Hyperspectral Image Classification using Feature Fusion Hypergraph Convolution Neural Network

Zhongtian Ma, Zhiguo Jiang, *Member, IEEE,* and Haopeng Zhang, *Member, IEEE*

***Abstract*—Convolutional neural networks (CNN) and graph representation learning are two common methods for hyperspectral image (HSI) classification. Recently, graph convolutional neural networks (GCN), a combination of CNN and graph representation learning, have shown great potential in HSI classification problem. However, the existing GCN-based methods have many problems, such as over dependence on the adjacency matrix, usage of a single modal feature, and lower accuracy than the mature CNN method. In this paper, we propose a feature fusion hypergraph convolutional neural network (F^2HNN) for HSI classification. F^2HNN first generates hyperedges from features of different modalities to construct a hypergraph representing multi-modal features in HSI. Then, the HSI and the extracted hypergraph are input into the hypergraph convolutional neural network for learning. In addition, we proposes three feature fusion strategies. The first strategy is the most basic spatial and spectral feature fusion. The second strategy fuses the spectral features extracted by a pre-trained multilayer perceptron (MLP) with the spatial features to reduce the redundant information of the original spectral features. The third strategy uses the fusion of CNN features, spectral features and spatial features to explore the capabilities of F^2HNN. Sufficient experiments on four datasets have proved the effectiveness of F^2HNN.**

***Index Terms*—Graph convolutional networks, hypergraph learning, hyperspectral image classification, feature fusion, deep learning.**

I. Introduction

REMOTE sensing images have received extensive attention in recent decades. In particular, due to the advancement of spectral imaging technology, hyperspectral images (HSI) have been widely used in food safety detection, medical aid diagnosis, and land-cover classification [1]. HSI refers to image data with continuous spectral resolution and narrow band obtained by using hyperspectral imaging technology [2]. Depart from traditional panchromatic images or multispectral images, HSI has a more satisfying performance and greater research value in the classic remote sensing task of land-cover classification due to the extra massive spectral information. The enormous abundance of data also brings redundancy of information. How to extract effective information from HSI and discover the potential structural features of the data has become widely discussed by researchers in the task of HSI classification.

Identical to most computer vision research, methods in the HSI classification field can be roughly divided into deep learning methods and non-deep learning methods. Before the great success of deep learning, most of the methods used the non-end-to-end strategy, i.e. first extracting features and then selecting the classifier for classification. In the feature extraction strategies, the researchers' work is centered on two aspects. The first one is to reduce the dimensionality of HSI to accommodate the Hughes phenomenon [3], such as using manual feature selection [4] and dictionary-based sparse representation [5], or simply applying Principal Component Analysis (PCA) [6]. The other one is to adopt effective strategies to combine spatial and spectral features [7]–[10]. The manifold learning method in the graph embedding framework has once become a quite prevalent feature extraction scheme [11] [12]. This type of method utilizes graph Laplacian of spectral features to reduce dimensionality [13], and can fuse spatial-spectral features by establishing the adjacency matrix through the context information as well as the spectral information.

Under the influence of the rapid development of deep learning in recent years, many deep learning methods have been proposed and applied to HSI classification problems [3]. Hu *et al.* [14] first uses convolution neural networks(CNN) to directly classify HSI in the spectral domain. Hang *et al.* [15] proposed a method regarding the spectral signature as a sequence and using recurrent neural network (RNN) with several convolution layers to extract discriminative features. The most significant difference between deep-learning-based methods and traditional methods is that deep learning adopts an end-to-end framework, using multi-layer network structure and learning methods to obtain the best feature expression, while traditional methods are based on handcrafted features.

Among the deep learning methods, the CNN-based method is the most widely used and has the best performance. Lee *et al.* designed a CNN model with multi-scale convolutional filter bank and residual learning for HSI classification. Chen *et al.* [16] adopted 3-D convolution to extract spectral-spatial features and perform end-to-end training. In spite of the fact that the classification accuracy of HSI has been visibly improved, there are still some shortcomings in the development of methods based on CNN. For instance, the convolution kernels have limited receptive fields, thus it is unable to focus on long-distance dependencies. Moreover, most of the existing

This work was supported in part by the National Key Research and Development Program of China (Grant No. 2019YFC1510905), and the Fundamental Research Funds for the Central Universities. (Corresponding author: Haopeng Zhang.)

Zhongtian Ma, Zhiguo Jiang, and Haopeng Zhang are with Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University, Beijing 102206, China, with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 102206, China, and with Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies, Ministry of Education, 102206, Beijing, China. (e-mail: mazhongtian@buaa.edu.cn; jiangzg@buaa.edu.cn; zhanghaopeng@buaa.edu.cn)

Manuscript received, 2021; revised , 2021.

CNN-based methods adopt a strategy of cutting the entire HSI into small patches containing dozens of pixels, which lose a lot of non-local information.

Due to the successful application of graph embedding structure and CNN in the field of HSI classification, a novel graph convolutional network (GCN) [17] has also been tried for HSI classification. GCN is a combination and evolution of graph embedding learning and deep learning, aiming to find a more efficient feature expression method for graph data. There are usually two classification tasks that GCN focuses on, one is the classification of the entire graph data, and the other is the classification of each node in the graph data. The GCNs used for these two different tasks are not exactly the same. For the HSI classification problem, it can be transformed into a node classification problem that GCN can handle [18]. Most of the current GCN methods regard each pixel as a node, and the spectral information of each pixel as the feature of the node, thereby utilize a specific distance metric method to construct graph data from HSI [18], [19]. Hong *et al.* [20] constructed an end-to-end network cascading CNN and GCN, and trained GCN in minibatchs. In [21], super-pixel segmentation and multi-step dynamic map construction were used for GCN training. The above methods prove that there exists potential graph structures in HSI.

Nevertheless, the current GCN-based methods still have some defects. For instance, existing methods rely heavily on the construction of adjacency matrix, which represents the topological structure of graph data. However, the adjacency matrix is basically built on prior knowledge and cannot be updated through the learning process. Moreover, modeling with a simple graph structure in which edges can only represent pairwise relationships, cannot discover potentially complex structures in the data, such as hypergraph structures.

In order to solving these problems, we adopt the recently proposed hypergraph neural network (HGNN) [22] improved from GCN, and specifically ameliorate the HGNN network to a feature fusion hypergraph neural network(F^2HNN) for HSI data. As a representation learning method, HGNN adopts hypergraph structure for modeling and generalizes graph convolution to hypergraph structure. The main difference between the hypergraph structure and simple graph structure lies in the definition of the edge. The edge in the simple graph structure contains two vertices, in other words, the degree of the edge is forced to 2. In contrast, the hyperedge in the hypergraph can contain any number of nodes and is therefore degree-free, as shown in Figure 1. This difference makes HGNN have more powerful representation capabilities. In addition, due to the existence of the hyperedge weight matrix, it does not rely too much on the construction of the original adjacency matrix. Furthermore, F^2HNN uses a feature fusion strategy based on HGNN. By using different features to generate hyperedges (such as spectral-spatial features), and setting the weights of the hyperedges to trainable parameters, F^2HNN updates the weights of the hyperedges in iterations to reach the optimum.

The contribution of this paper can be summarized as follows.

1) We propose a novel improved HGNN network named F^2HNN for HSI classification. F^2HNN effectively fuses multi-modal features and automatically updates the hyperedge weights, which solves the difficulties encountered by the previous GCN-based hyperspectral classification methods.
2) We design three different feature fusion strategies and conduct experiments on multiple datasets to verify the effectiveness of F^2HNN. Further, we explore the relationship between GCN and CNN by analyzing the results of different strategies on different datasets.
3) Compared with the previous GCN-based network, the proposed F^2HNN is a concise but efficacious end-to-end network. When simply fusing the spectral-spatial features, it can achieve state-of-the-art accuracy .

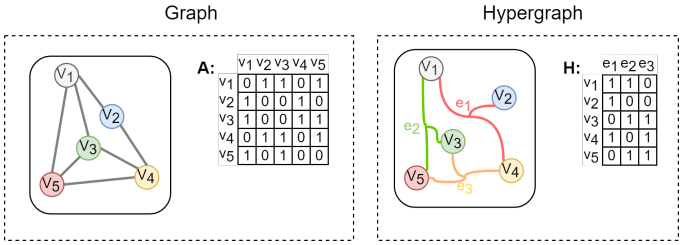


Fig. 1. Comparison of graph and hypergraph structure. Graph structure is usually represented by adjacency matrix $\mathbf{A}$, while hypergraph structure is represented by incidence matrix $\mathbf{H}$.

II. Related Works

In this section, we first review the development of hypergraph learning and its successful application in the field of computer vision. Then we introduce some researches on HSI classification using graph neural networks (GNN), summarize the characteristics of representative methods, and infer the possible development directions.

A. Hypergraph Learning

A hypergraph is a generalized graph structure that contains a set of vertices and hyperedges. The edge in the simple graph structure only connect two vertices with the degree of 2, while hyperedges in the hypergraph structure is able to connect multiple vertices and are degree-free. Compared with simple graphs, hypergraphs express more complex relationships than pairwise relationships, which can better model many practical data, such as social networks, citation networks, etc.

Hypergraph learning is generalized from graph learning and was first proposed in [23]. The purpose of hypergraph learning is to analyze data by learning the mode of information transfer between hypergraph structures. The typical tasks of hypergraph learning include node classification [24], link prediction [25], community detection [26], etc. Owing to the fact that hypergraph learning can establish high-order correlation models between data, it has been widely used in many fields, including computer vision. For instance, Wang *et al.* [27] introduced hypergraph learning into image retrieval. In the task of image classification, Yu *et al.* [28] generated hyperedges by linking the image with its nearest neighbors, and obtained the image labels and hyperedge effects at the same time through

adaptive hypergraph learning. In addition, hypergraph learning has also been successfully applied to computer vision tasks such as person re-recognition [29], 3D object classification [30], and video segmentation [31].

After the deep learning method revolutionized the area of machine learning, a series of studies that combined deep learning and graph representation learning appeared. Graph neural network (GNN) was first proposed in [32], and then various models have been gradually developed, such as graph convolution network [33], graph attention network [34], GraphSAGE (SAmple and aggreGatE) [35], etc. As a promotion of graph learning, hypergraph learning is also affected by deep learning. Feng *et al.* [22] first proposed a hypergraph neural network and introduced a spectral convolution operator on a static hypergraph. In [36], a dynamic hypergraph convolutional network was proposed. Bai *et al.* [37] embedded the two trainable operators of hypergraph convolution and hypergraph attention in GNN to extract the non-pairwise relationships, and proved the effectiveness of their method on the semi-supervised node classification task.

B. Graph neural networks for HSI classification

It is a conventional method to treat HSI as graph data and perform subsequent processing. Previous researchers used manifold learning to extract non-Euclidean relationships in HSI data. Sufficient graph learning methods have been tried for HSI classification. Gustavo *et al.* [38] adopted graph-based composite-kernel semi-supervised learning for classification. Gao *et al.* [39] proposed a bilayer graph representation learning method. In recent years, graph neural network method has become a research hotspot of HSI classification. Qin *et al.* [18] applied GCN to hyperspectral classification, and improved the original graph convolutional network to facilitate the fusion of spatial and spectral features. The ideas of [21] and [40] are similar. They both use superpixel segmentation first, and then input the segmented superpixels as nodes into the GNN for node classification. Hong *et al.* [20] explored the similarities and differences between GCNs and CNNs in HSI classification task, and proposed a classification framework by combining CNN and GCN. He *et al.* [41] adopted the common strategy of the CNN-based method, divided the HSI into a set of patches, and inputted the patches into the CNN network to extract features. Then the graph data was generated from the patch with extracted features and inputted into GCN for classification.

III. Proposed Method

The complete framework of our proposed F^{2}HNN is shown in Figure 2. We first design a strategy to extract multi-modal features from HSI, and then generate hypergraphs for these features. After having the hypergraph structure, HSI and the generated hypergraph structure are sent to HGNN for training. In this section, F^{2}HNN will be introduced in detail from three aspects: HGNN, hypergraph generation, and feature fusion strategy.

A. HGNN

1) Definition of Hypergraph: A simple undirected graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by the set of vertices $\mathcal{V}$ and edges $\mathcal{E}$. Unlike the simple graph structure, the hyperedge in the hypergraph is degree-free, which means that the hyperedge can connect more than two vertices. In addition, each hyperedge e also has a hyperedge confidence parameter $w(e)$ (usually we use weighted hypergraphs in the hypergraph learning method [23]). Therefore, a hypergraph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where the diagonal matrix $\mathbf{W}$ is the hyperedge weight matrix. Different from GCN using adjacency matrix $\mathbf{A}$ to represent graph data, HGNN utilizes incidence matrix $\mathbf{H}$ to denote hypergraph data. Both the rows and columns of $\mathbf{A}$ (size $|\mathcal{V}| \times |\mathcal{V}|$) depict vertices, while the rows and columns of $\mathbf{H}$ (size $|\mathcal{V}| \times |\mathcal{E}|$) represent nodes and edges respectively, as shown below.

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases} \tag{1}$$

Given an H matrix, the hypergraph Laplacian matrix can be calculated as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2} \tag{2}$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ denote the diagonal matrix of vertex degrees and edge degrees, with each vertex degree defined as $d(v) = \sum_{e\in\mathcal{E}} \omega(e)h(v, e)$ and each edge degree defined as $\delta(e) = \sum_{v\in\mathcal{V}} h(v, e)$. The role of $\mathbf{D}_v$ and $\mathbf{D}_e$ can be simply summarized as normalizing incidence matrix $\mathbf{H}$.

2) Convolution Operation on Hypergraph: Graph convolution is based on spectral graph theory (SGT) [42], [43]. In brief, SGT adopts the eigenvalues and eigenvectors of the graph Laplacian matrix to study the properties of the graph. How the graph convolution operation is calculated has been introduced in detail in [20], and the convolution of the hypergraph is derived from it. Given a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, the Fourier transform of a signal (vertex) $\mathbf{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n})$ is defined as

$$\hat{\mathbf{x}} = \mathbf{\Phi}^\top\mathbf{x} \tag{3}$$

where $\mathbf{\Phi}$ can be calculated by diagonalizing the positive semi-definite matrix $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^\top \tag{4}$$

where $\mathbf{\Phi} = \mathrm{diag}\,(\phi_1, \ldots, \phi_n)$ contains the orthonormal eigen vectors, and $\mathbf{\Lambda} = \mathrm{diag}\,(\lambda_\mathbf{1}, \ldots, \lambda_\mathbf{n})$ is a diagonal matrix composed of eigenvalues. Then, the hypergraph convolution operation of the signal $\mathbf{x}$ and filter $\mathbf{g}$ can be denoted as

$$\mathbf{g} \star \mathbf{x} = \mathbf{\Phi}\left(\left(\mathbf{\Phi}^\top\mathbf{g}\right) \odot \left(\mathbf{\Phi}^\top\mathbf{x}\right)\right) = \mathbf{\Phi}g(\mathbf{\Lambda})\mathbf{\Phi}^\top\mathbf{x} \tag{5}$$

where $g(\mathbf{\Lambda}) = \mathrm{diag}\,(\mathbf{g}\,(\lambda_1), \ldots, \mathbf{g}\,(\lambda_\mathbf{n}))$ is a function of the Fourier coefficients (can also be seen as the convolution kernel in graph convolution) and $\odot$ denotes the Hadamard product. In order to reduce the computational complexity of $\mathcal{O}\left(n^2\right)$ caused by calculating the eigenvalues and eigenvectors of $\mathbf{L}$, Chebyshev polynomial [17] can be utilized to fit the convolution kernel $g(\mathbf{\Lambda})$ as

$$g(\mathbf{\Lambda}) = \sum_{k=0}^{K-1} \beta_k T_k(\tilde{\mathbf{\Lambda}}) \tag{6}$$

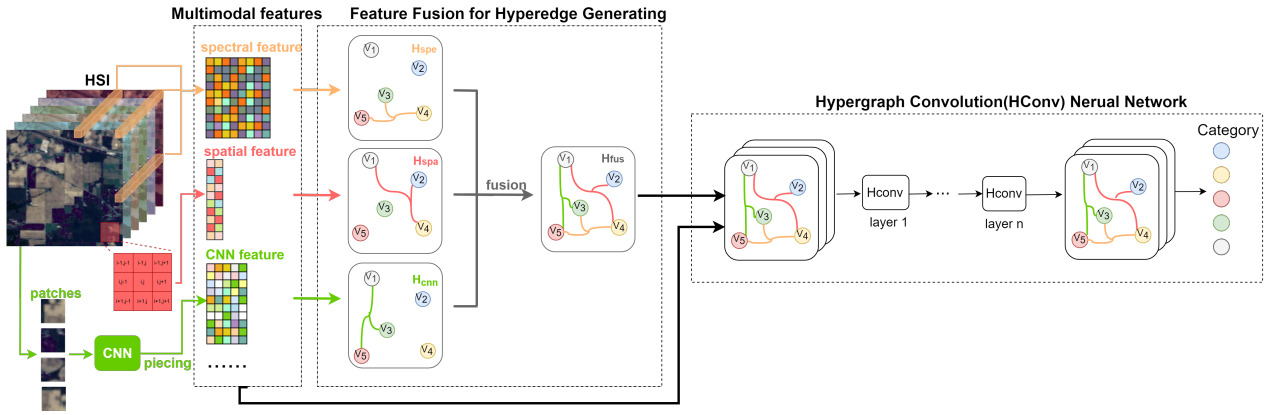


Fig. 2. Overview of our proposed F^{2}HNN. Taking the third fusion strategy as an example, this figure first shows the fusion of spectral features, spatial coordinate features and CNN features. Next, it shows the training process of the hypergraph neural network and the acquisition of classification results.

where $\tilde{\mathbf{\Lambda}}$ is the re-scaled $\mathbf{\Lambda}$ to ensure that the input of the Chebyshev polynomial is between $[1,-1]$, and T_k is the K order Chebyshev polynomial which can be computed by

$$T_k(x) = \cos(k \cdot \arccos(x)) \tag{7}$$

Substituting Equations 6, 7 into Equation 5, we can get

$$\mathbf{g} \star \mathbf{x} \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{\mathbf{L}})\mathbf{x} \tag{8}$$

where $\tilde{\mathbf{L}}$ is the re-scaled Laplacian $\tilde{\mathbf{L}} = \frac{2}{\lambda_{max}}\mathbf{L} - \mathbf{I}$. After reducing the computational complexity, we further set $K = 1$ and $\lambda_{max} \approx 2$ suggested by [22] and [33]. Thus, the convolution operation on hypergraph can be further simplified as

$$\mathbf{g} \star \mathbf{x} \approx \theta_0\mathbf{x} - \theta_1\mathbf{D}^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_\mathbf{e}^{-1}\mathbf{H}^\top\mathbf{D}_\mathbf{v}^{-1/2}\mathbf{x} \tag{9}$$

where θ_0 and θ_1 can be integrated by one parameter θ to avoid the overfitting, which is defined as

$$\begin{cases} \theta_1 = -\frac{1}{2}\theta \\ \theta_0 = \frac{1}{2}\theta\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2} \end{cases} \tag{10}$$

Then, the convolution operation on hypergraph is further derived as

$$\begin{aligned} \mathbf{g} \star \mathbf{x} &\approx \frac{1}{2}\theta\mathbf{D}_v^{-1/2}\mathbf{H}(\mathbf{W}+\mathbf{I})\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{x} \\ &\approx \theta\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{x} \end{aligned} \tag{11}$$

where $\mathbf{W}$ represents the weight matrix of the hyperedges, which is usually calculated in advance or directly initialized as an identity matrix.

Given a hypergraph data $\mathbf{X}^{n\times c_1}$ with n vertices and c_1 feature channels, the convolution operation on hypergraph can be formulated by

$$\mathbf{Y} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}\mathbf{\Theta} \tag{12}$$

where $\mathbf{W} = diag(w_1, w_2, ..., w_n)$ and $\mathbf{\Theta}^{c_1\times c_2}$ are the trainable parameters. $\mathbf{Y}^{n\times c_2}$ is the output after the convolution operation.

3) Hypergraph Convolution Neural Networks: The complete hypergraph convolutional layer is obtained by the hypergraph convolution operation described above plus the nonlinear activation function, which can be formulated as

$$\mathbf{X}^{(l+1)} = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\mathbf{\Theta}^{(l)}\right) \tag{13}$$

where $\mathbf{X}^{(l+1)}$ is the output of the lth layer and σ is the RELU function used for nonlinear activation.

B. Feature Fusion for Hyperedge Generating

We use the incidence matrix $\mathbf{H}$ to represent the topological structure of the hypergraph. For HSIs, the incidence matrix $\mathbf{H}$ cannot be obtained directly and needs to be generated from the image. GCN-based methods usually use single-modal features for graph construction and learning. This is because they utilize adjacency matrix $\mathbf{A}$ as input, which limits the number of edges. However, the fusion of multi-modal features, such as spatial features and spectral features [44]–[46], proved to be very effective in HSI classification tasks. Our proposed F^{2}HNN fuses multi-modal features to generate hyperedges, thereby needs to obtain the incidence matrix $\mathbf{H}$.

Given HSI data $\mathbf{Z}^{rows\times cols\times c}$, we treat each pixel as a vertice of the hypergraph and flatten the image to $\mathbf{X}^{n\times c}$, where $n = rows \times cols = |\mathcal{V}|$ represents the number of hypergraph vertices and c represents spectral dimensions. Then we use different feature extractors to extract features of different modalities $\mathbf{X}_i^{n\times c_i}$ from the original $\mathbf{X}^{n\times c}$.

For each vertex $v \in \mathcal{V}$ in $\mathbf{X}$, we select its k nearest neighbors to generate a hyperedge $e \in \mathcal{E}$, thus we obtain an incidence matrix $\mathbf{H}^{|\mathcal{V}|\times|\mathcal{E}|}(|\mathcal{V}| = |\mathcal{E}| = n)$:

$$h(i,j) = \begin{cases} e^{-\sigma\|\mathbf{x}_i - \mathbf{x}_j\|^2/\boldsymbol{mean}}, & \text{if } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{mean}$ represents the average of the Euclidean distance between all vertices $v \in \mathcal{V}$, and σ is an adjustable hyperparameter. The purpose of using $\boldsymbol{mean}$ in this equation is to normalize the distances of multi-modal fearures and to facilitate the adjustment of hyperparameters.

Given multi-modal features $[\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_m]$, for each $\mathbf{X}_i$, we use Equation 14 to calculate $\mathbf{H}_i$, and then we get

$[\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_m]$. The way we fuse the hyperedge obtained by multi-modal features is to concatenate these incidence matrices into $\mathbf{H}_{fuse} = [\mathbf{H}_1\ \mathbf{H}_2\ ...\mathbf{H}_m]$. Then the hypergraph convolution of Equation 12 becomes

$$\begin{aligned}\mathbf{Y} &= \mathbf{D}_v^{-1/2}\mathbf{H}_{fuse}\mathbf{W}_{fuse}\mathbf{D}_v^{-1/2}\mathbf{H}_{fuse}^T\mathbf{D}_e^{-1}\mathbf{X}\mathbf{\Theta} \\ &= \mathbf{D}_v^{-1/2}\begin{bmatrix}\mathbf{H}_1 & ... & \mathbf{H}_m\end{bmatrix}\begin{bmatrix}\mathbf{W}_1 & & \\ & ... & \\ & & \mathbf{W}_m\end{bmatrix}\mathbf{D}_e^{-1}\begin{bmatrix}\mathbf{H}_1^T \\ ... \\ \mathbf{H}_m^T\end{bmatrix} \\ &\quad \mathbf{D}_v^{-1/2}\mathbf{X}\mathbf{\Theta}\end{aligned} \tag{15}$$

If not considering the normalization matrices $\mathbf{D}_v$ and $\mathbf{D}_e$, Equation 15 can be transformed to

$$\begin{aligned}\mathbf{Y}^{'} &= \mathbf{H}_{fuse}\mathbf{W}_{fuse}\mathbf{H}_{fuse}^T\mathbf{X}\mathbf{\Theta} \\ &= \begin{bmatrix}\mathbf{H}_1 & ... & \mathbf{H}_2\end{bmatrix}\begin{bmatrix}\mathbf{W}_1 & & \\ & ... & \\ & & \mathbf{W}_m\end{bmatrix}\begin{bmatrix}\mathbf{H}_1 \\ ... \\ \mathbf{H}_m\end{bmatrix}\mathbf{X}\mathbf{\Theta} \\ &= \left(\mathbf{H}_1\mathbf{W}_1\mathbf{H}_1^T + \ ... \ + \mathbf{H}_m\mathbf{W}_m\mathbf{H}_m^T\right)\mathbf{X}\mathbf{\Theta}\end{aligned} \tag{16}$$

Equation 16 shows that using $\mathbf{H}_{fuse}$ for training in hypergraph convolution is equivalent to performing collaborative training on each subgraph $\mathbf{H}_i$ represented by each feature $\mathbf{X}_i$.

C. Different Fusion Strategies

After obtaining a network framework suitable for multi-modal feature fusion and training, how to obtain and select multi-modal features has become a matter of concern. In this section, we separately introduce three representative feature fusion strategies used in this paper, including the original unprocessed HSI data features and the features extracted by the pre-trained feature extractor.

1) Original spectral feature + Spatial feature: Spectral-spatial feature fusion is the most common and effective feature fusion strategy in HSI classification task. Consider the most general case, we first use the original unprocessed spectral feature and the spatial feature. The spatial feature $\mathbf{X}_{spatial}^{n\times 2}$ are obtained through pixel coordinates

$$\mathbf{X}_{spatial}[i] = [x(i), y(i)] \tag{17}$$

where $x(i)$ and $y(i)$ represent the horizontal and vertical coordinates of pixel i, respectively. Algorithm 1 summarizes the process of the proposed strategy.

2) Spectral feature extracted by multilayer perceptron + Spatial feature: In the task of hyperspectral classification, preprocessing of spectral features has proved to be necessary. Therefore, in the second strategy, we use a pre-trained multi-layer perceptron (MLP) as a feature extractor to preprocess the spectral features, and then fuse them with the spatial features obtained in Equation 17. The processing flow of this strategy is shown in Algorithm 2.

3) Spectral feature + Spatial feature + CNN feature: The CNN method in the HSI classification task divides a single HSI into different patches containing local information and inputs them into the network for training. This type of method effectively extracts the local information contained in the HSI patches. However, due to the existence of patches, CNN method ignores non-local information. Unlike CNN method, the graph constructed with spectral information discards the local information contained in Euclidean data and focuses on long-distance dependencies. Our third fusion strategy considers both the fusion of spectral-spatial features and the fusion of local and non-local information, and fuses the three features of spectral feature, coordinate-encoded spatial feature and CNN feature extracted by pre-trained model. Algorithm 3 indicates the process of the third strategy.

Algorithm 1 Pseudo code of the fusion strategy of "Original spectral feature + Spatial feature"

input Original HSI $\mathbf{X}_{ori}$, number of epochs $\mathcal{E}$, number of layers $\mathcal{L}$, number of neighbors k

Generate $\mathbf{X}_{spe}$ by flatting $\mathbf{X}_{ori}$
Generate $\mathbf{X}_{spa}$ according to Equation 17
$\mathbf{X} = [\mathbf{X}_{spe}\ \ \mathbf{X}_{spa}]$
Construct $\mathbf{H}_{spe}$ and $\mathbf{H}_{spa}$ according to Equation 14 and k
$\mathbf{H} = [\mathbf{H}_{spe}\ \ \mathbf{H}_{spa}]$
Calculate $\mathbf{D}_v$ and $\mathbf{D}_e$ from $\mathbf{H}$
Initialize $\mathbf{\Theta}$ and $\mathbf{W}$
for $i = 1\ to\ \mathcal{E}$ **do**
  for $l = 1\ to\ \mathcal{L}$ **do**
    $\mathbf{X}^l = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{l-1}\mathbf{\Theta}^{l-1}\right)$
  end for
  $\mathbf{X}_{pre} = SOFTMAX(\mathbf{X}^{\mathcal{L}})$
  Calculate loss and update parameters $\mathbf{\Theta}$ and $\mathbf{W}$ through gradient backpropagation
end for

output The predicted label of each pixel in HSI

Algorithm 2 Pseudo code of the fusion strategy of "Spectral feature extracted by multilayer perceptron + Spatial feature"

input Original HSI $\mathbf{X}_{ori}$, number of epochs $\mathcal{E}$, number of layers $\mathcal{L}$, number of neighbors k

Generate $\mathbf{X}_{spe}$ by flatting $\mathbf{X}_{ori}$
Use $\mathbf{X}_{spe}$ as input to pretrain a MLP, and obtain $\mathbf{X}_{mlp} = MLP(\mathbf{X}_{spe})$
Generate $\mathbf{X}_{spa}$ according to Equation 17
$\mathbf{X} = [\mathbf{X}_{mlp}\ \ \mathbf{X}_{spa}]$
Construct $\mathbf{H}_{mlp}$ and $\mathbf{H}_{spa}$ according to Equation 14 and k
$\mathbf{H} = [\mathbf{H}_{mlp}\ \ \mathbf{H}_{spa}]$
Calculate $\mathbf{D}_v$ and $\mathbf{D}_e$ from $\mathbf{H}$
Initialize $\mathbf{\Theta}$ and $\mathbf{W}$
for $i = 1\ to\ \mathcal{E}$ **do**
  for $l = 1\ to\ \mathcal{L}$ **do**
    $\mathbf{X}^l = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{l-1}\mathbf{\Theta}^{l-1}\right)$
  end for
  $\mathbf{X}_{pre} = SOFTMAX(\mathbf{X}^{\mathcal{L}})$
  Calculate loss and update parameters $\mathbf{\Theta}$ and $\mathbf{W}$ through gradient backpropagation
end for

output The predicted label of each pixel in HSI

Algorithm 3 Pseudo code of the fusion strategy of "Spectral feature + Spatial feature + CNN feature"

input Original HSI $\mathbf{X}_{ori}$, number of epochs $\mathcal{E}$, number of layers $\mathcal{L}$, number of neighbors k

Generate $\mathbf{X}_{spe}$ by flatting $\mathbf{X}_{ori}$
Use a set of patches $\mathcal{P}$ generated from $\mathbf{X}_{spe}$ as input to pretrain a CNN model
for each pixel j in HSI **do**
  Generate a 7×7 patch $\mathcal{P}_j$ with j as the center
  $x_j = CNN(\mathcal{P}_j)$
  Generate $\mathbf{X}_{cnn}$ by piecing each x_j
end for
Generate $\mathbf{X}_{spa}$ according to Equation 17
$\mathbf{X} = [\mathbf{X}_{spe} \ \ \mathbf{X}_{spa} \ \ \mathbf{X}_{cnn}]$
Construct $\mathbf{H}_{spe}$, $\mathbf{X}_{spa}$ and $\mathbf{H}_{cnn}$ according to Equation 14 and k
$\mathbf{H} = [\mathbf{H}_{spe} \ \ \mathbf{H}_{spa} \ \ \mathbf{X}_{cnn}]$
Calculate $\mathbf{D}_v$ and $\mathbf{D}_e$ from $\mathbf{H}$
Initialize $\mathbf{\Theta}$ and $\mathbf{W}$
for $i = 1$ *to* $\mathcal{E}$ **do**
  for $l = 1$ *to* $\mathcal{L}$ **do**
    $\mathbf{X}^l = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{l-1}\mathbf{\Theta}^{l-1}\right)$
  end for
  $\mathbf{X}_{pre} = SOFTMAX(\mathbf{X}^{\mathcal{L}})$
  Calculate loss and update parameters $\mathbf{\Theta}$ and $\mathbf{W}$ through gradient backpropagation
end for

output The predicted label of each pixel in HSI

IV. Experiment

A. Datasets

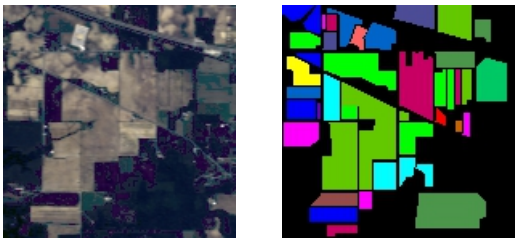
We select four representative datasets and perform comprehensive experiments to verify the effectiveness of our proposed method.

1) Indian Pines: The Indian Pines dataset was photographed in northwestern Indiana by Airborne Visible/Infrared Imaging Spectrom-eter (AVIRIS) sensor. The size of the data is 145 × 145 with 20m × 20m spatial resolution. The image contains 220 spectral bands covering 400 to 2500nm, of which 20 water absorption and noisy bands are removed. A total of 16 classes of land-covers are labeled for classification. Table I details the total number of samples for each category and the number of samples used for training. Figure 3 shows the pseudo-color image and ground-truth map of Indian Pines.

2) Kennedy Space Center: The Kennedy Space Center (KSC) dataset was collected in Florida by the AVIRIS sensor. It contains 224 spectral bands sampled from 400 to 2500nm and 614 × 512 pixels with a spatial resolution of 18m × 18m. The selected 176 bands and 13 labeled classes of land-covers are used for classification. Table III lists the total number of samples for each category and the number of samples used for training. The pseudo-color image and ground-truth map of Kennedy Space Center are shown in Figure 4.

3) Botswana: The Botswana dataset was acquired by NASA's EO-1 satellite over the Okavango Delta, Botswana. The sensor on EO-1 obtained a 1476 × 256 data with 30m pixel resolution and 242 spectral bands covering a range of 400-2500nm. It includes 14 land-cover classes. Since the atmospheric and water absorption bands is invalid, 145 out of the 242 bands are selected. Training and testing samples for Botswana are shown in Table III. Figure 5 is the pseudo-color image and ground-truth map of Botswana.

4) Pavia University: The Pavia University dataset is acquired by ROSIS sensors. After removing noisy bands, it has a total of 103 effective bands, covering the electromagnetic spectrum from 400 to 860nm. The dataset contains 9 land-cover classes and the size is 610×340. The spatial resolution of the dataset is 1.3m × 1.3m. Table IV lists the number of training samples. Figure 6 is the pseudo-color image and ground-truth map of Pavia University.



(a) (b)

Fig. 3. Visualization of India Pines dataset. (a) Pseudo-color map. (b) Ground truth map.

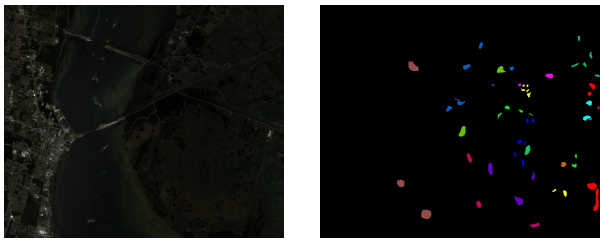
TABLE I
THE NUMBER OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS OF LAND-COVER IN INDIAN PINES.

Class No.	Class Color	Class Name	Training	Testing
1		Alfalfa	15	31
2		Corn Notill	50	1378
3		Corn Mintill	50	780
4		Corn	50	187
5		Grass Pasture	50	433
6		Grass Trees	50	680
7		Grass Pasture Mowed	15	13
8		Hay Windrowed	50	428
9		Oats	15	5
10		Soybean Notill	50	922
11		Soybean Mintill	50	2405
12		Soybean Clean	50	543
13		Wheat	50	155
14		Woods	50	1215
15		Buildings Grass Trees Drives	50	336
16		Stone Steel Towers	50	43
		Total	695	9554

B. Experimental Settings

In this section, we first introduce the detailed structure of our proposed F^{2}HNN and the parameter settings of the experiments. Then we show all the baseline models used for comparison and the settings of their parameters.

1) Detailed supplement of proposed F^{2}HNN : The three feature fusion strategies used in the experiments require two pre-trained models, i.e. the multilayer perceptron model adopted in the second strategy and the CNN model used in the third

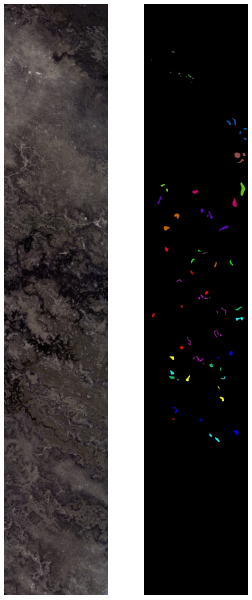


(a) (b)

Fig. 4. Visualization of KSC dataset. (a) Pseudo-color map. (b) Ground truth map.

TABLE II
THE NUMBER OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS OF LAND-COVER IN KENNEDY SPACE CENTER

Class No.	Class Color	Class Name	Training	Testing
1		Srub	30	728
2		Willow swamp	30	220
3		CP hammock	30	232
4		Slash pine	30	228
5		Oak/Broadleaf	30	146
6		Hardwood	30	207
7		Swamp	30	96
8		Graminoid	30	393
9		Spartina marsh	30	469
10		Cattail marsh	30	365
11		Salt marsh	30	378
12		Mud flats	30	454
13		Water	30	836
		Total	390	4752

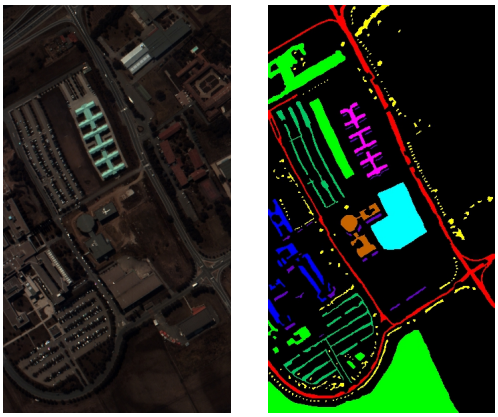


(a) (b)

Fig. 5. Visualization of Botswana dataset. (a) Pseudo-color map. (b) Ground truth map.

TABLE III
THE NUMBER OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS OF LAND-COVER IN BOTSWANA

Class No.	Class Color	Class Name	Training	Testing
1		Water	10	270
2		Hippo grass	10	101
3		Floodplain grasses 1	5	251
4		Floodplain grasses 2	5	215
5		Reeds	5	269
6		Riparian	5	269
7		Firescar	5	259
8		Island interior	5	203
9		Acacia woodlands	5	314
10		Acacia shrublands	5	248
11		Acacia grasslands	5	305
12		Short mopane	5	181
13		Mixed mopane	5	268
14		Exposed soils	5	95
		Total	140	3248



(a) (b)

Fig. 6. Visualization of Pavia University dataset. (a) Pseudo-color map. (b) Ground truth map.

TABLE IV
THE NUMBER OF TRAINING SAMPLES AND TEST SAMPLES FOR EACH CLASS OF LAND-COVER IN PAVIA UNIVERSITY

Class No.	Class Color	Class Name	Training	Testing
1		Asphalt	30	6601
2		Meadows	30	18619
3		Gravel	30	2069
4		Trees	30	3034
5		Painted metal sheets	30	1315
6		Bare soil	30	4999
7		Bitumen	30	1300
8		Self-blocking bricks	30	3652
9		Shadows	30	917
		Total	270	42566

strategy. We design a three-layer multilayer perceptron and a four-layer 3D CNN as the pre-trained model. The detailed parameters are shown in Table V. The training samples of the pre-trained model are consistent with the training samples in the subsequent F^{2}HNN. It is worth noting that for the Pavia University dataset with a large image size, we averagely cut it into four parts for classification according to the number of labeled samples included.

The network is implemented using the PyTorch framework, the experiment is set to full gradient descent, and Adam [47] is used for optimization with weight decay set to 0.0005. The number of total epoch is chosen 200, and the learning rate is set to be dynamically adjustable. The optimal initial learning rate 0.01 is chosen from $\{0.0001, 0.001, 0.01, 0.1, 1\}$, and the learning rate drops to one-half of the previous one after 30 epochs.

TABLE V
THE DETAILED PARAMETER SETTINGS OF THE ENTIRE F^{2}HNN, INCLUDING THE PRE-TRAINING NETWORK AND HGNN SETTINGS. THE RELU FUNCTION IS THE ACTIVATION FUNCTION WE USE, FC STANDS FOR FULLY CONNECTED LAYER, CONV REPRESENTS 3D CONVOLUTIONAL LAYER, POOL INDICATES 3D POOLING LAYER, BN MEANS BATCH NORMALIZATION LAYER, AND HCONV IS THE HYPERGRAPH CONVOLUTIONAL LAYER

		MLP	3D CNN
F^{2}HNN	Pre-trained model	ReLU	patch_size=7×7
		512×1024 FC	3×3×3 Conv
		ReLU	3×1×1 Pool
		1024×64 FC	3×3×3 Conv
		Softmax	3×1×1 Pool
			3×1×1 Conv
			2×1×1 Conv
			Softmax
	Follow-up model	HGNN	
		Compute $\mathbf{W}$	
		BN	
		C_{in}×128 Hconv	
		BN	
		ReLU	
		128×C_{nClass} Hconv	
		Softmax	

2) Settings of baseline methods: In order to verify the effectiveness of F^{2}HNN, we select several representative baseline methods, including support vector machine (SVM) with RBF kernel, 2D CNN, 3D CNN [48], GCN and FuNet-M [20]. At the same time, two pre-trained networks and three networks generated by different feature fusion strategies are also used for comparison experiments.

For convenience, we abbreviate the networks generated by the three different strategies as F^{2}HNN$_{SS}$ (Spectral feature + Spatial feature), F^{2}HNN$_{MS}$ (Multilayer perceptron feature + Spatial feature) and F^{2}HNN$_{SSC}$ (Spectral feature + Spatial feature + CNN feature).

C. Classification Results

Table VI to IX record the overall accuracy (OA), average accuracy (AA) and kappa coefficients of our own method and all baseline methods. Figure 7 to 10 are visualizations of all results.

For different datasets, the classification performance of the selected methods are different. Among all baseline methods, 3DCNN achieves the best overall performance. This is due to the powerful feature extraction capabilities of deep learning and the ability of the 3DCNN network to simultaneously learn spatial and spectral modal features. Conventional methods based on SVM and MLP can also obtain an acceptable classification performance, but their performance is not better compared with the methods based on deep learning. As for the baseline method based on graph neural network, GCN and FuNet-M have completely different performance on different datasets. For the Indian Pines dataset, GCN gets the worst classification accuracy, and FuNet's performance is only comparable to that of SVM. However, on the KSC, Botswana and Pavia University datasets, the performance of GCN exceeds that of SVM and is basically the same as that of 2DCNN, and FuNet-M achieves the highest accuracy of all baseline methods on these three datasets.

A reasonable inference is that the two methods based on graph neural networks both use spectral features to construct graph structures, that is, pay more attention to the long-distance dependence of HSIs. Comparing the distribution of the labeled samples in the four datasets, it can be found that the samples in the Indian Pines dataset contain more local information, which is contained in the spatial features of HSI and can be well extracted by CNN. In contrast, the distribution of labeled samples in the other three datasets is more discrete, and the long-distance dependence becomes more influential.

Remarkably, the proposed F^{2}HNN surpassed all baseline methods on the four datasets and achieved the highest OA, AA and kappa coefficient. On the Indian Pines dataset, F^{2}HNN can achieve 91.97% OA using only a basic spectral-spatial feature fusion strategy, which is 4.31 % higher than the highest one in the baseline methods. Moreover, the classification performance of the three strategies we designed is incremental. F^{2}HNN$_{MS}$ uses MLP for preprocessing, which removes the redundancy of spectral information, and performs better than F^{2}HNN$_{SS}$. F^{2}HNN$_{CSS}$ fuses the features of the three modalities and introduces CNN features that express local information, achieving the best performance of the three strategies.

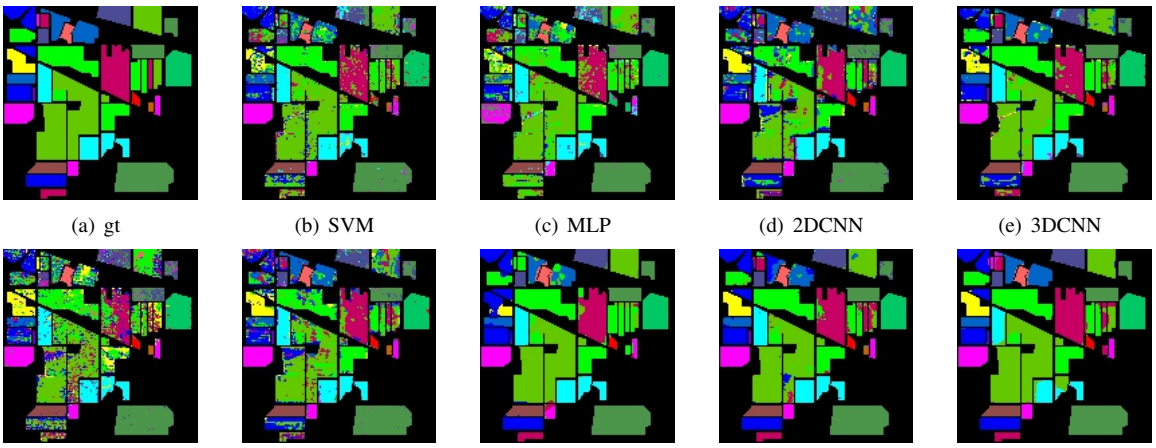
D. Hyperparameter Analysis

In this section, we utilize four datasets to quantitatively analyze the hyperparameters k and σ involved in F^{2}HNN. Theoretically, different k and σ can be selected for the feature of different modal, but in order to verify the robustness of the method, the k and σ of different modal features should be kept consistent as much as possible under the premise of little impact on the accuracy.

1) Analysis of k: Figure 11 is the analysis of the hyperparameter k. When analyzing k, σ is fixed to the optimal value shown in Table X. In the experiment, we select the same k value for the features of different modal. The optimal k value of each dataset is also shown in Table X.

TABLE VI
PER-CLASS ACCURACY, OVERALL ACCURACY(OA), AVERAGE ACCURACY(AA), AND KAPPA COEFFICIENT ACQUIRED BY DIFFERENT METHOD ON INDIAN PINES DATASET

Class No.	SVM	MLP	2DCNN	3DCNN [48]	GCN	FuNet-M [20]	F^2HNN_{SS}	F^2HNN_{MS}	F^2HNN_{SSC}
1	48.81	41.23	**100.00**	88.45	91.30	95.65	95.65	**100.00**	97.83
2	72.53	67.52	65.97	87.08	53.64	67.51	**96.85**	93.84	95.59
3	67.15	58.57	71.20	80.03	53.01	63.98	99.40	**99.88**	98.92
4	51.70	49.52	93.24	75.31	87.77	91.98	34.18	98.31	**100.00**
5	86.93	75.18	93.78	90.15	90.89	94.00	87.37	96.27	**100.00**
6	90.65	86.21	89.86	96.34	87.95	92.60	99.04	**99.59**	76.44
7	60.91	45.72	89.28	71.79	85.71	**92.86**	55.79	82.14	81.70
8	94.31	91.73	99.16	97.78	97.07	98.54	**100.00**	**100.00**	**100.00**
9	40.02	49.91	**100.00**	64.31	**100.00**	**100.00**	37.00	**100.00**	50.00
10	71.60	69.88	75.10	86.74	53.81	73.25	79.12	94.34	**96.50**
11	77.73	72.36	61.26	87.12	54.99	61.26	94.46	91.12	**98.33**
12	60.83	45.02	79.25	83.02	38.28	75.72	74.03	71.16	**95.79**
13	95.87	90.50	**100.00**	98.10	98.05	**100.00**	98.54	**100.00**	**100.00**
14	92.74	90.93	91.62	96.83	84.58	94.78	**100.00**	97.79	99.92
15	63.66	53.47	82.21	78.05	65.80	82.90	97.93	98.19	**100.00**
16	91.23	88.55	**100.00**	98.90	97.85	**100.00**	**100.00**	**100.00**	**100.00**
OA(%)	77.96	72.13	77.13	87.66	65.97	76.94	91.97	94.17	**96.25**
AA(%)	72.92	67.27	86.99	86.25	77.54	86.56	84.34	**95.16**	93.19
Kappa	0.7475	0.6781	0.7652	0.8603	0.6184	0.7431	0.9003	0.9322	**0.9514**



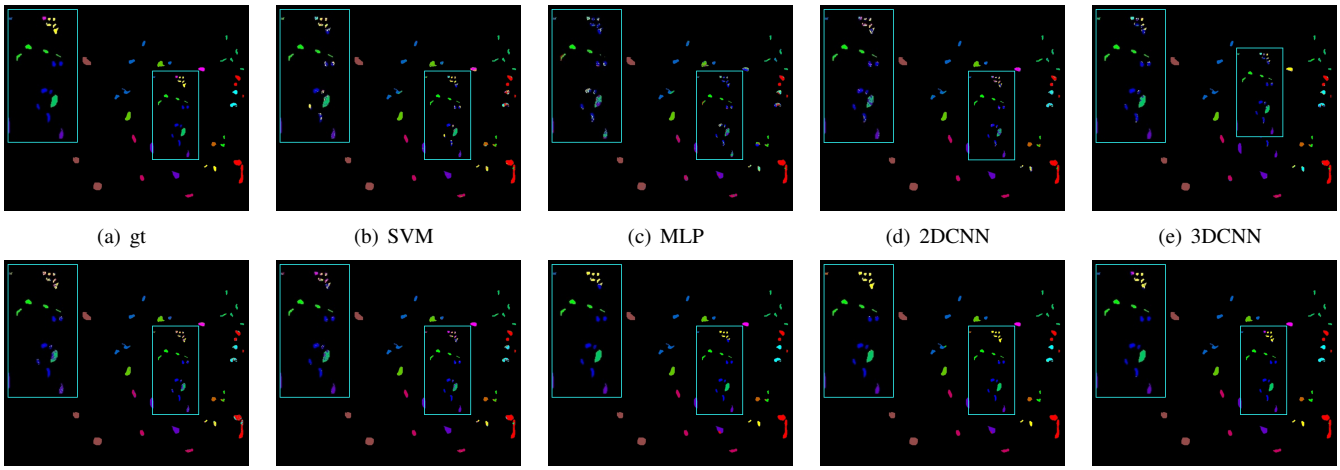
(a) gt (b) SVM (c) MLP (d) 2DCNN (e) 3DCNN

(f) GCN (g) FuNet-M (h) F^2HNN_{SS} (i) F^2HNN_{MS} (j) F^2HNN_{CSS}

Fig. 7. Visualization of the classification results of different methods on Indian Pines dataset. (a) gt. (b) SVM. (c) MLP. (d) 2DCNN. (e) 3DCNN. (f) GCN. (g) FuNet-M. (h) F^2HNN_{SS}. (i) F^2HNN_{MS}. (j) F^2HNN_{CSS}.

TABLE VII
PER-CLASS ACCURACY, OVERALL ACCURACY(OA), AVERAGE ACCURACY(AA), AND KAPPA COEFFICIENT ACQUIRED BY DIFFERENT METHOD ON KENNEDY SPACE CNETER DATASET

Class No.	SVM	MLP	2DCNN	3DCNN [48]	GCN	FuNet-M [20]	F^2HNN_{SS}	F^2HNN_{MS}	F^2HNN_{SSC}
1	90.17	87.91	95.93	95.81	86.33	95.40	**100.00**	**100.00**	99.34
2	86.68	80.81	81.89	83.82	84.36	86.00	94.24	**100.00**	**100.00**
3	69.23	47.32	83.20	78.79	92.19	93.36	**100.00**	98.43	99.61
4	52.96	26.59	48.41	33.00	68.65	55.56	85.32	**100.00**	95.63
5	61.71	55.67	76.40	43.31	75.16	91.93	81.99	81.99	**94.41**
6	40.8	38.94	82.10	68.77	78.17	74.67	**100.00**	97.82	99.13
7	80.81	38.25	**100.00**	89.53	99.04	**100.00**	**100.00**	**100.00**	**100.00**
8	84.53	66.89	77.26	89.74	81.90	87.47	**100.00**	99.30	**100.00**
9	92.87	81.03	98.65	96.91	96.73	99.23	93.27	98.08	**100.00**
10	96.91	89.01	96.78	97.80	90.10	96.53	**100.00**	**100.00**	**100.00**
11	94.79	94.36	99.52	98.71	97.37	98.57	**100.00**	**100.00**	99.76
12	89.84	82.19	93.04	96.48	90.06	90.85	**100.00**	93.44	**100.00**
13	99.93	98.67	**100.00**	**100.00**	99.78	99.89	**100.00**	**100.00**	**100.00**
OA(%)	86.74	77.57	90.75	88.81	89.83	92.43	97.79	98.39	**99.44**
AA(%)	80.14	68.28	87.17	82.51	87.68	89.96	96.52	97.62	**99.07**
Kappa	0.8521	0.7493	0.8738	0.8753	0.8815	0.9058	0.9239	0.9751	**0.9832**



(a) gt (b) SVM (c) MLP (d) 2DCNN (e) 3DCNN

(f) GCN (g) FuNet-M (h) F^{2}HNN$_{SS}$ (i) F^{2}HNN$_{MS}$ (j) F^{2}HNN$_{CSS}$

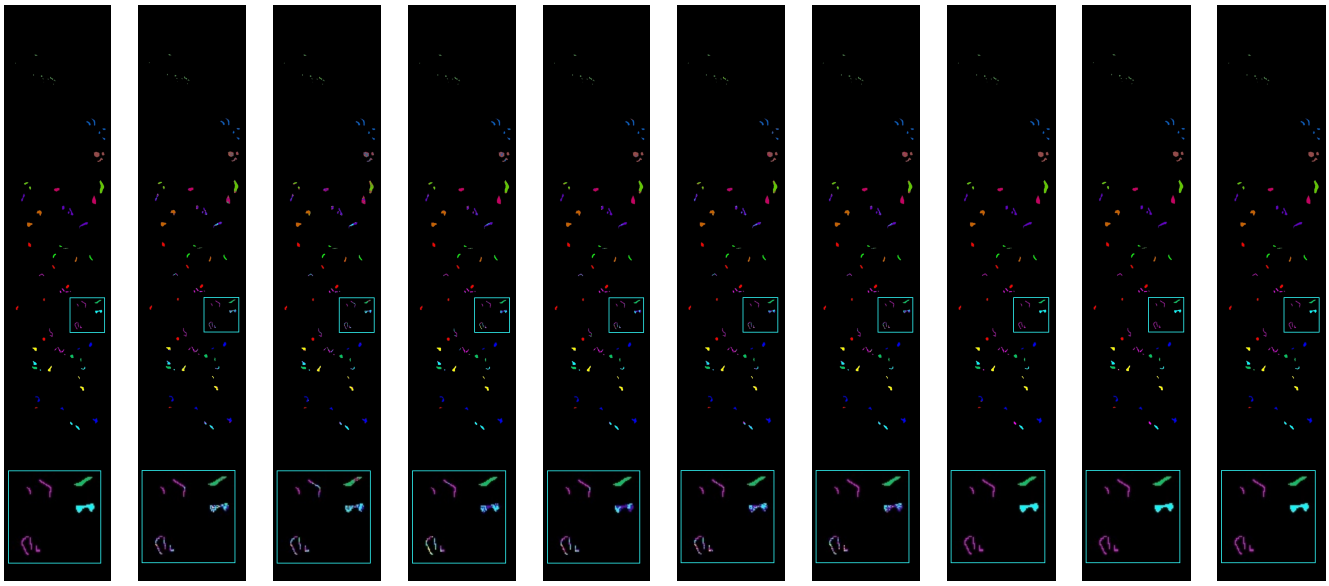
Fig. 8. Visualization of the classification results of different methods on KSC dataset. (a) gt. (b) SVM. (c) MLP. (d) 2DCNN. (e) 3DCNN. (f) GCN. (g) FuNet-M. (h) F^{2}HNN$_{SS}$. (i) F^{2}HNN$_{MS}$. (j) F^{2}HNN$_{CSS}$.

TABLE VIII
PER-CLASS ACCURACY, OVERALL ACCURACY(OA), AVERAGE ACCURACY(AA), AND KAPPA COEFFICIENT ACQUIRED BY DIFFERENT METHOD ON BOTSWANA DATASET

Class No.	SVM	MLP	2DCNN	3DCNN [48]	GCN	FuNet-M [20]	F^{2}HNN$_{SS}$	F^{2}HNN$_{MS}$	F^{2}HNN$_{SSC}$
1	99.81	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
2	91.83	97.03	98.02	96.80	98.02	**100.00**	**100.00**	**100.00**	**100.00**
3	91.47	94.02	**100.00**	98.61	98.01	99.60	98.01	**100.00**	**100.00**
4	87.40	88.37	**100.00**	96.95	97.67	99.07	**100.00**	**100.00**	**100.00**
5	82.55	69.52	74.35	87.28	80.67	84.01	**100.00**	**100.00**	**100.00**
6	65.63	75.09	79.93	79.03	65.43	75.09	85.13	85.13	**97.03**
7	97.49	91.12	98.46	99.63	96.14	96.91	**100.00**	**100.00**	**100.00**
8	88.80	92.12	99.01	99.23	98.03	98.52	**100.00**	**100.00**	**100.00**
9	79.64	79.30	82.48	90.10	80.25	87.26	**100.00**	**100.00**	**100.00**
10	79.48	74.60	99.60	93.04	94.76	99.19	**100.00**	**100.00**	**100.00**
11	89.91	73.77	96.07	94.77	86.89	90.49	**100.00**	**100.00**	**100.00**
12	92.44	97.79	98.34	93.86	86.74	98.34	**100.00**	**100.00**	**100.00**
13	84.86	66.42	83.21	95.79	91.42	92.16	**100.00**	**100.00**	**100.00**
14	96.52	86.32	98.95	97.62	82.11	97.89	93.68	**100.00**	93.68
OA(%)	86.81	83.19	92.36	94.02	89.22	93.20	98.43	98.77	**99.57**
AA(%)	85.72	84.68	93.46	94.48	89.72	94.18	98.34	98.94	**99.34**
Kappa	0.8521	0.8173	0.8996	0.9353	0.8745	0.9189	0.9713	0.9804	**0.9951**

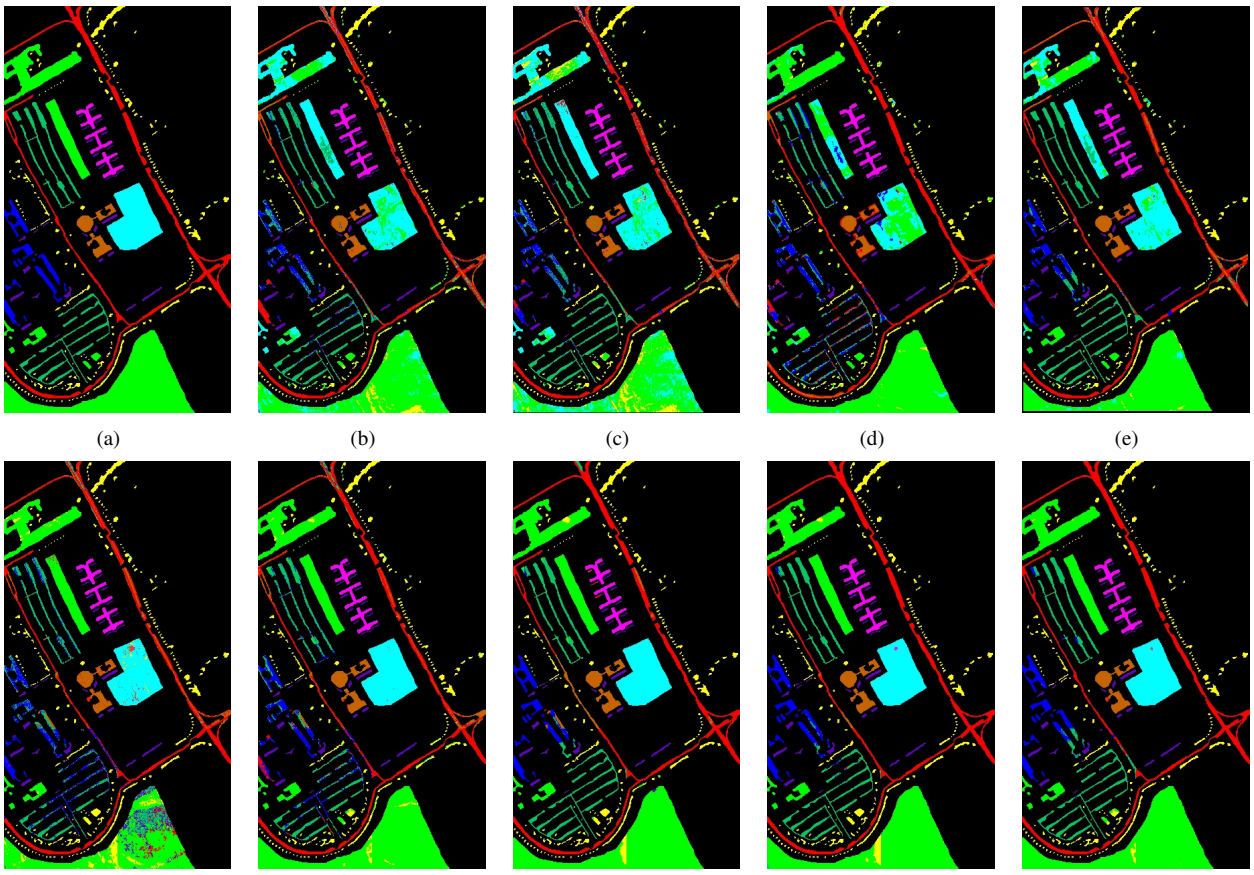
TABLE IX
PER-CLASS ACCURACY, OVERALL ACCURACY(OA), AVERAGE ACCURACY(AA), AND KAPPA COEFFICIENT ACQUIRED BY DIFFERENT METHOD ON PAVIA UNIVERSITY DATASET

Class No.	SVM	MLP	2DCNN	3DCNN [48]	GCN	FuNet-M [20]	F^{2}HNN$_{SS}$	F^{2}HNN$_{MS}$	F^{2}HNN$_{SSC}$
1	66.26	70.38	86.65	74.56	78.43	81.47	90.35	**98.08**	98.05
2	66.66	52.34	91.25	74.55	77.82	97.49	96.92	96.50	**98.73**
3	56.93	62.79	64.84	77.17	68.13	66.84	85.95	**95.33**	87.47
4	83.65	90.01	90.27	89.26	96.93	91.51	97.68	**99.15**	98.60
5	99.03	97.03	99.93	99.55	99.70	98.66	99.93	99.55	**100.00**
6	76.91	81.39	48.46	87.31	93.96	99.88	**100.00**	99.60	99.74
7	93.76	89.32	72.48	92.63	92.26	96.84	**100.00**	99.70	98.20
8	85.55	90.17	56.52	97.80	67.22	72.79	96.96	**97.39**	97.07
9	99.88	**99.89**	97.25	99.37	**99.89**	99.79	94.09	99.79	98.63
OA(%)	72.77	68.62	80.98	81.13	81.42	91.30	95.91	97.59	**98.06**
AA(%)	80.96	81.48	78.63	88.02	86.04	89.47	95.76	**98.34**	97.39
Kappa	0.6573	0.6169	0.7455	0.7615	0.7655	0.8852	0.9462	0.9683	**0.9744**



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 9. Visualization of the classification results of different methods on Botswana dataset. (a) gt. (b) SVM. (c) MLP. (d) 2DCNN. (e) 3DCNN. (f) GCN. (g) FuNet-M. (h) F^{2}HNN$_{SS}$. (i) F^{2}HNN$_{MS}$. (j) F^{2}HNN$_{CSS}$.



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

(f) (g) (h) (i) (j)

Fig. 10. Visualization of the classification results of different methods on Pavia University dataset. (a) gt. (b) SVM. (c) MLP. (d) 2DCNN. (e) 3DCNN. (f) GCN. (g) FuNet-M. (h) F^{2}HNN$_{SS}$. (i) F^{2}HNN$_{MS}$. (j) F^{2}HNN$_{CSS}$.

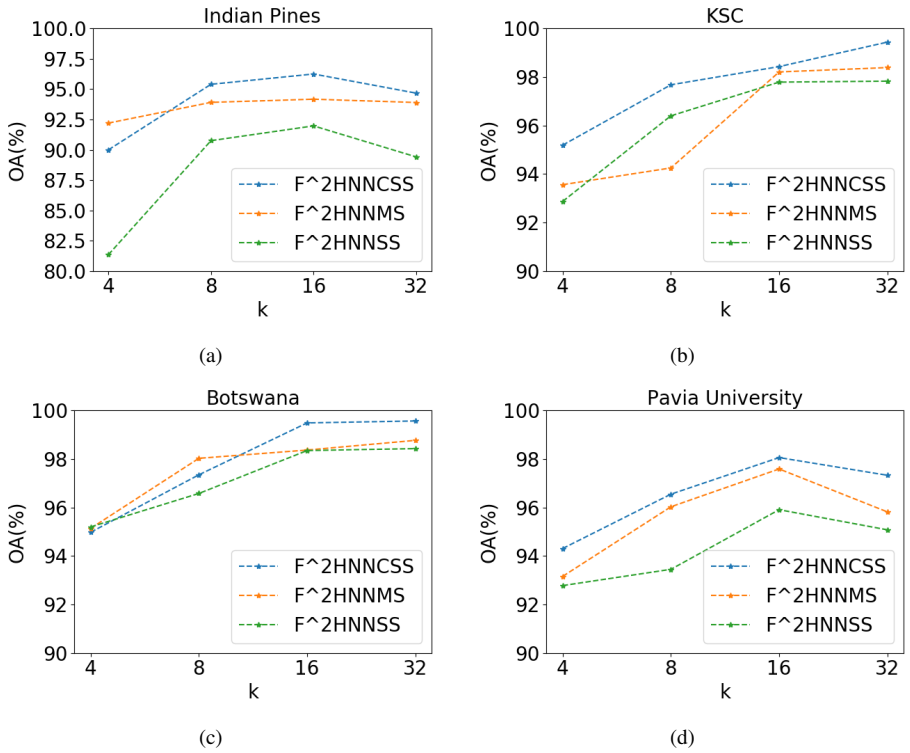


Fig. 11. Analysis of hyperparameter k on four datasets. (a) Indian Pines. (b) KSC. (c) Botswana. (d)Pavia University.

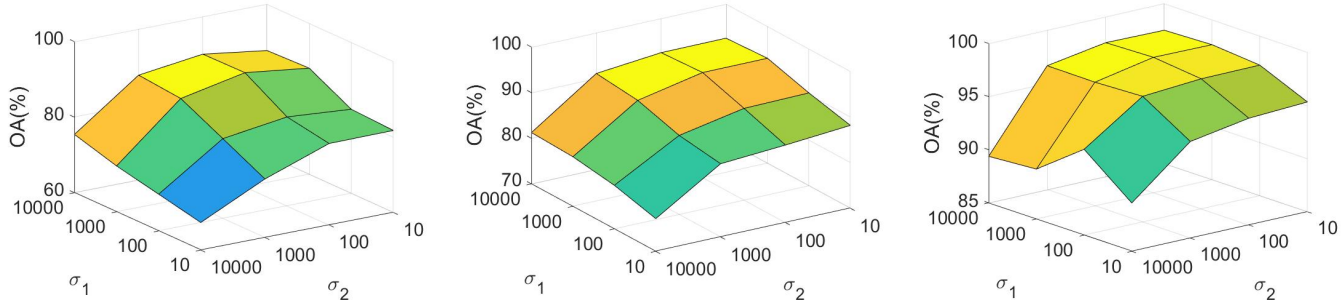


Fig. 12. Analysis of the hyperparameter σ of the three feature fusion strategies on the Indian Pines dataset. σ_1, σ_2, σ_3 and σ_4 represent the σ of spectral features, spatial features, MLP features and CNN features respectively. (a) $\text{F}^2\text{HNN}_{SS}$. (b) $\text{F}^2\text{HNN}_{MS}$. (c) $\text{F}^2\text{HNN}_{CSS}$

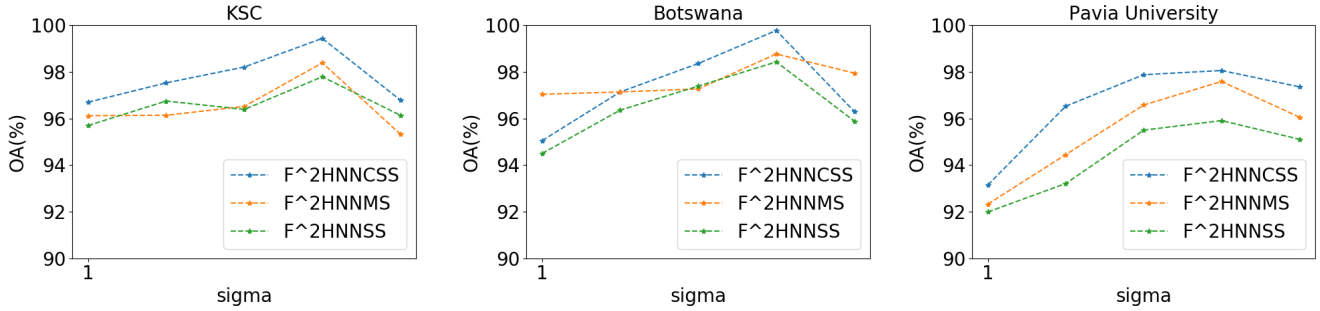


Fig. 13. Analysis of the hyperparameter σ of the three feature fusion strategies on the KSC, Botawana and Pavia University datasets. Here, we set $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = \sigma$. (a) KSC. (b) Botswana. (c) Pavia University.

2) Analysis of σ: In our three strategies, there are a total of 4 different modal features, namely original spectral feature, spatial feature, MLP feature, and CNN feature, and we denote the corresponding σ as σ_1, σ_2, σ_3, σ_4 respectively.

For the Indian Pines dataset, experiments show that it is sensitive to hyperparameter σ, so we selected different σ for different modal of features for analysis. Notably, since the impact of σ_1 on classification performance has been fully analyzed in F^{2}HNN$_{SS}$, we set σ_1 to a fixed value when analyzing the third strategy F^{2}HNN$_{CSS}$. For the KSC, Botswana and Pavia University datasets, we set $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4$, and proved through experiments that it is relatively insensitive to the change of σ. Figure 10 to 11 shows the influence of σ on the four datasets.

By analyzing Figure 12 to 13, we find that Indian Pines requires a relatively low σ for spatial features compared to the other three datasets. The possible reason is that compared to the other three datasets, Indian Pines has a more compact spatial distribution and more obvious geometric features.

The optimal σ values of different datasets selected for comparison with the baseline methods are shown in Table X.

TABLE X
SELECTION OF k AND σ ON FOUR DATASETS

Name	k	σ_1	σ_2	σ_3	σ_4
Indian Pines	16	1000	100	1000	1000
KSC	32	1000	1000	1000	1000
Botswana	32	1000	1000	1000	1000
Pavia University	16	1000	1000	1000	1000

E. Classification Performance using Limited Training Samples

In this experiment, we explore the classification performance of different methods with limited training samples. For the Indian Pines the Pavia University datasets, we select 5 to 25 training samples for each class. For the KSC dataset, the number of training samples is set to 2 to 10 per class. For the Botswana dataset, 1 to 5 samples are selected for each class. By analyzing the experimental results in Figure 14, we can find that the performance of F^{2}HNN far exceeds other compared methods. With extremely limited training samples, the three strategies of F^{2}HNN are still able to achieve considerable classification accuracy.

V. CONCLUSION

In this paper, we have proposed a novel F2HNN network for HSI classification. The F^{2}HNN network uses HGNN as framework, thus can dynamically update the hyperedge weights during the network training process. To some extent, this solves the problem of excessive dependence on the construction of adjacency matrix in the current GCN methods. Furthermore, F^{2}HNN is also very conducive to the fusion of multi-modal features, and we then designed three feature fusion strategies for HSI data. Sufficient experiments on four widely used datasets prove that F^{2}HNN surpasses all comparison methods and achieves state-of-the-art performance.

In the future, we plan to design more feature fusion strategies, and we will try to explore the connections and differences between features of different modal on this basis.

REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[2] J. A. Richards and J. Richards, *Remote sensing digital image analysis*. Springer, 1999, vol. 3.

[3] B. M. Shahshahani and D. A. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 5, pp. 1087–1095, 1994.

[4] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by svm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2297–2307, 2010.

[5] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[6] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 12–16, 2002.

[7] J. Xia, L. Bombrun, T. Adal, Y. Berthoumieu, and C. Germain, "Spectralspatial classification of hyperspectral images using ica and edge-preserving filter via an ensemble strategy," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4971–4982, 2016.

[8] Y. Fang, H. Li, Y. Ma, K. Liang, Y. Hu, S. Zhang, and H. Wang, "Dimensionality reduction of hyperspectral images based on robust spatial information using locally linear embedding," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1712–1716, 2014.

[9] X. Kang, S. Li, and J. A. Benediktsson, "Spectralspatial hyperspectral image classification with edge-preserving filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2666–2677, 2014.

[10] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 3791–3808, 2020.

[11] D. Lunga, 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 Processing Magazine*, vol. 31, no. 1, pp. 55–66, Jan 2014.

[12] H. Yu, L. Gao, W. Liao, B. Zhang, L. Zhuang, M. Song, and J. Chanussot, "Global spatial and local spectral similarity-based manifold learning group sparse representation for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3043–3056, 2019.

[13] M. M. Crawford, L. Ma, and W. Kim, "Exploring nonlinear manifold learning for classification of hyperspectral data," in *Optical Remote Sensing*. Springer, 2011, pp. 207–234.

[14] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015.

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

[16] 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 Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct 2016.

[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

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

[19] F. F. Shahraki and S. Prasad, "Graph convolutional neural networks for hyperspectral data classification," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov 2018, pp. 968–972.

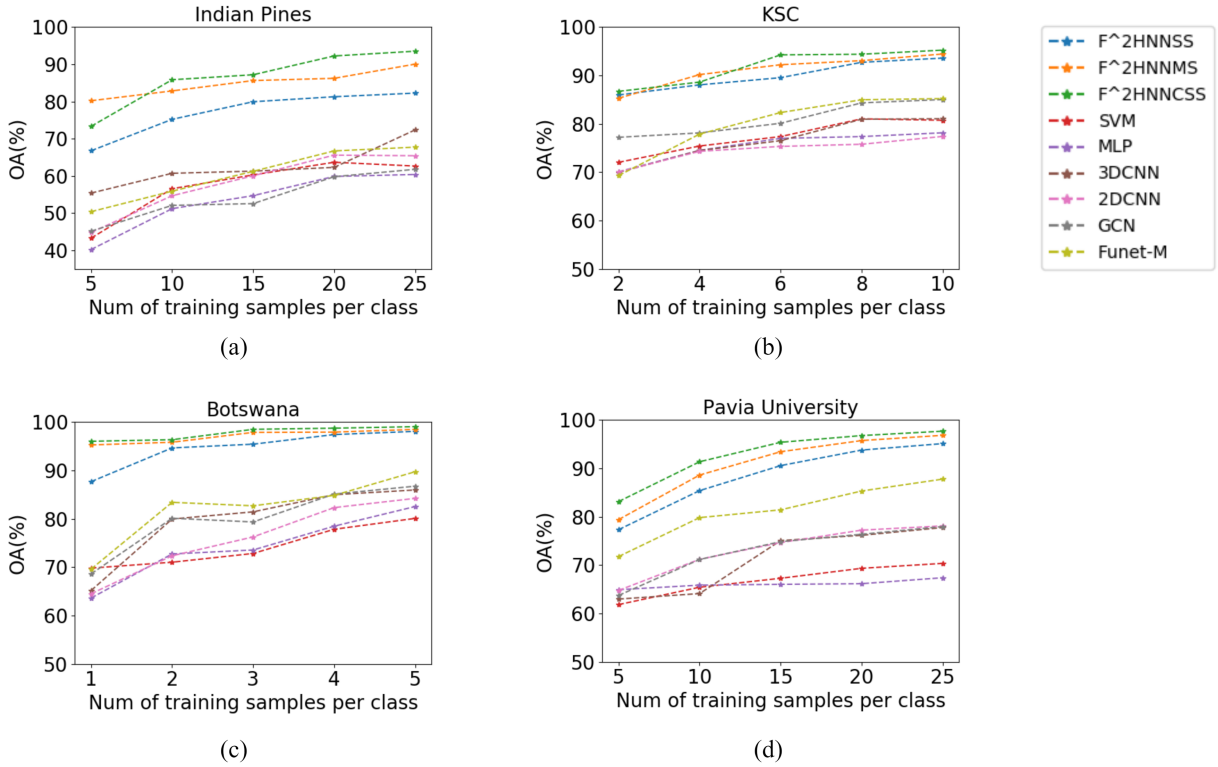


Fig. 14. Analysis of limited training samples classification performance on four datasets. (a) Indian Pines. (b) KSC. (c) Botswana. (d) Pavia University.

[20] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2020.

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

[22] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3558–3565.

[23] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," *Advances in neural information processing systems*, vol. 19, pp. 1601–1608, 2006.

[24] Q. Fang, J. Sang, C. Xu, and Y. Rui, "Topic-sensitive influencer mining in interest-based social media networks via hypergraph learning," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 796–812, 2014.

[25] D. Li, Z. Xu, S. Li, and X. Sun, "Link prediction in social networks based on hypergraph," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 41–42.

[26] I. Chien, C.-Y. Lin, and I.-H. Wang, "Community detection in hypergraphs: Optimal statistical limit and efficient algorithms," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 871–879.

[27] Y. Wang, L. Zhu, X. Qian, and J. Han, "Joint hypergraph learning for tag-based image retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4437–4451, 2018.

[28] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3262–3272, 2012.

[29] L. An, X. Chen, S. Yang, and X. Li, "Person re-identification by multi-hypergraph fusion," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2763–2774, 2016.

[30] Z. Zhang, H. Lin, X. Zhao, R. Ji, and Y. Gao, "Inductive multi-hypergraph learning and its application on view-based 3d object classification," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5957–5968, 2018.

[31] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 1738–1745.

[32] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–14.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–12.

[35] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–19.

[36] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao, "Dynamic hypergraph neural networks." in *IJCAI*, 2019, pp. 2635–2641.

[37] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, vol. 110, p. 107637, 2021.

[38] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3044–3054, 2007.

[39] Y. Gao, R. Ji, P. Cui, Q. Dai, and G. Hua, "Hyperspectral image classification through bilayer graph-based learning," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2769–2778, 2014.

[40] S. Wan, C. Gong, P. Zhong, S. Pan, G. Li, and J. Yang, "Hyperspectral image classification with context-aware dynamic graph convolutional network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 597–612, 2020.

[41] X. He, Y. Chen, and P. Ghamisi, "Dual graph convolutional network for hyperspectral image classification with limited training samples," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[42] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[43] D. Spielman, "Spectral graph theory," in *Combinatorial scientific computing*. Citeseer, 2012, no. 18.

[44] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2020.

[45] R. Hang, Q. Liu, H. Song, and Y. Sun, "Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion,"

IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 2, pp. 783–794, 2015.
[46] M. Liang, L. Jiao, S. Yang, F. Liu, B. Hou, and H. Chen, "Deep multiscale spectral-spatial feature fusion for hyperspectral images classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 8, pp. 2911–2924, 2018.
[47] D. Kingma, L. Ba *et al.*, "Adam: A method for stochastic optimization," 2015.
[48] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

Zhongtian Ma received his B.S. degree from Beihang University, Beijing, China, in 2019. He is currently working toward the M.S. degree in the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University. His research interests include image classification and deep learning applications in remote sensing.

Zhiguo Jiang received his B.S., M.S. and Ph.D. degrees from Beihang University, Beijing, China, in 1987, 1990 and 2005, respectively, where he is a professor at School of Astronautics. He currently serves as a standing member of the Executive Council of China Society of Image and Graphics and also serves as a member of the Executive Council of Chinese Society of Astronautics. He is an Editor for the Chinese Journal of Stereology and Image Analysis. His current research interests include remote sensing image analysis, target detection, tracking and recognition, and medical image processing. He is a member of IEEE.

Haopeng Zhang received his B.S. and Ph.D. degrees from Beihang University, Beijing, China, in 2008 and 2014, respectively, where he is currently an associate professor in the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics. He is a member of IEEE. His main research interests include remote sensing image processing, multi-view object recognition, 3D object recognition and pose estimation, and other related areas in pattern recognition, computer vision, and machine learning.