sent to the three branches of the MS-GEEM module for graph feature extraction to obtain superpixel multi hierarchical features. Secondly, according to formula (15) and formula (16), input X is used for spatial and spectral feature extraction through the shallow feature extraction module and MS-CFM, and the input and output sizes remain unchanged during the extraction process. Finally, the extracted superpixel features and pixel features are cascaded and sent to a softmax classifier for classification.

Table I
The implementation process of HCGN

Algorithm 1 The implementation process of HCGN

Input: HSI image data $X \in \mathbb{R}^{H\times W\times B}$, label $Y \in \mathbb{R}^{H\times W}$, scale factor $\lambda = 10,100,150$, Epoch=200.

Output: Classification accuracy and visual classification map.

1: Complete pixel to superpixel mapping of an LDA operated image, and use different scale factors to obtain three types of undirected graphs G_1, G_2, and G_3.

2: // Train the GEECN model.

3: **for** 1 to Epoch do

4: According to formula (11), input G_1, G_2, and G_3 into the MS-GEEM module to perform a graph convolution operation, and fuse the obtained results.

5: According to formula (13) and formula (14), spatial and spectral features are extracted through the shallow feature extraction module and MS-CFM.

6: Cascade the two features obtained.

7. Use the full batch gradient descent method to update network parameters.

8: Use the softmax function to identify the label.

9: **end for**

10: Save the parameters of the optimal model to obtain the classification accuracy and a visual classification map of ground object categories.

3. Experimental results and analysis

In this section, extensive experiments were conducted on four data sets using the proposed HCGN and some state-of-the-art methods for comparison, and presented all experimental results.

3.1 Dataset Description

The experiment used four relatively common HSI datasets, the Indian Pines, Pavia, Salinas, and WHU-Hi-LongKou datasets, respectively. Next, we will briefly introduce the category information and training sample size of the four datasets.

The Indian Pines dataset was captured in India by the AVIRIS imaging spectrometer in 1992. Image size is 145 × 145, including 16 categories, including corn, grass, soybeans, and woods. The image contains 220 spectral bands, excluding the water absorption band, and the remaining 200 bands are used for experiments. In addition, the spatial resolution of the image is 20m and the wavelength range

is 0.4-2.5 μm. The Pavia dataset was captured in Italy in 2003 by the ROSIS-03 imaging spectrometer. The spatial resolution of the image is 1.3m, and the spatial size is 610 × 340, and includes 103 available bands. The main types of ground objects are trees, asphalt roads, etc. The Salinas dataset was captured by the AVIRIS imaging spectrometer in the United States. Space size is 512 × 217, with 224 spectral bands and 16 categories. It is worth noting that the WHU-Hi LongKou dataset was captured on the DJI Matrice 600 Pro (DJI M600 Pro) drone platform in Longkou Town, Hubei Province, China, in 2018. Space size is 550 × 400, with a spatial resolution of about 0.463m, a wavelength range of 0.4um-1um, and 270 available spectral bands. In addition, the image is a simple farm, with main categories including corn, cotton, sesame, etc. The category details and training sample numbers of the above four datasets are shown in Table II. As can be seen, we selected a limited number of samples for each category as the training set, with the remaining being the test set.

Table II
Category Names and Number of Samples for the Four Datasets

Indian Pines				Salinas			
Class	Class name	Training	Test	Class	Class name	Training	Test
1	Alfalfa	3	42	1	Brocoil-green-weeds_1	10	1909
2	Corn-notill	42	1286	2	Brocoil-green-weeds_2	18	3540
3	Corn-mintill	24	748	3	Fallow	9	1878
4	Corn	7	214	4	Fallow-rough-plow	6	1325
5	Grass-pasture	14	435	5	Fallow-smooth	13	2545
6	Grass-trees	21	658	6	Stubble	19	3762
7	Grass-pasture-mowed	3	25	7	Celery	17	3401
8	Hay-windrowed	14	431	8	Grapes-untrained	56	10708
9	Oats	3	17	9	Soil-vinyard-develop	31	5893
10	Soybean-notill	29	875	10	Corn-senesced-green-weeds	16	3115
11	Soybean-mintill	73	2210	11	Lettuce-romaine-4wk	5	1015
12	Soybean-clean	17	534	12	Lettuce-romaine-5wk	9	1831
13	Wheat	6	185	13	Lettuce-romaine-6wk	4	871
14	Woods	37	1139	14	Lettuce-romaine-7wk	5	1017
15	Bldg-Grass-Tree-Drivers	11	348	15	Vinyard-untrained	36	6905
16	Stone-Steel-Towers	3	84	16	Vinyard-vertical-trellis	9	1717
/	Total	307	9231	/	Total	263	51432
Pavia				WHU-Hi-LongKou			
Class	Class name	Training	Test	Class	Class name	Train	Test
1	Asphalt	33	6300	1	Corn	34	34339
2	Meadows	93	17717	2	Cotton	8	8333
3	Gravel	10	1995	3	Sesame	3	3016
4	Trees	15	2911	4	Broad-leaf soybean	63	62896
5	Painted metal sheets	6	1278	5	Narrow-leaf soybean	4	4131
6	Bare Soil	25	4778	6	Rice	11	11795
7	Bitumen	6	1264	7	Water	67	66721
8	Self-blocking bricks	18	3498	8	Roads and houses	7	7089
9	Shadows	4	900	9	Mixed weed	5	5203
/	Total	210	40641	/	Total	202	203523

3.2 Experimental Setup

1) Implementation platform details: To ensure fairness in the experiment, all experiments are conducted on the same platform. The hardware devices used are an Intel (R) Core (TM) i9-9900K CPU with 128GB of memory and a NVIDIA GeForce RTX 3090 GPU with 24GB. In addition, the language framework used is PyTorch, and all final experimental results in this article are taken as the average of 30 experimental results.

2) Evaluation indicators: In order to more effectively evaluate the advantages of the model, this article uses four commonly used performance evaluation indicators, including classification accuracy of each category, overall accuracy (OA), average accuracy (AA), and Kappa coefficient.

3) Comparative algorithms: In order to evaluate the classification performance of the proposed HCGN, this article selects some state-of-the-art hyperspectral image classification networks based on deep learning, including five CNN-based methods: 3DCNN [28], DBDA [37], Hybrid-SN [29], A^2S^2KResNet [38], and FECNet [39], two Transformer-based methods: SSTN [44] and BS2T [46], and two GCN-based methods: CEGCN [54] and FDGCN [55].

3.3 Sensitivity Analysis of Parameters

In the proposed HCGN network, some super parameters can affect the classification performance of the network. For example, in order to obtain the graph topology representation of HSI, a scale factor

is introduced in the process of superpixel segmentation. Different , divided regions, and constructed graph representations differ, which will affect the amount of information interaction between nodes. Therefore, it is necessary to select the optimal for different . In this section, we selected a combination of multiple to conduct exploratory experiments on four datasets, and the experimental results are shown in Fig. 7. As can be seen from the figure, the abscissa represents the combination of four different scale factor λ, and the ordinate represents the classification accuracy. According to the spatial resolution of different datasets, the selection of different λ group aggregations varies. Specifically, λ sets in the Indian Pines dataset are $\lambda=\{10,50,100,150\}$, the λ set is $\lambda=\{50,100,150,200\}$ on the Pavia, Salinas, and WHU-Hi-LongKou datasets, and The three elements are taken out of the set sequentially and without overlap. In addition, the yellow column represents the OA value, the red column represents the AA value, and the green column represents the Kappa value. It is not difficult to find that the optimal results obtained in the four data sets are in case3, case1, case1, and case1, with the corresponding λ combinations being $\{10,100,150\}$, $\{50,100,150\}$, $\{50,100,150\}$, and $\{50,100,150\}$. In addition, from the obtained combination results, it can be found that often smaller segmentation scales can achieve better classification performance, because smaller segmentation scales can obtain more nodes, more detailed node information, and richer node interaction. However, this inevitably brings more parameters, making network training more difficult. Therefore, the optimal segmentation scale factor λ combinations for the proposed network in the four datasets are $\{10,100,150\}$, $\{50,100,150\}$, $\{50,100,150\}$, and $\{50,100,150\}$.

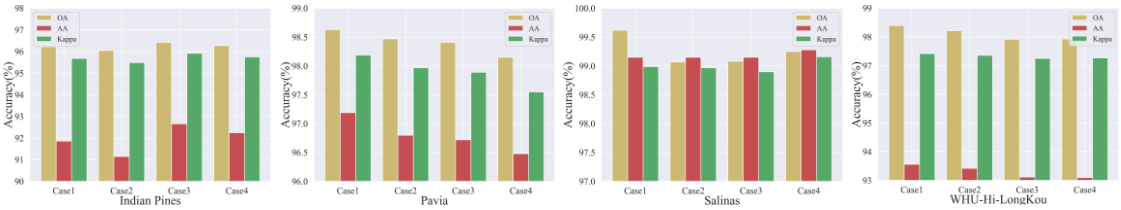


Fig. 7 Classification accuracy (including OA, AA, and Kappa) of MS-GEEM using different combinations of superpixel scale factors on four datasets.

3.4 Numerical and Visual Comparison with Other Algorithms

3.4.1 Quantitative Evaluation

In order to verify the advantages of HCGN, this article selects some state-of-the-art deep learning networks for comparison. The experimental results on the Indian Pines, Pavia, Salinas, and WHU-Hi LongKou datasets are recorded in Tables III-VI. The experimental results include the classification accuracy of each category, overall accuracy, Kappa coefficient, parameters, training time, and testing time. From Table III-VI, it can be seen that our method combines GCN and CNN, and the classification accuracy obtained under limited training samples is better than the method based on CNN and Transformer. This is because HCGN focuses more on global feature smoothing than local smoothing. This not only improves intra class similarity, but makes inter class features more discriminative. Compared with the GCN-based method, our method also has significant advantages in classification accuracy. We infer that this is because we not only use the interaction of multiscale spatial information to assist in the convolution operation of the graph, but use multi scale CNN blocks to obtain richer high-level features of the image. In addition, in the comparison of the parameters and running time results given in Tables III-VI, the parameters required by our method are relatively moderate, but the training time and testing time required by us are satisfactory. It is worth noting that the parameters and calculation time are within our capabilities, and the method HCGN proposed in this article can achieve optimal classification accuracy.

Specifically, on Indian Pines dataset, the OA values obtained by our proposed method HCGN are 30.51%, 0.40%, 25.05%, 0.72%, 0.23%, 1.18%, 1.07%, 1.39%, and 4.17% higher than those obtained by 3DCNN, DBDA, Hybrid-SN, A^2S^2KResNet, FECNet, SSTN, BS2T, CEGCN, and FDGCN, respectively, which verifies that the method proposed in this article has better and greater classification advantages than the method based on Transformer and CCN. On the Pavia dataset, the OA values obtained by HCGN were 18.30%, 0.82%, 11.24%, 3.33%, 0.50%, 4.32%, 2.66%, 0.28%, and 7.40% higher than those obtained by 3DCNN, DBDA, Hybrid-SN, A^2S^2KResNet, FECNet, SSTN, BS2T, CEGCN, and FDGCN, respectively. On the Salinas dataset, the OA, AA, and Kappa values obtained by HCGN are 99.18%, 99.21%, and 99.09%, respectively. The three values are the optimal values in all methods and can reach above 99%. On the Salinas dataset, the OA values obtained by method HCGN

are 8.67%, 0.54%, 4.46%, 1.67%, 0.32%, 3.65%, 0.39%, 0.56%, and 3.78% higher than those obtained by 3DCNN, DBDA, Hybrid-SN, A^2S^2KResNet, FECNet, SSTN, BS2T, CEGCN, and FDGCN, respectively. Therefore, through experimental comparison, we fully verify that the method proposed in this article has higher classification accuracy. In addition, by comparing Transformer based and CNN based methods, we demonstrate that GCN and CNN integrated networks that focus on global feature smoothing and multiscale features have better classification performance.

Table III
Classification accuracy (%), parameters and running time (s) of all methods on the Indian Pines dataset.

Indian Pines	CNNs					Transformers		GCNs		
	3DCNN [28]	DBDA [37]	Hybrid-SN [29]	A^2S^2KReNet [38]	FECNet [39]	SSTN [44]	BS2T [46]	CEGCN [54]	FDGCN [55]	HCGN
1	**100**	96.48	54.29	82.69	93.52	98.88	96.32	41.81	85.19	90.60
2	55.16	**95.71**	74.98	95.29	95.05	95.26	94.67	94.50	92.01	93.70
3	53.08	95.92	76.35	96.83	**97.17**	95.09	95.16	91.42	90.87	93.36
4	92.04	93.26	71.90	95.65	95.27	93.51	93.72	80.45	90.64	**96.68**
5	90.30	**99.19**	98.77	99.08	98.80	98.15	98.46	92.82	92.75	92.82
6	66.77	98.87	93.00	95.51	98.02	98.36	96.94	**99.59**	94.88	98.80
7	**99.17**	69.91	93.75	88.89	68.70	73.53	65.63	70.79	60.64	67.32
8	91.41	100	93.32	100	100	99.93	99.93	99.98	99.39	**100**
9	**87.53**	80.98	71.43	54.55	64.24	76.47	83.24	28.00	78.34	74.75
10	66.04	92.50	66.05	90.55	94.63	90.63	91.13	93.31	89.49	**94.90**
11	58.37	97.34	80.83	97.66	97.87	95.98	96.83	97.07	93.37	**98.01**
12	56.50	92.45	68.42	93.91	89.42	91.12	90.96	94.07	86.34	**95.10**
13	98.95	98.70	97.44	98.33	98.20	98.37	98.43	99.53	90.75	**99.75**
14	84.31	97.67	88.89	95.02	97.43	96.41	97.59	98.93	97.83	**99.61**
15	71.97	94.91	**95.67**	93.44	94.84	94.68	92.85	88.61	90.42	93.97
16	**99.82**	97.00	77.06	96.30	95.14	96.59	95.01	92.90	74.20	93.45
OA	65.81	95.92	71.27	95.60	96.09	95.14	95.25	94.93	92.15	**96.32**
AA	79.46	**93.81**	81.38	92.11	92.39	93.31	92.93	85.23	87.94	92.67
Kappa	60.01	95.66	78.53	94.98	95.55	94.46	94.59	94.21	91.05	**95.80**
Parameter	586k	382k	403k	373k	317k	20k	383k	166k	2445k	452k
Train	5.50	99.67	68.80	50.30	141.50	78.98	206.31	8.01	30.83	15.84
Test	1.08	4.16	1.07	4.09	7.49	1.39	20.32	1.22	0.69	0.13

Table IV
Classification accuracy (%), parameters and running time (s) of all methods on the Pavia dataset.

Pavia	CNNs					Transformers		GCNs		
	3DCNN [28]	DBDA [37]	Hybrid-SN [29]	A^2S^2KReNet [38]	FECNet [39]	SSTN [44]	BS2T [46]	CEGCN [54]	FDGCN [55]	HCGN
1	79.00	96.67	78.33	91.42	**99.24**	92.00	92.61	98.91	83.13	98.53
2	83.20	99.21	94.55	97.75	99.25	96.11	98.58	99.89	98.55	**99.93**
3	55.57	96.80	76.54	96.26	**98.44**	94.32	95.12	84.28	90.29	96.94
4	98.13	98.19	95.48	**98.82**	97.80	97.38	97.79	94.32	83.25	93.36
5	98.74	98.79	97.51	99.18	99.69	97.76	98.71	99.97	97.87	**99.98**
6	69.33	99.25	94.64	97.80	99.04	99.22	98.47	99.70	96.99	**99.83**
7	79.26	99.37	74.54	98.98	**99.98**	98.92	99.40	98.80	98.36	98.87
8	69.54	90.69	62.25	81.46	89.04	81.98	84.66	97.73	74.15	**98.44**
9	98.50	**99.15**	81.76	96.09	98.27	98.83	98.40	98.46	86.15	89.65
OA	80.32	97.80	87.38	95.29	98.12	94.30	95.96	98.34	91.22	**98.62**
AA	81.25	97.57	83.96	95.31	**97.86**	95.17	95.97	96.90	89.86	97.28
Kappa	72.97	97.08	83.09	93.72	97.50	92.37	94.62	97.79	88.31	**98.16**
Parameter	165k	202k	402k	221k	171k	13k	207k	152k	1983k	453k
Train	3.13	23.28	80.43	55.94	42.20	51.80	66.95	6.29	21.61	47.33
Test	2.75	10.52	4.78	16.74	18.60	5.28	80.14	1.08	3.10	0.63

Table V
Classification accuracy (%), parameters and running time (s) of all methods on the Salinas dataset.

Salinas	CNNs					Transformers		GCNs		
	3DCNN [28]	DBDA [37]	Hybrid-SN [29]	A^2S^2KReNet [38]	FECNet [39]	SSTN [44]	BS2T [46]	CEGCN [54]	FDGCN [55]	HCGN
1	95.31	**100**	99.59	**100**	**100**	**100**	99.61	99.60	99.89	99.64
2	95.81	99.95	97.95	**100**	**100**	96.69	99.71	**100**	99.21	**100**
3	92.76	97.12	97.31	99.09	99.24	94.89	98.67	**100**	99.83	**100**
4	98.96	95.24	94.38	95.10	95.52	95.08	95.05	99.16	92.10	**99.21**
5	97.37	99.55	97.28	**99.81**	99.46	99.53	99.43	98.48	97.01	97.85

6	99.50	**100**	96.13	99.84	99.99	98.15	99.98	99.95	97.59	99.74
7	94.78	98.59	99.60	99.49	**99.99**	99.66	99.50	99.96	99.75	99.84
8	71.75	93.52	89.74	93.55	96.65	88.08	92.67	96.14	98.61	**98.72**
9	96.91	99.03	99.87	99.79	99.91	97.72	99.57	**100**	99.66	**100**
10	83.34	96.98	97.67	**99.53**	98.82	96.33	97.02	96.72	98.33	98.42
11	93.29	95.35	91.53	95.49	98.08	88.64	95.98	98.99	95.71	**99.02**
12	93.27	98.98	99.89	99.69	99.90	99.84	99.20	99.96	93.41	**99.98**
13	96.51	99.80	92.24	98.79	99.88	**100**	99.59	99.52	79.73	99.36
14	94.27	95.43	91.65	97.03	95.43	91.64	97.35	**98.54**	93.59	97.63
15	70.41	89.77	86.71	97.70	96.78	91.31	92.78	96.77	93.67	**98.65**
16	96.58	99.88	94.98	**100**	**100**	**100**	99.99	97.56	99.90	99.43
OA	86.17	96.30	94.45	97.84	98.44	94.67	96.68	98.31	97.24	**99.18**
AA	91.93	97.45	95.41	98.43	98.73	96.08	97.88	98.83	96.12	**99.22**
Kappa	84.52	95.88	93.82	97.60	98.26	94.06	96.30	98.11	96.93	**99.09**
Parameter	595k	389k	403k	83k	323k	20k	390k	166k	2449k	438k
Train	5.25	89.48	59.41	66.13	125.60	76.43	191.34	29.85	29.84	105.54
Test	6.23	23.30	5.97	22.43	43.00	7.79	104.32	0.64	3.82	1.04

Table VI
Classification accuracy (%), parameters and running time (s) of all methods on the WHU-Hi LongKou dataset.

WHU-Hi -LongKou	CNNs					Transformers		GCNs		
	3DCNN [28]	DBDA [37]	Hybrid-SN [29]	A^2S^2KReNet [38]	FECNet [39]	SSTN [44]	BS2T [46]	CEGCN [54]	FDGCN [55]	HCGN
1	96.16	99.72	95.47	99.85	99.71	99.36	99.76	99.81	98.10	**99.89**
2	53.91	95.06	72.98	87.35	**97.48**	72.62	96.67	93.90	89.98	96.71
3	88.72	98.04	85.51	99.82	99.89	84.67	**99.91**	89.34	98.54	92.91
4	87.03	97.90	92.94	96.06	98.78	95.43	97.63	99.41	97.24	**99.50**
5	52.57	94.98	82.41	**98.93**	93.48	72.01	94.92	89.36	66.81	91.41
6	90.70	99.92	92.40	99.02	99.70	97.29	**99.94**	98.15	97.48	97.93
7	99.40	99.58	99.73	98.88	99.42	98.86	99.68	99.97	98.91	**99.99**
8	71.04	82.62	76.27	82.37	85.05	75.86	83.94	85.31	83.32	**89.81**
9	76.53	90.07	90.35	76.58	80.32	81.19	**91.36**	71.59	53.86	83.06
OA	89.73	97.86	93.94	96.73	98.08	94.75	98.01	97.84	94.62	**98.40**
AA	79.56	95.32	87.56	93.21	94.87	86.36	**95.98**	91.87	87.14	94.58
Kappa	86.37	97.19	91.99	95.68	97.47	93.06	97.38	97.15	92.93	**97.89**
Parameter	455k	509k	402k	483k	421k	21k	510k	174k	1983k	460k
Train	7.65	99.42	48.17	71.15	144.80	63.07	92.21	75.44	20.82	120.71
Test	33.10	126.58	22.88	140.15	229	33.76	404.74	1.21	15.03	1.20

3.4.2 Visual Evaluation

We also show the classification results of all methods on four datasets, as shown in Fig. 8-11. As can be seen from the figure, some CNN and Transformer based methods have more salt and pepper noise. This is because the CNN and Transformer based methods do not fully consider the spatial relationship between samples, which greatly limits classification performance. Although the GCN-based methods CEGCN and FDGCN take into account spatial relationships, the obtained classification maps still have misclassification and noise. To improve classification performance, our method combines GCN and CNN. By obtaining rich multiscale features and contextual information, the resulting classification map not only suppresses misclassification and noise, but preserves detailed information well.

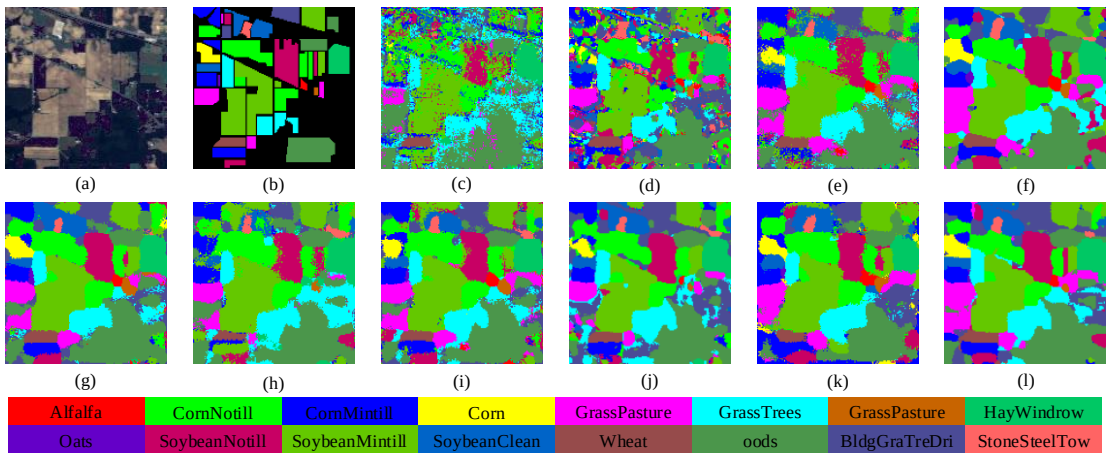


Fig. 8. Classification results of all methods on Indian Pines data. (a) Ground truth, (b) false-color map, (c) - (l) are 3DCNN (65.81%), DBDA (95.92%), Hybrid-SN (71.27%), A^2S^2KRsNet (95.60%), FECNet (96.09%), SSTN (95.14%), BS2T (95.25%),

CEGCN (94.93%), FDGCN (92.15%), and HCGN (96.32%), respectively.

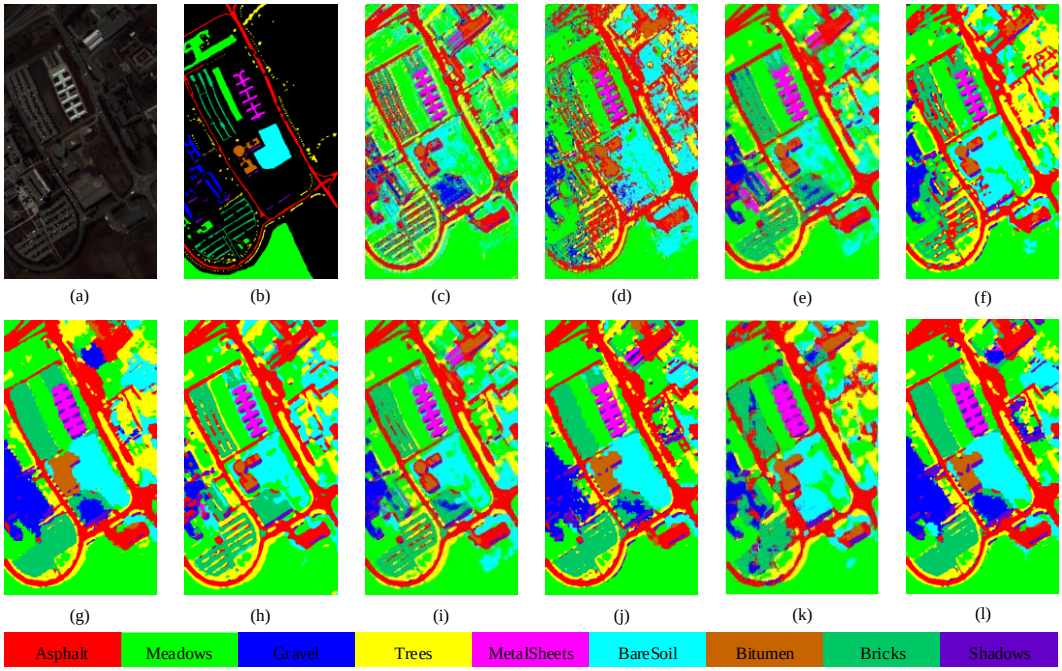


Fig. 9. Classification results of all methods on Pavia data. (a) Ground truth, (b) false-color map, (c) - (l) are 3DCNN (80.32%), DBDA (97.80%), Hybrid-SN (87.38%), A^{2}S^{2}KRsNet (95.29%), FECNet (98.12%), SSTN (94.30%), BS2T (95.96%), CEGCN (98.34%), FDGCN (91.22%), and HCGN (98.62%), respectively.

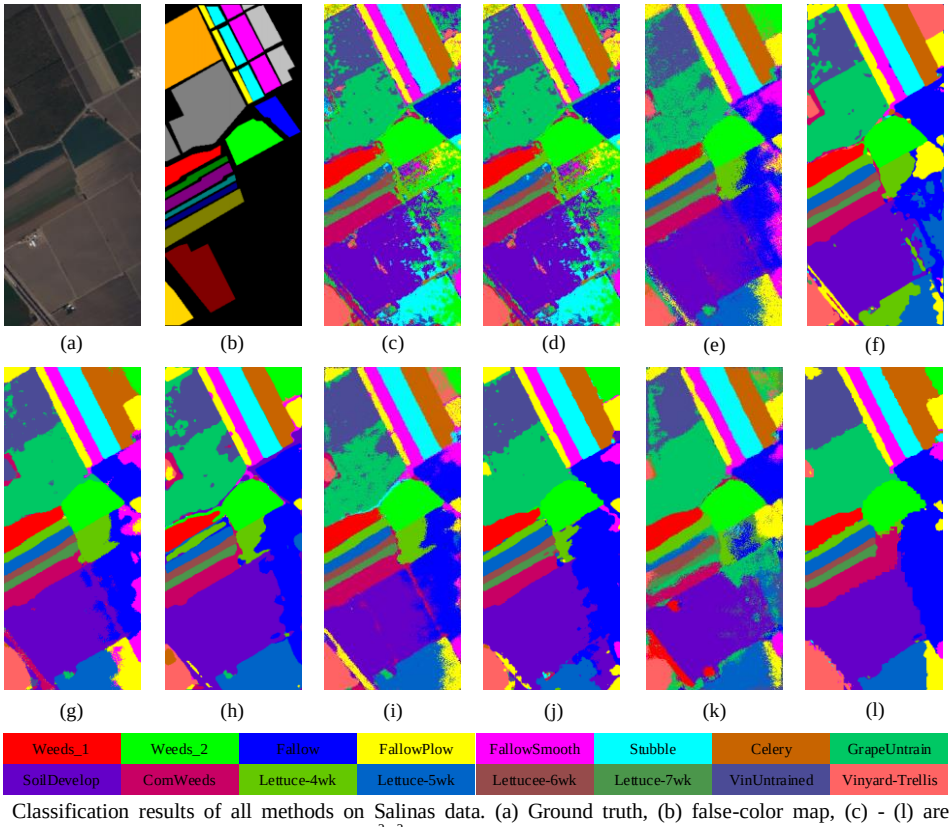


Fig. 10. Classification results of all methods on Salinas data. (a) Ground truth, (b) false-color map, (c) - (l) are 3DCNN (86.17%), DBDA (96.30%), Hybrid-SN (94.45%), A^{2}S^{2}KRsNet (97.84%), FECNet (98.44%), SSTN (94.67%), BS2T (96.68%), CEGCN (98.31%), FDGCN (97.24%), and HCGN (99.18%), respectively.

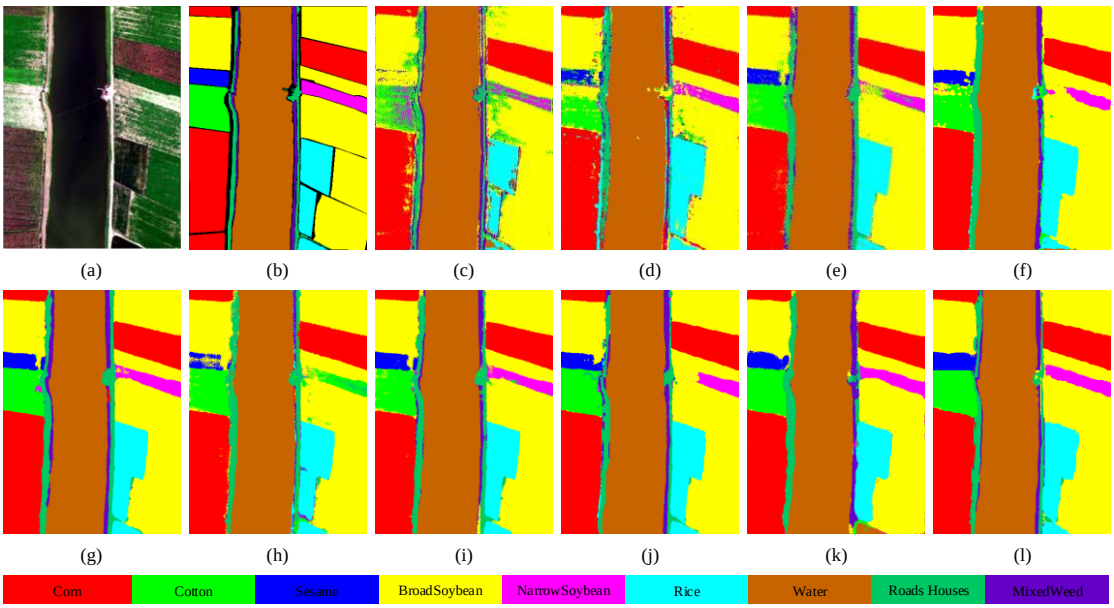


Fig. 11. Classification results of all methods on WHU-Hi-LongKou data. (a) Ground truth, (b) false-color map, (c) - (l) are 3DCNN (89.73%), DBDA (97.86%), Hybrid-SN (93.94%), A^2S^2KRsNet (96.73%), FECNet (98.08%), SSTN (94.75%), BS2T (98.01%), CEGCN (97.84%), FDGCN (94.62%), and HCGN (98.40%), respectively.

In order to clearly visualize the distribution of features, we also used t-SNE on four datasets to verify whether the features extracted by HCGN are beneficial for feature clustering in network training. The visualization results are shown in Fig. 12-15. In particular, we selected SSTN based on the Transformer framework, FECNet based on the CNN framework, and CEGCN based on the GCN framework as comparative methods. As shown in Fig. 12, in the feature visualization images obtained by SSTN and FECNet, there is serious confusion among different categories. The feature visualization obtained by CEGCN and HCGN based on the GCN framework is superior to that obtained by SSTN and FECNet. Compared to CEGCN, HCGN can maintain greater inter class distance. Similar to the results on the Indian Pines dataset, feature visualization of CEGCN and HCGN are significantly superior to SSTN and FECNet on Pavia, Salinas, and WHU-Hi LongKou datasets, as shown in Fig. 13-15. However, compared to CEGCN, HCGN has a larger inter class and smaller intra class distance due to its multiscale features and contextual information. Therefore, compared to other methods, HCGN combines CNN and GCN, focusing more on global feature smoothing, improving intra class similarity, and making inter class features more discriminative.

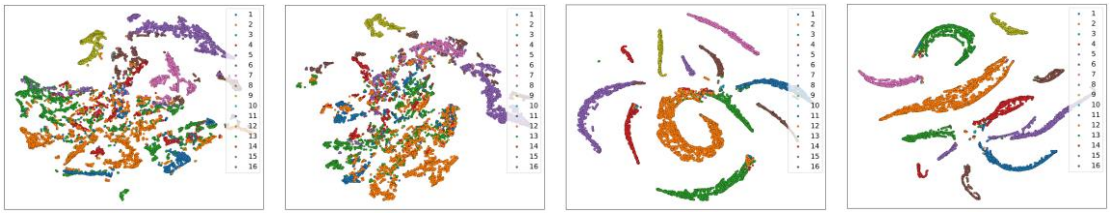


Fig. 12. Different methods for visual effects of the distribution of features using t-SNE on the Indian Pines dataset. (a) SSTN. (b) FECNet. (c) CEGCN. (d) HCGN.

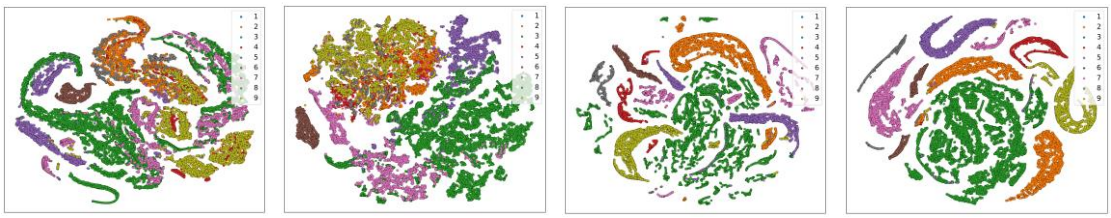
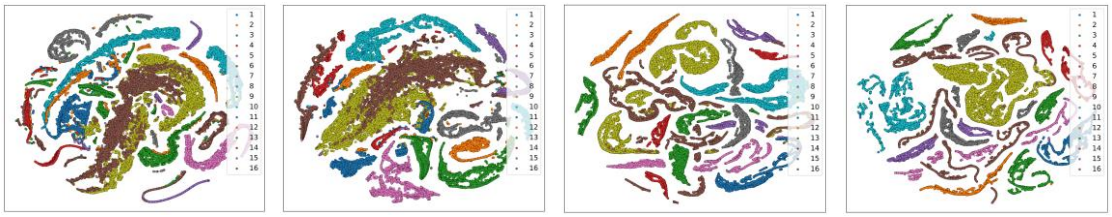
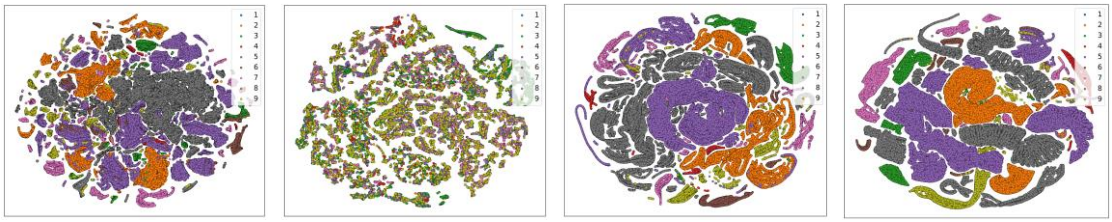


Fig. 13. Different methods for visual effects of the distribution of features using t-SNE on the Pavia dataset. (a) SSTN. (b) FECNet. (c) CEGCN. (d) HCGN.



(a) (b) (c) (d)

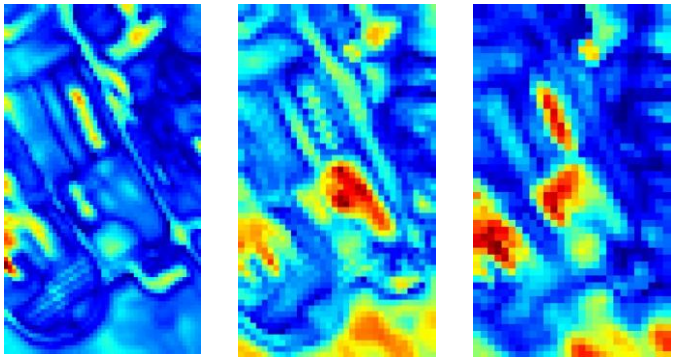
Fig. 14. Different methods for visual effects of the distribution of features using t-SNE on the Salinas dataset. (a) SSTN. (b) FECNet. (c) CEGCN. (d) HCGN.



(a) (b) (c) (d)

Fig. 15. Different methods for visual effects of the distribution of features using t-SNE on the WHU-Hi-LongKou dataset. (a) SSTN. (b) FECNet. (c) CEGCN. (d) HCGN.

In addition to the given classification result map and feature distribution visualization, in order to analyze the impact of different superpixel scale factors on feature extraction, we also conducted graph convolution output feature visualization experiments on Pavia dataset using scale factors of 50, 100, and 150, respectively. The experimental results are shown in Fig. 16. Blue indicates a small response, and red indicates a large response. It can be found that when the scale factor is small, the extracted features pay more attention to the fine features of the image, as shown in Fig. 16 (a). However, as the scale factor gradually increases, the extracted features pay more attention to global features, as shown in Fig. 16 (c). This also fully demonstrates that using different scale factors can enrich the hierarchical features of images and promote the improvement of classification performance.



(a) (b) (c)

Fig. 16. Construct a graph representation using three different superpixel scale factors on the Pavia dataset for graph convolution operations, and output feature visualization results. (a) $\lambda = 50$. (b) $\lambda = 100$. (c) $\lambda = 150$.

3.4.3 Ablation Experiment

In the proposed HCGN model, there are mainly three main modules, including GPM, MS-CFM, and MS-GEEM. In order to verify the effectiveness of multiscale image convolution, we conducted a model ablation experiment. It is worth noting that we also observed the contribution of the single branch structure GEEM of MS-GEEM to the classification accuracy OA. The experimental results are shown in Table VII. As can be seen from the table, MS-CFM can significantly improve the classification accuracy of the model. Next, GEEM fully considers the spatial relationship of the graph, effectively improving the OA value. Considering the complex structure of ground objects contained, context information extracted at different scales can enrich the regional features of the image. On the basis of GEEM, the MS-GEEM module was proposed, and the OA values of the Indian Pines, Pavia, Salinas, and WHU-Hi Long Kou datasets under limited training samples were increased to 96.32%, 98.62%, 99.18%, and 98.40%, respectively. Compared to GEEM, MS-GEEM has improved OA values by 0.29%, 0.27%, 0.50, and 0.11% on four datasets. Therefore, ablation experiments have shown that the proposed MS-CFM and MS-GEEM can effectively improve accuracy and have strong generalization ability.

Table VII
Contribution of different modules to classification accuracy OA (%).

GPM	MS-CFM	GEEM	MS-GEEM	Indian Pines	Pavia	Salinas	WHU-Hi-Long Kou
√				68.18	67.98	82.41	92.92
√	√			95.37	82.73	96.76	97.68
√	√	√		96.03	98.35	98.68	98.29
√	√		√	96.32	98.62	99.18	98.40

In addition, in order to further verify the effectiveness of the proposed module, this article compares the proposed MS-CFM with common CNN modules, with the structure shown in Fig. 17 (a) and (b). As can be seen from the figure, a typical CNN module only directly connects four convolutional blocks, while MS-CFM cross-connects four convolutional blocks. The experimental results are shown in Fig. 18. The OA, AA, and Kappa values of the two modules are represented in different colors. As can be seen from the figure, MS-CFM has higher OA, AA, and Kappa values, which also verifies that using cross fusion structures to extract spatial spectral features of HSI is more effective.

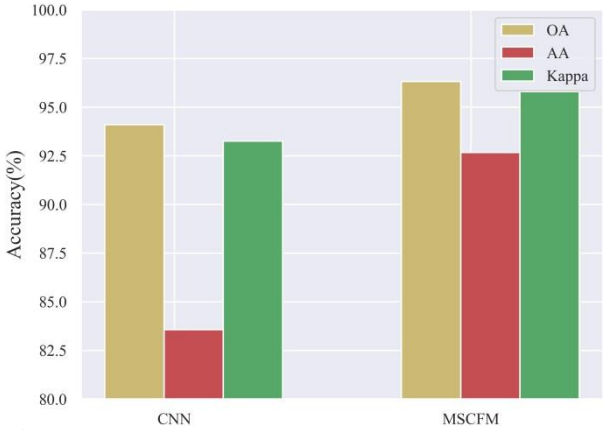


Fig. 18. Comparison of classification results between traditional CNN module and MS-CFM on Indian Pines data.

3.4.4 Impact of Training Sample Size

In this section, we investigate the impact of different training sample percentage on the classification accuracy OA of HCGN and comparison methods. Due to the significant differences in the number of labeled samples in different datasets, the percentage of labeled samples we selected in the Indian Pines, Pavia, Salinas, and WHU-Hi LongKou datasets also varies. Among them, the Indian Pines dataset selects the training sample percentage set as $\{1\%,3\%,5\%,10\%\}$, the Pavia dataset selects the training sample percentage set as $\{0.5\%,1\%,3\%,5\%\}$, the Salinas dataset selects the training sample percentage set as $\{0.1\%,0.5\%,1\%,3\%\}$, and the WHU-Hi LongKou dataset selects the training sample percentage set as $\{0.1\%,0.5\%,1\%,2\%\}$. Fig. 19 records the experimental results for four data sets. The abscissa in the figure is the training sample scale, and the ordinate is the overall accuracy OA value. Curves with different colors represent different methods. From the experimental results, it can be seen that our proposed method can still maintain good competitiveness under relatively limited training samples, which also proves that our method can still have strong feature learning ability even when the number of samples is very insufficient. This is because we fully consider the graph spatial relationships of HSIs, effectively extracting multiscale graph spatial spectral features, enriching feature diversity, and improving the discriminative ability of advanced features.

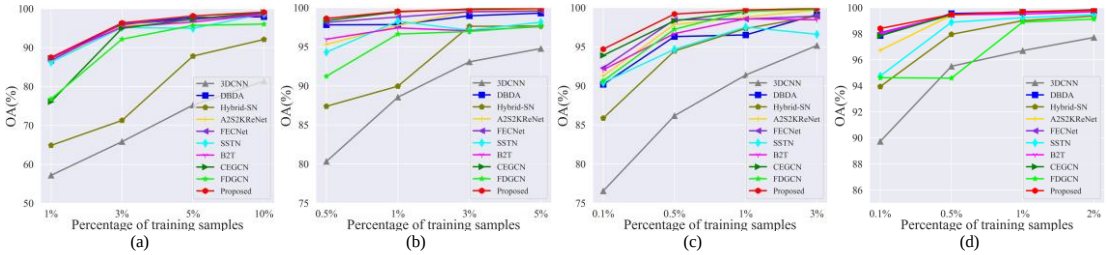


Fig. 19. Overall classification accuracy OA comparison of all methods using different training sample ratios on four datasets. (a) Indian Pines dataset. (b) Pavia dataset. (c) Salinas dataset. (d) WHU-Hi-LongKou dataset.

4. CONCLUSIONS

In this article, a hybrid CNN-GCN network was developed for hyperspectral image classification. Firstly, in order to improve the representation ability of edge sets, we enhanced the edge information of the adjacency matrix and designed an edge enhanced module to fully learn the superepixel feature information of surrounding nodes. In order to obtain rich context graph information, a multiscale graph edge enhanced module based on the edge enhanced module was proposed. However, due to the fact that the spatial structure of GCN based on superpixels cannot take into account the individual features of pixels, a multiscale cross fusion module was proposed to obtain multi hierarchical fine features at the pixel-level of an image by learning small scale regular regions. Then, the extracted superpixel-level and pixel-level features are cascaded. Finally, extensive experiments have proved that the proposed method can achieve better classification performance under limited training samples compared to other state-of-the-art CNNs and Transformer methods.

The proposed HCGN method in this paper combines multiscale CNN, GCN, and superpixel segmentation to achieve excellent classification performance with moderate training parameters. In the future, the research of lightweight GCN is an important direction, and we will continue to work in this direction.

Acknowledgment

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the reviewers for their insightful comments and suggestion.

REFERENCES

[1] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi and J. A. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690-6709, Sept. 2019.

[2] A. Bannari, A. Pacheco, K. Staenz, H. McNairn, and K. Omari, "Estimating and mapping crop residues cover on agricultural lands using hyperspectral and IKONOS data," in *Remote Sensing of Environment*, vol. 104, no. 4, pp. 447–459, 2006.

[3] Zhang, C., Du, L., & Wang, X. "Hyperspectral remote sensing technology and its applications in agriculture: A review, " in *Journal of Integrative Agriculture*, 19(3), 631-647, 2020.

[4] Du, Q., Wang, Y., & Li, J. "Urban land-use mapping using multi-temporal high-resolution hyperspectral imagery, " in *ISPRS International Journal of Geo-Information*, 7(4), 152, 2018.

[5] B. Fang, Y. Li, H. Zhang, and J. C.-W. Chan, "Collaborative learning of lightweight convolutional neural network and deep clustering for hyperspectral image semi-supervised classification with limited training samples, " in *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 161, pp. 164–178, 2020.

[6] C. Szegedy et al., "Going deeper with convolutions, " in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2015, pp. 1–9.

[7] K. He, X. Zhang, S. Ren, and J. Sun, " Deep residual learning for image recognition, " in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[8] R. Girshick, "Fast R-CNN, " in *IEEE Conference on Computer Vision and Pattern Recognition*, Dec. 2015, pp. 1440–1448.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection, " in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 779–788.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation, " in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[11] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review, " in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[12] D. Banesh et al., "An image-based framework for ocean feature detection and analysis, " *in Journal of Geovisualization and Spatial Analysis*., vol. 5, no. 2, pp. 1–21, Dec. 2021.

[13] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, " in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, Nov. 2016.

[14] F. Yin, Y.-C. Wu, X.-Y. Zhang, and C.-L. Liu, "Scene text recognition with sliding convolutional character models, " 2017, arXiv:1709.01727.

[15] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing, " in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[16] [26] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing, " in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021.

[17] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks, " in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018.

[18] Z. Tan, J. Chen, Q. Kang, M. Zhou, A. Abusorrah, and K. Sedraoui, "Dynamic embedding projection-gated convolutional neural networks for text classification, " in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 973–982, Mar. 2021.

[19] Y. Chen, Z. Lin, Z. Xing et al., "Deep Learning-Based Classification of Hyperspectral Data, " in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094-2107, 2017.

[20] M. D. Farrell and R. M. Mersereau, "On the impact of PCA dimension reduction for hyperspectral detection of difficult targets, " in *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 192–195, 2005.

[21] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative Adversarial Networks, " in *Proc. NIPS*, 2014, pp. 2672–2680.

[23] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis With Auxiliary Classifier GANs, " 2016.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, " in *Proc. ICLR*, Jan. 2016, pp. 1–16.

[25] S. T. Li, W. W. Song, L. Y. Fang et al., "Deep Learning for Hyperspectral Image Classification: An Overview, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690-6709, Sep, 2019.

[26] W. Hu, Y. Y. Huang, L. Wei et al., "Deep Convolutional Neural Networks for Hyperspectral Image Classification, " in *J. Sensors*, 2015.

[27] Y. Chen, H. Jiang, C. Li et al., "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232-6251, 2016.

[28] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-d deep learning approach for remote sensing image classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

[29] C. Yu, R. Han, M. Song et al., "Feedback Attention-Based Dense CNN for Hyperspectral Image Classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, no. 99, pp. 1-16, 2021.

[30] Li, G., Li, W., Zhang, L., & Shi, T. "Deep hyperspectral image classification with gradient boosting networks. " in *IEEE Transactions on Geoscience and Remote Sensing*, 57(10), 7929-7942, 2019.

[31] Kang, W., Zhang, H., & Wu, Y. "A new method of hyperspectral image classification based on deep learning with dropout regularization. " in *Remote Sensing*, 12(4), 706, 2020.

[32] Zhang, L., Li, W., & Shi, T. "Deep neural network pruning for hyperspectral image classification. " in *IEEE Transactions on Geoscience and Remote Sensing*, 58(8), 5667-5680, 2020.

[33] Li, D., Wu, Y., Chen, J., & Hu, J. "Hyperspectral image classification with attention-based deep feature fusion network. " in *IEEE Transactions on Geoscience and Remote Sensing*, 58(4), 2548-2560, 2020.

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

[35] Chen, X., & Zhang, L. (2020). "Active learning for hyperspectral image classification with minimum class variance. " in *IEEE Transactions on Geoscience and Remote Sensing*, 59(5), 3865-3877.

[36] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multiattention mechanism network for hyperspectral image classification, " in *Remote Sensing*, vol. 11, no. 11, p. 1307, Jun. 2019.

[37] R. Li, S. Zheng, C. Duan, Y. Yang, and X. Wang, "Classification of hyperspectral image based on double-branch dual-attention mechanism network, " in *Remote Sensing*, vol. 12, no. 3, p. 582, Feb. 2020.

[38] Roy S K, Manna S, Song T, et al.. "Attention-Based Adaptive Spectral-Spatial Kernel ResNet for Hyperspectral Image Classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, 2020:1-13.

[39] C. Shi, D. Liao, T. Zhang and L. Wang, "Hyperspectral Image Classification Based on Expansion Convolution Network," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-16, 2022.

[40] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale, " 2020, arXiv:2010.11929.

[41] X. He, Y. Chen, and Z. Lin, "Spatial-spectral transformer for hyperspectral image classification, " in *Remote Sensing*, vol. 13, no. 3, p. 498, 2021.

[42] Y. Qing, W. Liu, L. Feng, and W. Gao, "Improved transformer net for hyperspectral image classification, " in *Remote Sensing*, vol. 13, no. 11, p. 2216, 2021.

[43] D. Hong et al., "SpectralFormer: Rethinking Hyperspectral Image Classification With Transformers," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-15, 2022.

[44] Z. Zhong, Y. Li, L. Ma, J. Li and W. -S. Zheng, "Spectral–Spatial Transformer Network for Hyperspectral Image Classification: A Factorized Architecture Search Framework," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-15, 2022.

[45] L. Sun, G. Zhao, Y. Zheng and Z. Wu, "Spectral–Spatial Feature Tokenization Transformer for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-14, 2022.

[46] R. Song, Y. Feng, W. Cheng, Z. Mu and X. Wang, "BS2T: Bottleneck Spatial–Spectral Transformer for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-17, 2022.

[47] W. Guo, G. Xu, W. Liu, B. Liu, and Y. Wang, "CNN-combined graph residual network with multilevel feature fusion for hyperspectral image classification, " in *IET Computer Vision*, vol. 15, no. 8, pp. 592–607, 2021.

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

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

[50] L. Mou, X. Lu, X. Li, and X. X. Zhu, "Nonlocal graph convolutional networks for hyperspectral image classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 12, pp. 8246–8257, Dec. 2020.

[51] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[52] P. Yang, L. Tong, B. Qian, Z. Gao, J. Yu, and C. Xiao, "Hyperspectral image classification with spectral and spatial graph using inductive representation learning network, " in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 791–800, 2021.

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

[54] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "CNN-enhanced graph convolutional network with pixel-and superpixel-level feature fusion for hyperspectral image classification, " in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 10, pp. 8657–8671, Oct. 2020.

[55] Q. Liu, Y. Dong, Y. Zhang and H. Luo, "A Fast Dynamic Graph Convolutional Network and CNN Parallel Network for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-15, 2022.

[56] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods, " in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting, " in *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[58] S. Zhang and S. Li, "Spectral-spatial classification of hyperspectral images via multiscale superpixels based sparse representation, " in *Proc. IEEE IGARSS*, Jul. 2016, pp. 2423–2426.

[59] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds, " in *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, Nov. 2019.

[60] J. Bai, B. Ding, Z. Xiao, L. Jiao, H. Chen and A. C. Regan, "Hyperspectral Image Classification Based on Deep Attention Graph Convolutional Network," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-16, 2022, Art no.

5504316.