

# Multiscale Superpixel HGCN Combining CNN for Semi-Supervised Hyperspectral Image Classification

Jiayue Lu and Sei-ichiro Kamata \*

Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan

Email: lujiayue@ruri.waseda.jp (J.L.); kam@waseda.jp (S.K.)

\*Corresponding author

**Abstract**—In recent years, Graph Convolutional Networks (GCN) have witnessed increasing applications in hyperspectral image classification tasks. In comparison to Convolutional Neural Networks, graph representations providing a more effective means to exploit the complex interplay of spatial and spectral features in hyperspectral images, emphasizing their potential to address the challenges associated with limited labeled data in hyperspectral image classification tasks. Although Graph Convolutional Networks are able to capture Hyperspectral Image (HSI) spatial context structure well, they lack the ability to capture pixel-level spectral spatial features compared to Convolutional Neural Networks (CNNs). In order to fully utilize the advantages of Convolutional Neural Networks and Graph Convolutional Networks, in this paper, we propose a model that combines superpixel-based Hypergraph Convolutional Networks features with patch-based Convolutional Neural Network features, engaging in feature learning on both small-scale regular regions and large-scale irregular regions. To test the model, we select 2% of the total number of dataset labels for training, 2% of the total number of dataset labels for validation and the 96% labels for testing. An overall accuracy of 92.37% and 95.86% was obtained in the Indian Pines and Pavia University dataset which is higher than other state-of-the-art methods and achieved a more accurate classification results on the landcover boundary areas.

**Keywords**—computer vision, hyperspectral image classification, graph learning, convolutional networks

## I. INTRODUCTION

In recent years, the application of deep learning techniques has significantly reshaped the landscape of remote sensing image analysis, particularly in the Hyperspectral Image (HSI) classification task, which aims to discriminate the class of each pixel. Compared with the RGB image, hyperspectral image contains hundreds of contiguous spectral bands. Rich spectral information enables the discrimination of subtle material differences through, making HSIs play an important role in Earth Observation tasks, such as precision

agriculture [1], mineral exploration [2], urban planning, and others [3]. Achieving high accuracy in HSI classification not only enhances the quality of the practical applications but also represents an extensively investigated subject in the field of remote sensing research.

However, HSI data has the characteristics of a large number of spectral channels and spatial variability of spectral features, these characteristics of HSI data bring difficulties and challenges to the classification task. Specially, in practical scenarios, due to the expert annotation of HSI is labor-intensive and time-consuming, the availability of labeled samples is significantly restricted. Early on, the strategy of combining Feature Extraction (FE) with classifiers was an important research topic in HSI classification. In comparison to traditional approaches, deep learning methods have the capability to automatically learn adaptive and robust features from training data. Among these, Convolutional Neural Network (CNN) has is the most widely used method for extracting spectral-spatial features from HSI. Various CNN architectures have been proposed, ranging from 1D-CNN [4] to 3D-CNN [5] to enhance the learning capacity of spectral-spatial features.

While CNN-based methods have achieved commendable accuracy in classification, they face limitations due to the size of the fixed convolutional kernels. These methods overlook the dependency between pixels at longer distances and struggle to aggregate similar pixels distributed across different regions in HSI. Consequently, their performance is not so good when dealing with small-scale training samples. Due to the extremely limited availability of HSI labeling data, recently, Semi-Supervised Learning (SSL) has gained significant attention. SSL can leverage both labeled and unlabeled data, combining the advantages of supervised and unsupervised algorithms to yield more accurate classification results. Due to its ability to perform convolution on graphs, Graph Convolutional Networks (GCN) have been widely used in the HSI semi-supervised classification task. Qin *et al.* [6] proposed a semi-supervised GCN method, where Hyperspectral Images (HSI) are initially encoded into a graph.

However, dealing with a large number of pixels in HSI poses computational challenges when considering each pixel as a node in the graph. To address this issue, Wan *et al.* [7] suggested using superpixels instead of pixels as nodes, significantly reducing the computational complexity and making GCN more practical. However, existing methods still have some problems in the HSI semi-supervised classification task:

- The HSI classification task encounters difficulties attributed to the intricate multidimensional properties of HSI, encompassing both spectral and spatial domains. Relationships between pixels involve not only spectral features but also various aspects such as spatial structure and spatial-spectral associations.
- Regarding the multi-range relations in spatial positions, it refers to the fact that in hyperspectral images, the spatial relationships between pixels cover various scales or ranges due to the potential existence of different spatial structures in the landscape.

Therefore, in this work, we propose a semi-supervised HSI classification model called MSHFC which has the following contributions:

- We propose to construct hypergraphs on the superpixel region to enhance the capability of capturing HSI contextual information. Superpixels can minimize the influence of noise at the individual pixel level. Additionally, hypergraphs encode relationships between superpixels as a graph structure offering a richer representation of spectral-spatial associations.
- We designed a multiscale superpixel-level Hypergraph Convolutional Network (HGCN) model to offer distinct feature representations due to HSI contains various land cover details at different scales. Utilizing a multiscale superpixel hypergraph allows the model to consider both detailed features and global patterns, offering a more comprehensive depiction of spectral and spatial characteristics.

We propose to utilize a CNN module with spatial and spectral attention which can supplement the detailed features of ground objects that are lost by HGCN. By fusion the features obtained from HGCN and CNN, the adaptability of the classification model to complex scenes

can be enhanced, thereby achieving better classification performance.

The rest of the study is divided into the following sections: Section II introduce the basic concepts of the proposed method and summarizes some previous research, Section III shows the proposed methodology, Section IV provides the experiment results and discussion. Section V concludes with a summary and our future works.

## II. MATERIALS AND METHODS

In Section II, we review the concepts and principles CNN, HGCN and superpixel segmentation. They are the basic algorithms of our proposed method.

### A. Convolutional Neural Networks (CNNs)

In the region of HSI classification, Convolutional Neural Networks (CNNs) have become a fundamental tool for feature extraction. Adapted and extended from their original 2D image design, CNNs prove effective in processing HSI by capturing both spatial and spectral information. The CNN method generally first styles the original HSI images into square patches, and then extracts features through convolution operations within each patch as shwon in Fig. 1.

CNNs are a good fit for the complex and high-dimensional nature of HSI data because of their remarkable ability to autonomously learn hierarchical representations from data. Countless studies have applied different forms of CNNs to HSI classification tasks. For example, Ying *et al.* [8] proposed a 3-D convolutional network for learning HSI spectral-spatial features. Hang *et al.* [9] designed a two branch 2D-CNN based model which incorporated the spectral and spatial attention mechanism for capturing more discriminative channels or positions. However, traditional CNNs mainly focus on the spatial local relationship between pixels, while ignoring the rich spectral information in hyperspectral images. In addition, CNNs have limited processing capabilities for imbalanced data, which can easily lead to poor classification performance for categories with a small number of samples. Therefore, we also try to combine CNN with other mechanisms in this article to obtain better classification results.

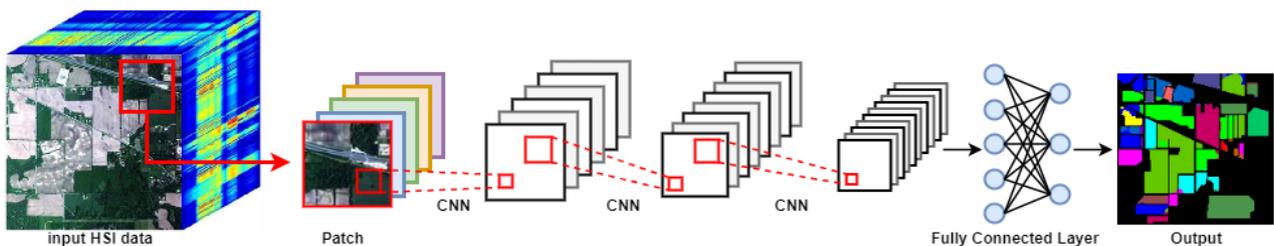


Fig. 1. The convolutional network architecture (The red box indicates patch).

B. Hypergraph Convolutional Networks (HGCN)

In the HSI classification task, a noteworthy advancement is the Hypergraph Convolutional Network (HGCN) [10] algorithm. It is an extension of the widely used Graph Convolutional Network (GCN). Fig. 2 shows the general structure of graph and hypergraph. An edge in a typical graph structure may only establish a one-to-one pair of first-order interactions. However, an edge of a hypergraph can contain several nodes, which expresses higher-order relationships in the data. This is the main distinction between a graph and a hypergraph. Due to the advantages of hypergraph's structure, it is often used in applications that deal with complex data correlation of multimodal or multiple types of data, such as social networks and recommendation systems. While traditional GCN excels in capturing relationships in graph-structured data, HGCN takes a step further structures, offering a more expressive way to model complex dependencies among the HSI data.

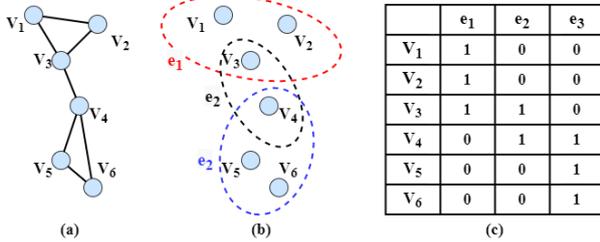


Fig. 2. General structure of graph and hypergraph. (a) A simple graph; (b) A hypergraph; (c) Incident matrix for the hypergraph.

C. Superpixel Segmentation

Ren and Malik [11] introduced an image segmentation technique called superpixel segmentation (SP). Superpixels are groups of adjacent pixels with similar characteristics that are gradually combined into visually

meaningful irregular pixel clusters. These characteristics include texture, color and other visual characteristics. Establishing local regions in hyperspectral image (HSI) classification is crucial to capture spatial features in spectral spatial models and many researches have explored and implement superpixel segmentation in HSI classification tasks [12].

A commonly used superpixel segmentation method for HSI classification is Entropy Rate superpixel Segmentation (ERS) [13]. The ERS algorithm can be simply summarized into the following two steps:

Firstly, determine the desired total number of superpixels, define as parameter  $S$ . Then, construct the graph, where represents the set of vertices corresponding to pixels, and is the set of edges computed using a Gaussian decreasing function based on pixel distances. Secondly, the primary objective of Entropy Rate Superpixel Segmentation (ERS) is to identify an optimal subset that can partition the graph into connected subgraphs. Formally, the segmentation task can be formulated as an optimization problem as below:

$$\arg Q \max H(Q) + \gamma B(Q) \quad (1)$$

where  $H(Q)$  is the constraint of entropy rate to make clusters.  $B(Q)$  is the balance term to facilitate the clusters to have a similar spatial size.  $\gamma \geq 0$  is a coefficient to adjust the influence of the balance term.

III. METHODS

The overall architecture of the proposed method is shown in Fig. 3. The main processes of this work can be described as the following parts: multiscale superpixel segmentation, hypergraph construction and HGCN, CNN feature extraction with spectral-spatial attention and the fusion operation. The details of each module will be introduced in Section III.

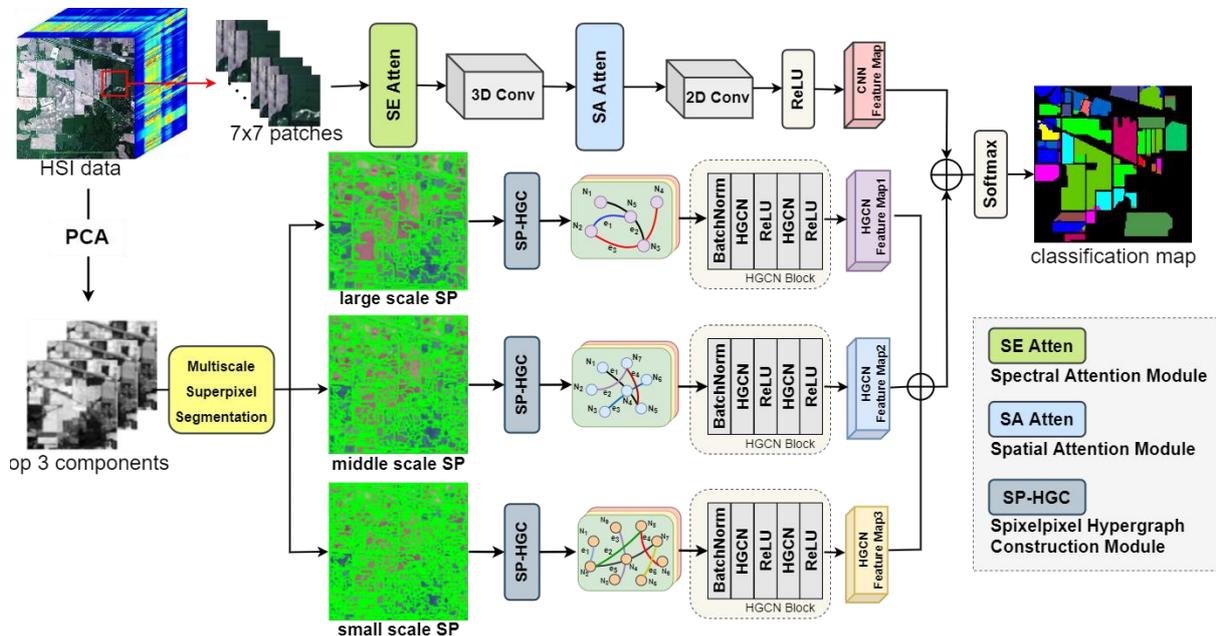


Fig. 3. The overall architecture of the proposed method.

### A. Multiscale Superpixel Segmentation

Objects in Hyperspectral Images (HSI) typically exhibit complex spatial structures, varying in size, shape, and distribution. Numerous studies have highlighted the efficacy of multiscale superpixel segmentation as a valuable strategy to utilize spatial information in HSI.

Multiscale superpixel segmentation allows modeling of images' objects at different scales, facilitating a more comprehensive capture of their spatial characteristics.

In this study, we drew inspiration from the multiscale superpixel segmentation method proposed by Xi *et al.* [14]. It is a heuristic approach to determine the superpixel number, taking into account the inherent characteristics of hyperspectral images (HSIs), such as spatial size and resolution. The calculation for the number of superpixels at scale  $m$  is expressed as:

$$S_m = \text{floor}\left(\frac{H \times W}{\text{floor}(100 \times 0.7^{\sqrt{\text{res}}}) \times 2^{m-1}}\right), m = \{1, 2, \dots, M\} \quad (2)$$

where floor is the value down to the nearest integer.  $H \times W$  is the spatial size and res is the spatial resolution. Meanwhile,  $M$  is the number of considered scales.

Taking the Indian Pines (IP) dataset, the most widely used data set in HSI classification tasks, as an example, we set three different scales: small, middle and large ( $m$  in Formula 3.3 is set to 0, 1, 2), and the resulting scale set is: {1051, 525, 262}.

### B. Superpixel Hypergraph Construction (SP-HGC Module) and HGCN

#### 1) SP-HGC module

Fig. 4 is the schematic illustration of hypergraph construction based on the superpixel regions. After ERS [13] superpixel segmentation, the HSI is segmented into several superpixels, which are denoted as  $S = \{s_1, s_2, \dots, s_N\}$ . Each superpixels region can be taken as a hypergraph node, and the hypergraph can be constructed by the similarity or the relationships between the nodes.

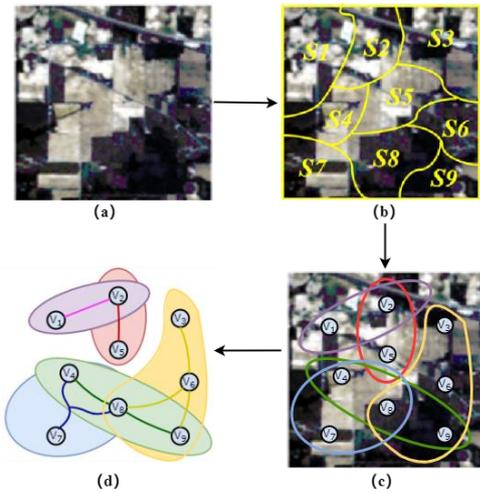


Fig. 4. Hypergraph construction process. (a) input HSI; (b) superpixel segmentation results; (c) hyperedge generation: blue dots are the hypergraph nodes; (d) constructed hypergraph: each color area represents a hyperedge.

For predicting the label of each pixel, it is necessary to assign the features of the superpixels to individual pixels. For this purpose, we inspired by Xu *et al.* work [15], they establish a mapping matrix  $\mathbf{M} \in \mathbb{R}^{wh \times N}$  to decode superpixels into pixels. The 2D HSI flatten data  $\mathbf{P}' \in \mathbb{R}^{wh \times c}$  are represent by the 3D HSI along the spatial dimension as the following calculation:

$$\mathbf{P}'(h(i-1) + j, :) = \mathbf{P}(i, j, :) \quad (3)$$

where  $\mathbf{P}(i, j, :)$  is the pixel vector of the 3D HSI at the spatial position  $(i, j)$ ,  $(i = 1, 2, \dots, w; j = 1, 2, \dots, h)$ . The mapping matrix  $\mathbf{M}$  is defined as:

$$\mathbf{M}(i, j) = \begin{cases} 1, & \text{if } \mathbf{P}'(i, :) \in s_j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where  $\mathbf{P}'(i, :)$  is the  $i$ th pixel of the HSI flatten data.

A hypergraph can be defined as  $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{A})$ , where  $\mathbf{V}$ ,  $\mathbf{E}$ , and  $\mathbf{A}$  are the set of nodes, edges and an attribution of nodes. By using the encoder in Eq. (5), HSI can be easily encoded into graph nodes. Each node of  $G$  corresponds to a superpixel of HSI.

$$\mathbf{V} = \text{Encode}(\mathbf{P}; \mathbf{M}) = \mathbf{M}'^T \mathbf{P}' \quad (5)$$

where  $\mathbf{M}'$  is the normalized  $\mathbf{M}$  by column. Suppose the node set  $V$  contains  $m$  nodes, each node  $\mathbf{v}_i$  is assigned with a weight vector  $\mathbf{x}_i$  which is computed as follows:

$$\mathbf{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{d}_j \quad (6)$$

where  $n_i$  is the number of pixels in the  $i$ th superpixel, and  $\mathbf{d}_j \in \mathbb{R}^{1 \times c}$  ( $i = 1, 2, \dots, n_i$ ) is the spectral vector of the  $j$ th pixel. We can obtain weight set of nodes  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}^T \in \mathbb{R}^{N \times c}$ .

For node  $\mathbf{v}_i$ ,  $n$  nodes with the most similar attribute to  $\mathbf{v}_i$  are selected form the neighborhood of center node  $\mathbf{v}_i$  as  $\mathcal{N}(\mathbf{v}_i)$ . The hyperedge  $\mathbf{e}_i$  contains the center node  $\mathbf{v}_i$  and the nodes  $\mathbf{v}_j$  in the neighborhood  $\mathcal{N}(\mathbf{v}_i)$ . The definition of hyperedge  $\mathbf{e}_i$  is as follows:

$$\mathbf{e}_i = \mathcal{N}(\mathbf{v}_i) \cup \{\mathbf{v}_i\} \quad (7)$$

The adjacency matrix of hypergraph  $\mathbf{H} \in \mathbb{R}^{N \times N}$  is defined as:

$$\mathbf{H}(i, j) = \begin{cases} 1, & \text{if } \mathbf{v}_i \in \mathbf{e}_j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Each hyperedge's importance is considered equal, and the weight of every hyperedge is set to 1. Consequently, the attribution matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is set as an identity matrix. The elements of the node degree matrix  $\mathbf{D}_v \in \mathbb{R}^{N \times N}$  are:

$$\mathbf{D}_v(i, i) = \sum_{j=1}^N \mathbf{A}(i, i) \mathbf{H}(i, j) \quad (9)$$

The degree matrix  $\mathbf{D}_e \in \mathbb{R}^{N \times N}$  of hyperedges is:

$$\mathbf{D}_e(i, i) = \sum_{j=1}^N \mathbf{H}(i, j) \quad (10)$$

#### 2) Hypergraph Convolutional Network (HGCN)

After constructing the hypergraph, we operate HGCN to update the features of the nodes. Given a hypergraph

representation for a HSI, the normalized hypergraph Laplacian  $L$  is defined as:

$$L_H = I - D_V^{-1/2} H A D_e^{-1} H^T D_V^{-1/2} \quad (11)$$

where  $I$  is the identity matrix.

In a hypergraph, the convolution on a signal  $\mathbf{x}$  can be defined as the convolution operation between the signal  $\mathbf{x}$  and a filter  $\mathbf{g}$ , similar to graph convolution in the spectral domain.

$$\mathbf{g} \times \mathbf{x} = \Phi \mathbf{g}(\Lambda) \Phi^T \mathbf{x} \quad (12)$$

where  $\Phi = (\varphi_1, \varphi_2, \dots, \varphi_N) \in \mathbb{R}^{N \times N}$  is composed of the orthogonal eigenvectors of by column, and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$  are the corresponding non-negative eigenvalues. After the row-normalization [10], Eq. (12) can be expressed as follows:

$$\mathbf{g} \times \mathbf{x} \approx \sigma(D_V^{-1} H A D_e^{-1} H^T \mathbf{x}) \quad (13)$$

To facilitate efficient information propagation on the constructed hypergraph for HSI classification, we employ classification through the use of Eq. (14). The layer-wise propagation within the hypergraph is outlined as follows:

$$\mathbf{X}^{(l+1)} = \sigma(D_V^{-1} H A D_e^{-1} H^T \mathbf{X}^{(l)} \Theta^{(l)}) \quad (14)$$

where  $\Theta^{(l)}$  is the trainable weight at the  $l$ -th layer,  $\sigma$  is the activation function, are the features of hypergraph st the  $l + 1$ th and  $l$ -th layer.

After the HGCN layers, the features of pixels  $P'$  are computed as decoding the features of superpixels by using the matrix  $M$ . The features of nodes need be assigned to pixels because the task is to classify pixels instead of superpixels. To propagate the node features to pixels, with the association matrix  $M$ , which can be written as:

$$\mathbf{P}^* = \text{Decode}(\mathbf{V}; \mathbf{M}) = \text{Reshape}(\mathbf{M}\mathbf{V}) \quad (15)$$

where  $\text{Reshape}(\cdot)$  means restoring the spatial dimension of the flattened data. After the decoding procedure, the hypergraph features can be projected back to the image space.

HGCN can efficiently propagate information in graph structures, capturing relationships between nodes effectively, so they can exhibit outstanding performance even in shallower networks. In this work, we use two HGCN layer to capture the HSI features.

### C. CNN Feature Extraction with Spectral-Spatial Attention

Considering that in the HGCN branch, I perform dimensionality reduction preprocessing on the original HSI images, reducing the original hundreds of bands to single digits. This may lose the ample spectral information in the HSI data during the model training process. So, I designed CNN branches with spectral attention and spatial attention as the supplement.

Initially, the input HSI data is divided into  $7 \times 7$  patches. Hu *et al.* proposed a method called SENet [16], which improves accuracy by modeling the correlation between feature channels and intensifying important

features. Inspired by their work, we design a spectral attention module.

By recording spectrum dependencies among various bands, the spectral attention module seeks to highlight spectral features in HSI data, as shown in Fig. 5. It can obtain a weight for each band, which it can then use to adjust the 3D convolutional layer's input. This allows us to express the spectral attention as follows:

$$h_{avg}^k = \frac{1}{H \times H} \sum_{i=1}^H \sum_{j=1}^H f(k, i, j) \quad (16)$$

where  $f \in \mathbb{R}^{H \times H \times C}$  is the feature obtained from the processing of the neighborhood pixel block through the convolution layers,  $f(k, i, j)$  represents the value of the position  $(i, j)$  of the  $k$ th channel of the feature map  $f$ ,  $h_{avg}$  is the result of global average pooling,  $h_{avg}^k$  is the value of the  $k$ th channel of  $h_{avg}$ .

$$A_{se} = \sigma_2(FC_2(\sigma_1(FC_1(h_{avg})))) \quad (17)$$

where  $\sigma_1$  and  $\sigma_2$  represent LeakyReLU and sigmoid activation functions, respectively.  $FC_1$  and  $FC_2$  are two fully connected layers.

By calculating the average and maximum values of various channels at various spatial locations, spatial attention is able to determine the correlation between various geographical locations.

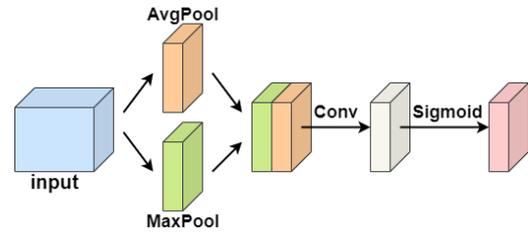


Fig. 5. Spatial attention module (SA Atten).

Concatenation is done once pooling procedures are complete. A convolutional layer is used in further processing to draw attention to important spatial regions. Fig. 6 illustrate the spatial attention module' process. The spatial attention can be formulated as:

$$A_{sa} = \sigma([\text{AvgPool}(f); \text{MaxPool}(f) \times W]) \quad (18)$$

where  $f$  is the feature map obtained from the 3D-CNN layer,  $\sigma$  is the sigmoid function and  $W$  is the weight of the convolution.

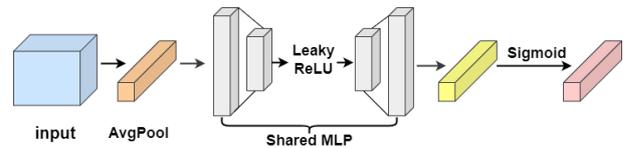


Fig. 6. Spatial attention MLP module (SA Atten).

Considering the multi-scale hypergraph learning already has a certain computational complexity, the CNN branch needs to be lightweight. Fig. 7 displays the architecture of the 2D and 3D CNNs that were utilized in this study.

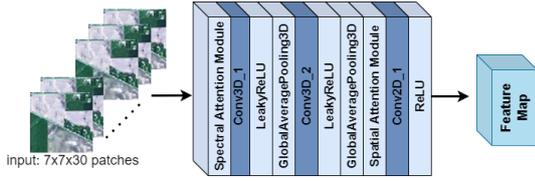


Fig. 7. Structure of 3D-CNN and 2D-CNN used in this work.

#### D. Feature Fusion

After obtaining features of different superpixel scales, we fusion the three feature maps by the manually set weights. The sum of the three scale superpixel HGCN weights is 1, which constitutes the final feature of the HGCN branch. We set a larger weight as 0.4 for the features extracted by middle-scale superpixel HGCN. Considering that the features obtained by small-scale and large-scale HGCN complement the overall features, the weights of the two are evenly set to 0.2 and 0.2. Therefore, the features obtained from the whole HGCN branch can be calculated as:

$$F_{HG} = 0.2 \times F_{H1} + 0.4 \times F_{H2} + 0.2 \times F_{H3} \quad (19)$$

where  $F_{H1}$ ,  $F_{H2}$  and  $F_{H3}$  are the feature maps of small scale HGCN, middle scale HGCN and large scale HGCN, respectively. The fusion illustration is shown in Fig. 8.

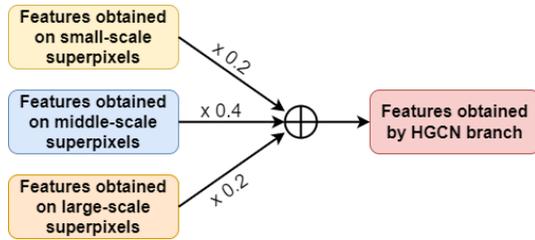


Fig. 8. Fusion of features obtained by HGCN on superpixels of different scales.

The CNN and HGCN branches have different feature distributions because they use different neural network models. The model's integration can be improved by assigning distinct weights  $\eta$  to the two types of feature extraction algorithms. The whole feature extraction process can be expressed by:

$$F = \eta \times F_C \oplus (1 - \eta) \times F_{HG} \quad (20)$$

where  $F_C$  and  $F_{HG}$  are the feature maps CNN branch and HGCN branch, respectively.

## IV. RESULT AND DISCUSSION

### A. Datasets

The Indian Pines (IP) dataset contains 224 spectral channels with the spatial size is  $145 \times 145$  pixels with a spatial resolution as 20 m. To enhance the relevance of the dataset, water absorption bands (nos. 104–108, 150–163, and 220) were removed, resulting in the utilization of 200 spectral bands. There are 10,269 labeled pixels distributed across 16 categories. The ground truth map is shown in Fig. 9.

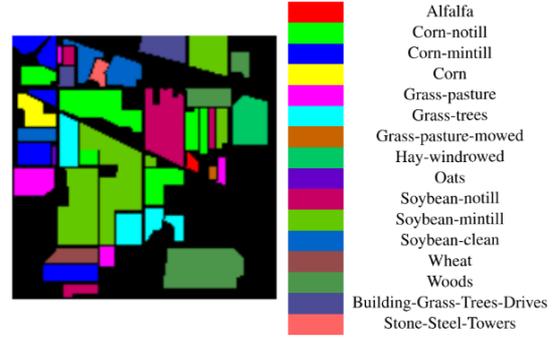


Fig. 9. Indian Pines dataset.

The Pavia University (PU) dataset comprises an image with 115 bands spanning from 430 to 860 nm and a spatial resolution of 1.3 m. In the initial dataset, 12 bands were excluded due to significant noise, leaving 103 bands for experimental purposes. The image dimensions are  $610 \times 340$ . There are 42,776 labeled pixels distributed across 9 categories. The ground truth map are shown in Fig. 10.

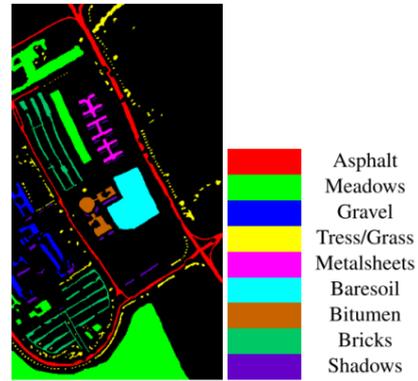


Fig. 10. Pavia University dataset.

Table I shows the number of samples of each category in IP and PU datasets.

TABLE I. THE NUMBER OF SAMPLES OF EACH CATEGORY IN THE IP DATASET AND PU DATASET

Indian Pines dataset		Pavia University dataset	
Class No.	Samples	Class No.	Samples
1	46	1	6631
2	1428	2	18649
3	830	3	2099
4	237	4	3064
5	483	5	1345
6	730	6	5029
7	28	7	1330
8	478	8	3682
9	20	9	947
10	972	-	-
11	2455	-	-
12	593	-	-
13	205	-	-
14	1265	-	-
15	386	-	-
16	93	-	-
Total	10249	Total	42776

**B. Implement Details**

For each dataset, we select 2% of the total number of labels in each category for training, 2% for validation, 96% for testing. The samples are randomly selected. Therefore, for IP and PU dataset, the number of samples in the training, validation and testing sets are: {205, 205, 9859} and {856, 856, 41064}, respectively. Each dataset dimension of the spectral bands is reduced to 3 by Principal Component Analysis (PCA).

The ERS superpixel segmentation method is implemented to get the multiscale superpixels. The superpixel numbers in three scales are, {1051, 525, 262} for IP dataset, {3190, 1583, 791} for PU dataset. The node neighborhood in the hypergraph modeling is set to 8. The training epoch is set to 200. Details of the experimental environment setting are shown in Table II.

TABLE II. DETAILS OF THE EXPERIMENTAL ENVIRONMENT SETTING

Configuration Category	Item	Configuration
Hardware	GPU	RTX 3090 × 1
Framework	PyTorch	1.10.2
Hyperparameters	Learning rate	0.0001
	Weight decay	0.01
	Optimizer	Adam
	Loss function	Cross-entropy loss

**C. Evaluation Matrix**

This study uses the per-Class Average (CA), Overall Accuracy (OA), Average Accuracy (AA), and the kappa coefficient for a quantitative assessment of the proposed method.

The percentage of accurately categorized pixels relative to the total number of test samples is represented by OA. Because it does not account for specific class circumstances, this statistic assesses the overall categorization performance of HSI and is hence less affected by unequal class distributions. Meanwhile, AA denotes the average accuracy of all land cover classifications, providing insights into the precision of different categories.

TABLE III. CA, OA, AA, AND KAPPA COEFFICIENT ON THE INDIAN PINES DATASET

Class No.	Class Name	3D-CNN [17]	GCN [6]	miniGCN [18]	CEGCN [19]	F <sup>2</sup> HNN [20]	Ours
1	Alfalfa	49.25	0.68	35.41	43.84	79.35	82.44
2	Corn-notill	84.02	69.42	85.97	82.70	85.79	89.88
3	Corn-mintill	93.26	75.27	78.14	91.52	92.38	91.67
4	Corn	86.60	67.74	55.22	88.40	86.56	85.90
5	Grass-pasture	87.94	71.42	72.69	82.97	89.01	89.44
6	Grass-trees	96.80	83.75	93.50	91.27	97.14	97.71
7	Grass-pasture-mowed	78.88	5.50	6.15	58.90	98.85	97.38
8	Hay-windrowed	99.76	92.95	98.62	97.17	99.87	97.87
9	Oats	78.68	4.38	10.53	68.60	88.22	93.95
10	Soybean-notill	90.78	64.36	88.78	86.77	90.70	88.14
11	Soybean-mintill	95.84	90.16	94.92	94.47	96.86	96.35
12	Soybean-clean	76.25	58.30	72.21	76.60	79.58	83.04
13	Wheat	98.73	76.67	94.76	94.16	95.41	85.30
14	Woods	97.38	94.81	93.51	96.78	97.29	99.35
15	Building-Grass-Tress-Drivers	89.46	62.01	90.03	92.84	90.13	93.64
16	Stone-Steel-Towers	82.33	2.81	74.68	86.30	84.36	86.27
	Overall Accuracy	91.64	78.19	87.31	89.83	92.49	93.21
	Average Accuracy	86.62	57.51	71.57	83.33	90.72	91.77
	kappa	90.47	75.04	85.46	88.42	91.45	92.26

A statistical parameter called the kappa coefficient is used to evaluate the degree of agreement between the classification results and the ground truth. This allows for an overall evaluation of the classifier’s efficacy. A greater Kappa value, which ranges from -1 to 1, denotes improved classification performance with the selected approach.

**D. Results**

*1) Fusion ratio evaluation*

In Section III-D, we propose to set a parameter  $\eta$  to fuse the features of the HGNC branch and the CNN branch. Here, I conducted the experiments by setting the  $\eta$  to 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 to find the optimized fusion ratio.

To test different ratios, we conducted experiments on the IP and PU datasets respectively. The classification results are shown in Fig. 11.

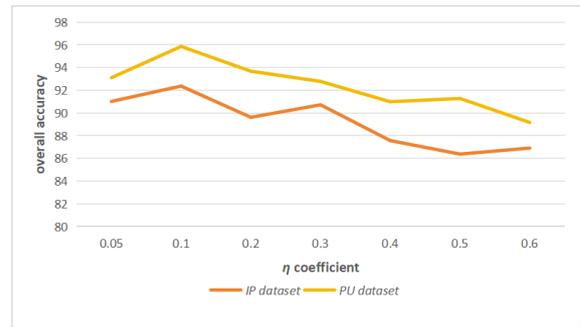


Fig. 11. Classification accuracy on different fusion ratio.

From the results we can see, for IP dataset and PU dataset, when  $\eta = 0.1$  can get the best results. So, in subsequent experiments, we will keep this setting for further research.

*2) Quantitative results*

Using the two datasets, the proposed method is compared with five other HSI classification methods. Tables III and IV shows the classification results on the IP dataset and PU dataset, respectively.

TABLE IV. CA, OA, AA, AND KAPPA COEFFICIENT ON THE PAVIA UNIVERSITY DATASET

Class No.	Class Name	3D-CNN [17]	GCN [6]	miniGCN [18]	CEGCN [19]	F <sup>2</sup> HNN [20]	Ours
1	Asphalt	91.63	78.14	85.05	93.17	91.51	92.06
2	Meadows	98.39	90.65	97.84	97.71	98.81	99.57
3	Gravel	79.37	50.19	74.87	83.05	87.35	88.99
4	Trees	68.41	15.62	69.70	77.76	79.44	87.48
5	Metal sheets	99.11	99.04	99.63	99.37	98.49	99.74
6	Bare soil	99.36	63.44	95.81	97.88	98.43	99.42
7	Bitumen	84.59	15.07	83.59	91.35	86.78	95.14
8	Bricks	87.84	64.71	84.02	91.27	93.03	95.49
9	Shadows	31.89	4.01	28.79	47.98	54.85	69.64
	Overall Accuracy	91.59	71.92	89.23	93.08	93.82	95.86
	Average Accuracy	82.29	53.43	79.70	86.61	87.63	91.95
	kappa	88.79	61.57	85.66	90.82	91.80	94.50

According to the IP dataset classification results as shown in Table III, the proposed method’s CA in some categories is somewhat lower than previous SOTA approaches. However, CA for most categories are better than the SOTA method, and achieve the final OA, AA, and kappa value on 93.21%, 91.77% and 0.9226, which are all outperformed than other methods. Table IV shows the classification results on the PU dataset. The proposed method exceeds the classification accuracy of the SOTA model on almost every category and achieves the OA of 95.86%, AA of 91.95% and kappa coefficient of 0.945.

Especially for the landcovers in the dataset that have fewer samples than other categories, such as the ninth category in the IP dataset, oats. Our method improves the

classification accuracy from 88.22% in the SOTA model to 93.95%. Similarly, for the ninth category in the PU dataset, shadows. Our method improves the classification accuracy from 54.85% in the SOTA model to 69.64%.

### 3) Qualitative results

Figs. 12 and 13 shows the classification maps of the IP dataset and the PU dataset. From Figs. 12 and 13, we can find, the proposed method can achieve more accurate classification results at the landcover boundary than previous SOTA models, as demonstrated by the classification maps. It can also achieve good classification results on categories that seriously lack labels in the datasets.

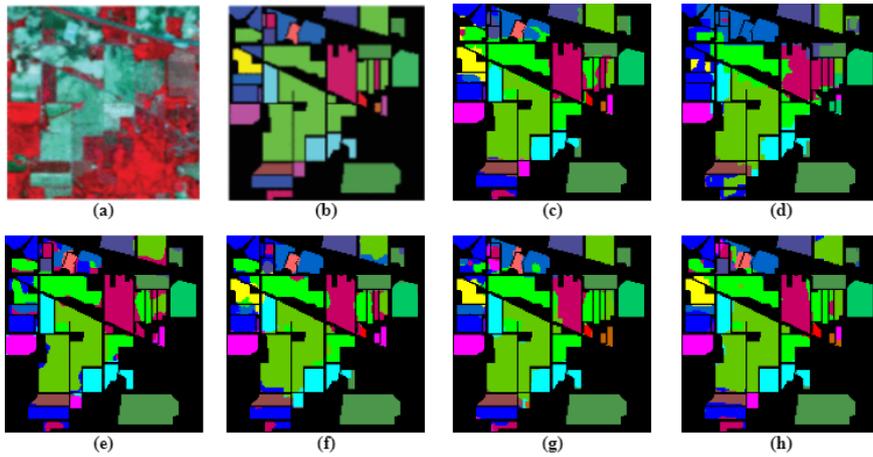


Fig. 12. Classification maps of different methods for the Indian Pine dataset. (a) False-color image. (b) Ground truth. (c) 3D-CNN [17]. (d) GCN [6]. (e) miniGCN [18]. (f) CEGCN [19]. (g) F2HNN [20]. (h) Proposed.

## V. CONCLUSION

This study proposes a new semi-supervised HSI classification technique. The proposed framework mainly includes two branches: the CNN feature extraction branch with spectral and spatial attention operation, and the HGCN feature extraction branch which conduct HGCN on superpixel regions at different scales.

Hypergraph learning can effectively utilize extremely limited training samples. Multi-scale superpixel segmentation can help the model capture detailed features and global features simultaneously. The CNN branch embedded with the attention mechanism captures the spectral-spatial information lost due to HGCN dimensionality reduction in a higher dimension. The

efficiency of the proposed method has been demonstrated by extensive experimental findings on the two HSI datasets. We achieve an overall accuracy of 93.21% and 95.86% on the Indian Pine and Pavia University dataset, respectively.

The visual findings make it clear that, when compared to other SOTA models, the suggested model performs better in terms of categorizing pixels at land cover type boundaries or intersections.

In future work, we look forward to applying the proposed model to practical tasks, such as agricultural detection or disaster situations. This requires further improving the generalization ability of the model and reducing the computational complexity.

Although the proposed method have achieved fine classification result, how to better utilize the rich spectral-spatial information in HSI to construct hypergraphs is an important work afterwards. For multiscale superpixel segmentation part, we are also considering whether the

segmentation scale can be learned automatically by the features of HSI data to make the model be robust on other HSI data to improve the model's generalization capability.

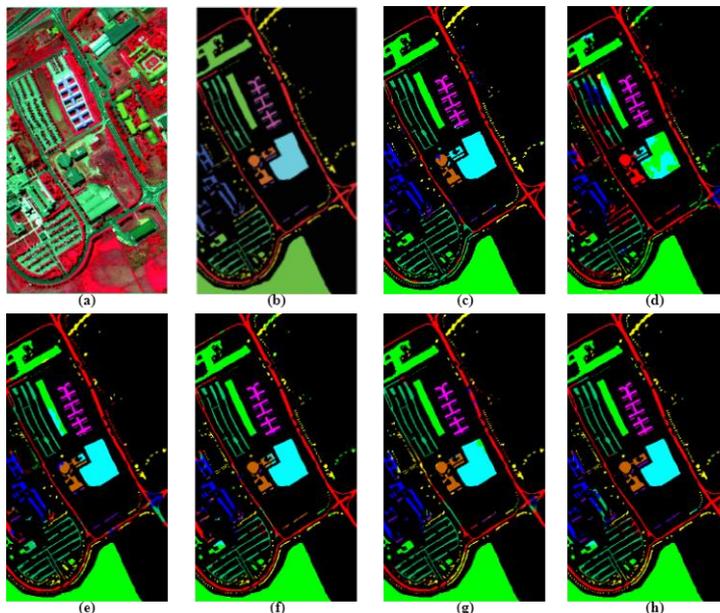


Fig. 13. Classification maps of different methods for the Pavia University dataset. (a) False-color image. (b) Ground truth. (c) 3D-CNN [17]. (d) GCN [6]. (e) miniGCN [18]. (f) CEGCN [19]. (g) F2HNN [20]. (h) Proposed.

#### CONFLICT OF INTEREST

The authors declare no conflict of interest.

#### AUTHOR CONTRIBUTIONS

Jiayue Lu: evaluated previous methods, gave the new idea and implemented experiments, then composed the paper. Sei-ichiro Kamata: initiated the challenge, supervised, reviewed the paper in the whole writing process. Both authors had approved the final version.

#### ACKNOWLEDGMENT

The authors wish to express appreciation towards every members in the Waseda University, Graduate School of Information, Production and Systems, Image Media Laboratory for giving the advice and supports.

#### REFERENCES

- [1] A. Lowe, N. Harrison, and A. P. French, "Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress," *Plant Methods*, vol. 13, no. 1, 80, 2017.
- [2] S. Peyghambari and Y. Zhang, "Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: An updated review," *Journal of Applied Remote Sensing*, vol. 15, no. 3, 31501, 2021.
- [3] M. J. Khan, H. S. Khan, A. Yousof, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.
- [4] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, pp. 1–12, 2015.
- [5] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sensing*, vol. 9, no. 1, 67, 2017.
- [6] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral-spatial graph convolutional networks for semi supervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, 2019.
- [7] 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, 2019.
- [8] L. Ying, H. Zhang and S. Qiang, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sensing*, vol. 9, no. 1, 67, Jan. 2017.
- [9] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya, "Hyperspectral image classification with attention-aided CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 2281–2293, March 2021.
- [10] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, vol. 110, 107637, 2021.
- [11] Ren and Malik, "Learning a classification model for segmentation," in *Proc. Ninth IEEE International Conference on Computer Vision*, IEEE, 2003, pp. 10–17.
- [12] S. Subudhi, R. N. Patro, P. K. Biswal, and F. Dell'Acqua, "A survey on superpixel segmentation as a preprocessing step in hyperspectral image analysis," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 5015–5035, 2021.
- [13] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. CVPR 2011*, IEEE, 2011, pp. 2097–2104.
- [14] B. Xi, J. Li, Y. Li, R. Song, W. Sun, and Q. Du, "Multiscale context-aware ensemble deep KELM for efficient hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5114–5130, 2020.
- [15] Q. Xu, J. Lin, B. Jiang, J. Liu, and B. Luo, "Hypergraph convolutional network for hyperspectral image classification,"

- Neural Computing and Applications*, vol. 35, no. 29, pp. 21863–21882, 2023.
- [16] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.
- [17] 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.
- [18] 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*, vol. 59, no. 7, pp. 5966–5978, 2020.
- [19] 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,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 10, pp. 8657–8671, 2020.
- [20] Z. Ma, Z. Jiang, and H. Zhang, “Hyperspectral image classification using feature fusion hypergraph convolution neural network,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License ([CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.