



## Article

# Spatial-Pooling-Based Graph Attention U-Net for Hyperspectral Image Classification

Qi Diao <sup>1</sup>, Yaping Dai <sup>1</sup> , Jiacheng Wang <sup>1</sup>, Xiaoxue Feng <sup>1</sup>, Feng Pan <sup>1,\*</sup> and Ce Zhang <sup>2</sup>

<sup>1</sup> School of Automation, Beijing Institute of Technology, Beijing 100081, China; 3120185444@bit.edu.cn (Q.D.); daiyaping@bit.edu.cn (Y.D.); 3220225163@bit.edu.cn (J.W.); fengxiaoxue@bit.edu.cn (X.F.)

<sup>2</sup> School of Geographical Sciences, University of Bristol, Bristol BS8 1SS, UK; ce.zhang@bristol.ac.uk

\* Correspondence: panfeng@bit.edu.cn

**Abstract:** In recent years, graph convolutional networks (GCNs) have attracted increasing attention in hyperspectral image (HSI) classification owing to their exceptional representation capabilities. However, the high computational requirements of GCNs have led most existing GCN-based HSI classification methods to utilize superpixels as graph nodes, thereby limiting the spatial topology scale and neglecting pixel-level spectral-spatial features. To address these limitations, we propose a novel HSI classification network based on graph convolution called the spatial-pooling-based graph attention U-net (SPGAU). Specifically, unlike existing GCN models that rely on fixed graphs, our model involves a spatial pooling method that emulates the region-growing process of superpixels and constructs multi-level graphs by progressively merging adjacent graph nodes. Inspired by the CNN classification framework U-net, SPGAU's model has a U-shaped structure, realizing multi-scale feature extraction from coarse to fine and gradually fusing features from different graph levels. Additionally, the proposed graph attention convolution method adaptively aggregates adjacency information, thereby further enhancing feature extraction efficiency. Moreover, a 1D-CNN is established to extract pixel-level features, striking an optimal balance between enhancing the feature quality and reducing the computational burden. Experimental results on three representative benchmark datasets demonstrate that the proposed SPGAU outperforms other mainstream models both qualitatively and quantitatively.



**Citation:** Diao, Q.; Dai, Y.; Wang, J.; Feng, X.; Pan, F.; Zhang, C.

Spatial-Pooling-Based Graph Attention U-Net for Hyperspectral Image Classification. *Remote Sens.*

2024, 16, 937. <https://doi.org/10.3390/rs16060937>

Academic Editor: Farid Melgani

Received: 19 January 2024

Revised: 1 March 2024

Accepted: 5 March 2024

Published: 7 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** graph convolutional network; dynamic graph; attention mechanism; hyperspectral image classification

## 1. Introduction

The rapid advancement in aerospace vehicle and optics technology has substantially improved hyperspectral imaging techniques [1], resulting in higher-quality and more cost-effective acquisition of hyperspectral images (HSIs) [2]. Each pixel within these images contains an abundance of spectral information, encompassing hundreds of distinct bands [3]. This extensive spectral information renders HSIs suitable for numerous practical applications, including environmental inspection [4], agricultural assessment [5], and military investigations [6]. The advantages of HSI extend to the capability of assigning specific land-cover labels to each hyperspectral pixel, hence promoting interest in HSI classification.

To address the challenge of HSI classification, various methods have been introduced [7]. In essence, HSI classification categorizes pixels by extracting image features, thereby emphasizing the need for discriminative feature extraction [8]. In the early stages, feature extraction methods were often manually designed using machine learning techniques, primarily based on spectral and spatial information found in spectral images. For example, methods like support vector machine (SVM) [9], logistic regression [10], linear discriminant analysis (LDA) [11], and principal component analysis (PCA) [12], which rely on spectral information, were proposed. However, these methods typically focus on HSI features while

disregarding spatial information, resulting in less-than-ideal classification performance [13]. To enrich the feature space, it is imperative to harness spatial information in conjunction with spectral data in HSIs [14]. As a result, several spatial–spectral-based techniques, which consider both spatial and spectral information, have been developed. Among these, specific joint spatial–spectral classifiers leverage the characteristics of HSI data cubes. For instance, the utilization of 3-D wavelet analysis (WT) [15,16] and morphological information methods [17,18] is noteworthy. Furthermore, other methods place a higher emphasis on the spatial relationship between pixels, utilizing concepts like manifold learning [19], superpixel learning [20], and graph representation learning [21]. These techniques enable the construction of structural information within HSI data. While these methods have achieved considerable success in HSI classification, traditional machine learning classification methods are constrained by their reliance on human-crafted features and empirical parameters, which hinders in-depth feature extraction [22].

In comparison, deep learning (DL) can autonomously extract depth features through multi-layer neural networks [23,24]. Following their achievements in the realm of RGB images, DL models have been widely adopted in HSI classification due to their strong feature representation capabilities [25,26]. In their nascent stages, Chen et al. spearheaded the use of DL models for HSI classification by devising a stacked autoencoder for high-level feature extraction [27]. Subsequently, a plethora of DL classifiers for HSI classification have surfaced, achieving exceptional results. Notable examples include convolutional neural networks (CNNs) [28], recurrent neural networks (RNNs) [29], deep belief networks (DBNs) [30], and various other models. Among these deep learning approaches, CNNs have emerged as the most widely employed framework for HSI classification. The learning paradigms of CNN-based methods cover a variety of models, including transfer learning [31], meta-learning [32], active learning [33], and so on. Several CNN models with distinct architectures have been introduced to maximize the utilization of spectral and spatial data in HSIs, enhancing the network's expressiveness by focusing on dimensions, network depth, and complexity. For example, Hu et al. pioneered the incorporation of the CNN model into HSI classification, developing a 1-D CNN classifier for spectral information extraction [34]. To leverage the spatial information in HSIs, Li et al. elevated the dimensionality of CNN, presenting a 3D-CNN model for extracting spectral–spatial features [35]. Additionally, the extraction of spectral–spatial features can be realized through multi-channel fusion, as exemplified by two-channel CNN (TCCNN) [36] and multi-channel CNN (MCCNN) [37]. Furthermore, inspired by the visual attention mechanism, attention techniques have been integrated into CNNs in various recent methods [38,39]. While existing CNN-based methods have made significant performance strides, the convolution operations in CNN models operate on regular square regions, neglecting pixel-to-pixel relationships. Although attention-based CNNs can assign different weights to distinct feature dimensions, they cannot directly evaluate pixel correlations due to CNN models' inability to handle non-Euclidean data, such as the adjacency relationships between pixels.

Non-Euclidean spatial data pervade the real world. Graphs which connect vertices and edges offer a means to represent non-Euclidean data, with graph neural networks (GNNs) capable of directly extracting information from these graph structures [40]. Consequently, GNNs enable comprehensive extraction of spectral–spatial features in hyperspectral imaging (HSI) and yield impressive classification results. Presently, GNN approaches for HSI classification encompass three main methods of constructing graphs: pixel-based [41], selection region-based [42], and superpixel-based [43]. Among these, the pixel-based approach designates each pixel as a vertex, generating the graph structure based on all HSI pixels. Given the typically high resolution of HSIs, this method entails significant computational demands [41]. In contrast, the selection region-based approach constructs the graph using randomly chosen pixel patches from any HSI region rather than the entire HSI dataset. Hong et al. [42] effectively harnessed constructive relations and image features through a dual-channel mode employing GCNs and CNNs. On the other hand, superpixels contain adjacent pixels with similar features, making them a useful preprocessing tool for HSIs.

They mitigate memory consumption and maintain the edge information of ground objects. Ren et al. devised superpixel adjacency graph structures at various scales and adapted these structures at each scale using feature similarity and nuclear diffusion. This approach facilitates high-level feature extraction while conserving computational resources [43]. While the former extracts multi-scale features by simultaneously constructing graph structures of different scales, our method evolves into multi-scale graph structures using the spatial pooling module, subsequently fusing features of varying scales throughout the U-shaped structure.

However, as discussed in [44], pixel-level graph nodes result in substantial computation. In response, a spatial-pooling-based graph attention neural network (SPGAU) is proposed for HSI classification. In this network, we use a superpixel-based method to preprocess the original HSI and greatly reduce the number of graph nodes. Moreover, to extract multi-scale spatial features from HSI, we design a spatial pooling module to obtain multi-scale maps. As the number of graph nodes decreases layer-by-layer, the computational burden gradually lightens. The primary contributions of this article are as follows:

1. We propose a novel spatial pooling module that emulates the process of superpixel merging and conducts random merging operations on neighboring nodes exhibiting high similarity in the graph. This operation not only reduces computational costs but also abstracts the multi-scale graph structure.
2. We incorporate an attention mechanism into the graph convolution network to enhance the efficacy of feature extraction.
3. We employ 1-D CNN to extract per-pixel image features, thus mitigating the potential loss of pixel-level information resulting from superpixel segmentation.

The remainder of the paper is structured as follows: In Section 2, we introduce the proposed SPGAU model. Section 3 describes three HSI classification datasets and experimental settings. In Section 4, we present a systematic discussion of the experimental results. Finally, Section 5 presents the conclusions and provides insight into future research directions.

## 2. Related Work

### 2.1. Graph Convolution

In recent years, graph neural networks (GNNs) have achieved state-of-the-art performance in various graph-related tasks by successfully extending convolution operators to graph-structured data. Existing GCN algorithms can be broadly categorized into two groups: spectral-based and spatial-based approaches. Spectral-based methods predominantly employ Fourier transforms and the graph Laplacian operator to define the convolution operation as a filter, with the goal of noise reduction in the graph signal. The ChebNet [45] algorithm, for instance, utilizes Chebyshev expansion for approximating k-polynomial filters, enhancing computational efficiency. GCN [46] simplifies the Chebyshev polynomial into a local spectrum filter, resulting in a first-order approximation of ChebNet. This approach enables weight and parameter sharing, significantly reducing complexity. To address the computational load associated with Laplacian matrix feature decomposition, the graph wavelet neural network (GWNN) [47] employs local wavelet transforms instead of global Fourier transforms for convolution operations. This strategy leverages the inherent sparsity in wavelet transforms, reducing network complexity. However, these models have limitations, such as their inability to handle directed graphs and their confinement to shallow networks.

In contrast to spectral-based methods, spatial-based methods do not rely on graph theory and directly aggregate neighborhood information for convolution operations. Given the inherent disarray in graph structures, GNN [48] utilizes a random walk method to determine node neighborhoods. However, when dealing with numerous neighboring nodes, using all neighbors for aggregation becomes inefficient. To tackle this challenge, GraphSAGE [49] samples a fixed number of neighbors for each node, significantly reducing computational costs. Additionally, the graph attention network (GAT) [50] introduces an

attention mechanism to learn the weights between nodes, differentiating aggregation based on the importance of adjacent nodes. Nevertheless, these two branches of GCN mainly focus on learning node representations, making them incapable of generating more abstract graph representations due to the lack of subsampling operations.

## 2.2. Graph Pooling

Pooling operations in convolutional neural networks (CNNs) effectively reduce the size of input feature graphs and the number of training parameters, thereby enhancing model performance. Similarly, graph pooling techniques can also enhance model performance by suitably compressing nodes within a graph. Existing graph pooling methods can be categorized into global pooling and hierarchical pooling based on their roles in GNNs [51]. Global graph pooling captures node information representing the entire graph's characteristics, often employed to link GNN layers with classifiers for more discriminative feature information in classification tasks. Common global graph pooling approaches include averaging and summing [45], attention mechanisms [52], feature sorting, and splicing [53]. Hierarchical graph pooling mirrors traditional CNN-based pooling, focusing on capturing the topological structure of a given graph while learning multi-scale information through structural modifications. Hierarchical graph pooling is analogous to traditional convolutional neural network pooling, with a primary focus on capturing the topological structure of the graph and learning multi-scale information by modifying the graph's structure. Hierarchical pooling can be broadly classified into two approaches: graph coarsening [54] and node selection [55]. Graph coarsening methods aim to map nodes to distinct clusters and condense the graph into a higher-level substructure. On the other hand, node selection methods involve evaluating individual nodes or edges between nodes to determine which should be merged or eliminated.

## 3. Methodology

In this section, we initially outline the overall framework of SPGAU. Subsequently, we elucidate the method for constructing superpixel graphs. Moreover, we provide a concise introduction to the attention-based graph convolution module employed in our approach. Finally, we present the spatial pooling module, which interconnects the graph convolutional modules to evolve the graph structure.

### 3.1. Overview of the SPGAU

We propose a spatial-pooling-based graph attention neural network (SPGAU), as illustrated in Figure 1. Initially, the original HSI is divided into local patches (superpixels). Simultaneously, to enhance the learning capacity for pixel-level subtle features, a 1D-CNN is introduced for the efficient extraction of superpixel spectral features. Furthermore, an attention mechanism is integrated into the graph convolution process to adaptively aggregate feature efficiency. The similarity between nodes provides a similarity measure for the node merging in the spatial pooling module. Through the superpixel merging process, spatial pooling generates a more abstract graph structure by merging nodes with high similarity. In contrast to fusing features of different scales concurrently, the feature fusion of our spatial-pooling-based graph attention U-Net (SPGAU) unfolds gradually from shallow to deep layers, optimally using the U-Net structure to fuse information across various scales, naturally aligning with the pixel-level classification task in HSI.

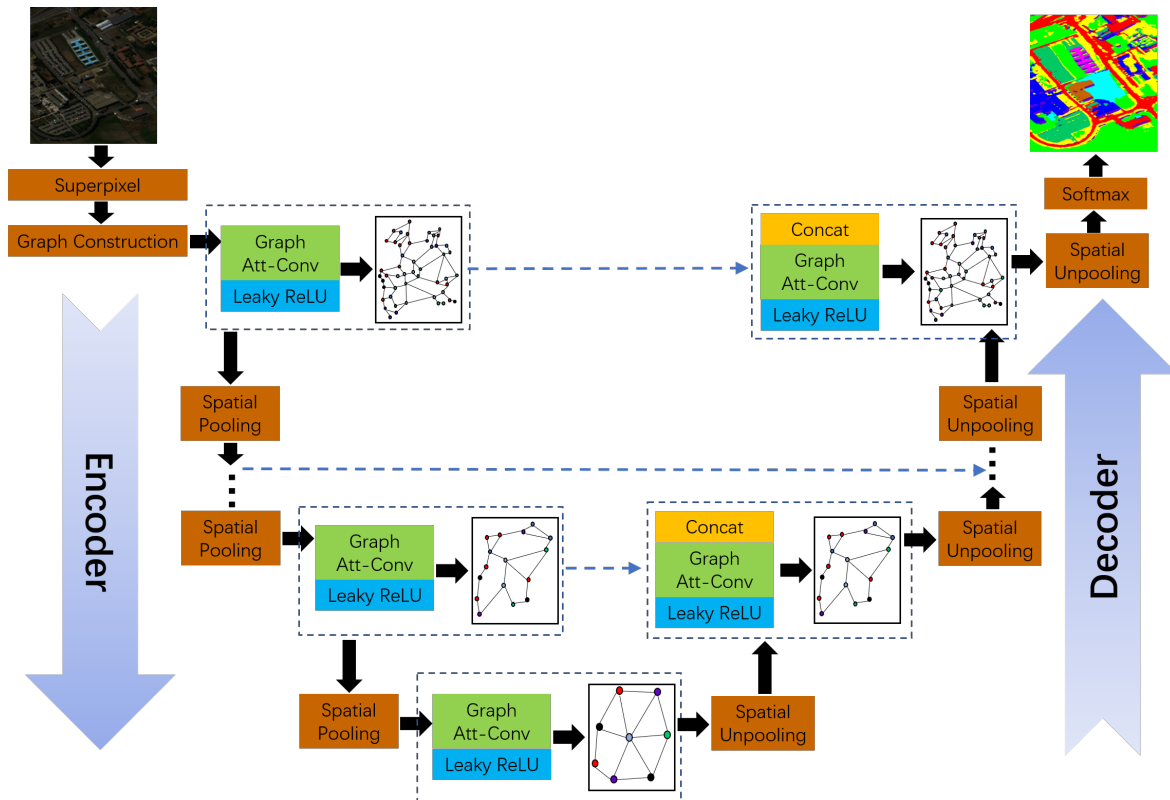


**Figure 1.** The conceptual architecture of SPGAU. Firstly, the superpixels obtained by hyperspectral image segmentation (SLIC) are used as graph nodes, the hidden states of the nodes are the feature vectors extracted by the convolutional neural network, and the graph is constructed according to the spatial relationship between the superpixels. Then, multi-scale feature information is extracted by graph evolution through a set of stacked graph attention convolution and spatial pooling modules. Finally, the segmentation result of HSI is obtained by the softmax function. The blue dashed line indicates the skip connection, and the black arrow is the direction of the information flow.

### 3.2. Graph Construction

Hyperspectral images typically encompass hundreds of thousands of pixels, with each pixel containing hundreds of spectral channels. The vast amount of image data exerts significant computational demands on HSI classifiers. To mitigate this computational complexity while retaining the local structure and feature information in the image, we apply the simple linear iterative clustering (SLIC) [56] technique for segmenting HSI into spatially connected superpixels. Each superpixel is considered a node; then, an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is constructed depending on the adjacency relationships between superpixels in the HSI. Here,  $\mathcal{V}$  denotes the nodes, and  $\mathcal{E}$  represents the edges. Typically, these are portrayed using the superpixel adjacency matrix  $\mathbf{A} \in \mathbb{R}^{F \times F}$  and node feature matrix  $\mathbf{H} \in \mathbb{R}^{F \times D}$ , where  $F$  represents the number of superpixels, and  $D$  is the feature dimension. According to the position relationship of the superpixels in graph  $\mathcal{G}$ , the superpixel adjacency matrix  $\mathbf{A}$  can be defined as

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if } S_i \text{ and } S_j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where  $S_i$  represents the  $i$ th superpixel, consisting of several HSI pixels  $\{p_1, p_2, \dots, p_{N_i}\}$ , and  $N_i$  signifies the number of pixels in  $S_i$ .

In practice, the noise of the hyperspectral images interferes with the reliability of the features of the superpixels. To suppress the noise, we use 1D-CNN to perform convolution

operations on the spectral channel of HSI, and the feature  $\mathbf{X}_b(p_0)$  of the  $b$ th spectral channel of the pixel  $p_0$  is denoted as

$$\mathbf{X}_b(p_0) = \text{RELU}(\mathbf{W}_b \cdot \hat{\mathbf{X}}_b(p_0) + a_b), \quad (2)$$

where  $\hat{\mathbf{X}}_b(p_0)$  is the original spectral value of the  $b$ th channel at pixel  $p_0$ ,  $\mathbf{W}_b$  and  $a_b$  are the learnable weights and bias. Then, the spectral feature vector  $\mathbf{h}(p_0)$  can be written as

$$\mathbf{h}(p_0) = (\mathbf{X}_1(p_0), \mathbf{X}_2(p_0), \dots, \mathbf{X}_B(p_0)), \quad (3)$$

where  $B$  is the number of spectral channels. Finally, we compute the average features of a superpixel as a node vector, and the graph node feature matrix  $\mathbf{H} \in \mathbb{R}^{F \times D}$  can be mathematically calculated as

$$\mathbf{H} = \left[ \frac{1}{N_1} \sum_{f=1}^{N_1} \mathbf{h}_f^1, \frac{1}{N_2} \sum_{f=1}^{N_2} \mathbf{h}_f^2, \dots, \frac{1}{N_F} \sum_{f=1}^{N_F} \mathbf{h}_f^F \right]^T. \quad (4)$$

### 3.3. Graph Attention Convolutional Neural Network

We use superpixels as bridges to transform HSI into graph structures. GCN [57] is a graph-based multi-layer neural network, which can directly perform convolution operations on graph data and guide the flow of information through nodes. To better aggregate neighbor information, we introduce an attention mechanism into GCN to adaptively adjust the degree of information aggregation by calculating the similarity between every two neighboring nodes. Considering the similarity between nodes, the similarity coefficient  $a_{ij}$  between node  $i$  and node  $j$  can be written as

$$a_{ij} = \text{Softmax} \left( \left( \mathbf{h}_i^T \mathbf{W}_a \right) \left( \mathbf{h}_j^T \mathbf{W}_a \right)^T \right), \quad (5)$$

where  $\mathbf{W}_a$  are the learnable parameters; the similarity of node information is represented by calculating the inner product of two nodes. To facilitate the subsequent calculation, the softmax function can ensure the numerical stability of the similarity coefficient  $a_{ij}$  [40]. Furthermore, considering the neighbor information, the attention coefficient  $k_{ij}$  is calculated by using the normalization function,

$$k_{ij} = \text{Softmax}(a_{ij}) = \frac{\exp(a_{ij})}{\sum_{k=1}^{k \in \mathcal{N}_i} \exp(a_{ik})}, \quad (6)$$

where  $\mathcal{N}_i$  signifies the neighborhood of node  $i$ . To measure the similarity between adjacent nodes adaptively, we introduce an attention mechanism. Therefore, we define the attention matrix  $\hat{\mathbf{A}}$  as

$$\hat{\mathbf{A}} = \mathbf{K} \odot \mathbf{A} + \zeta \mathbf{I}, \quad (7)$$

where  $\odot$  is the Schur product,  $\mathbf{K} \in \mathbb{R}^{F \times F}$  denotes the attention coefficient matrix,  $\zeta$  is a learnable parameter, and  $\mathbf{I}$  is the identity matrix to ensure learning stability [57]. In addition, the attention-based graph convolution  $f_G$  can be expressed as

$$f_G(\mathbf{H}, \hat{\mathbf{A}}) = \sigma \left( \mathbf{D}^{-\frac{1}{2}} \hat{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}} \mathbf{H} \mathbf{W} + \mathbf{B} \right), \quad (8)$$

where  $\sigma(\cdot)$  is the leaky ReLU [58],  $\mathbf{D}$  is the degree matrix of  $\hat{\mathbf{A}}$ , and  $\mathbf{W}$  and  $\mathbf{B}$  are learnable parameters.

### 3.4. Spatial Pooling

According to the current research, graph convolution is a very effective method for the spectral-spatial feature extraction task. However, there are still some problems to be

solved; for instance, not all nodes contribute significantly to the classification process, and there is a potential for significant redundancy in calculations when dealing with large spatial sizes of HSI blocks. Therefore, reducing similar nodes in the graph can effectively reduce computational resources. In addition, the more simplified topology obtained by merging similar nodes can also provide feature information at different scales. For HSI classification, multi-scale features are crucial. Therefore, we use evolvable spatial pooling to address the above challenges. The core idea is to use the progressive merging relationship of superpixels to obtain superpixel graphs of different scales (the merging process is shown in Figure 2). From the perspective of graphics, spatial pooling can merge similar nodes to obtain a more abstract and reliable topology. From the feature perspective, spatial pooling is another screening of the features extracted by convolution, which can retain more high-quality features.

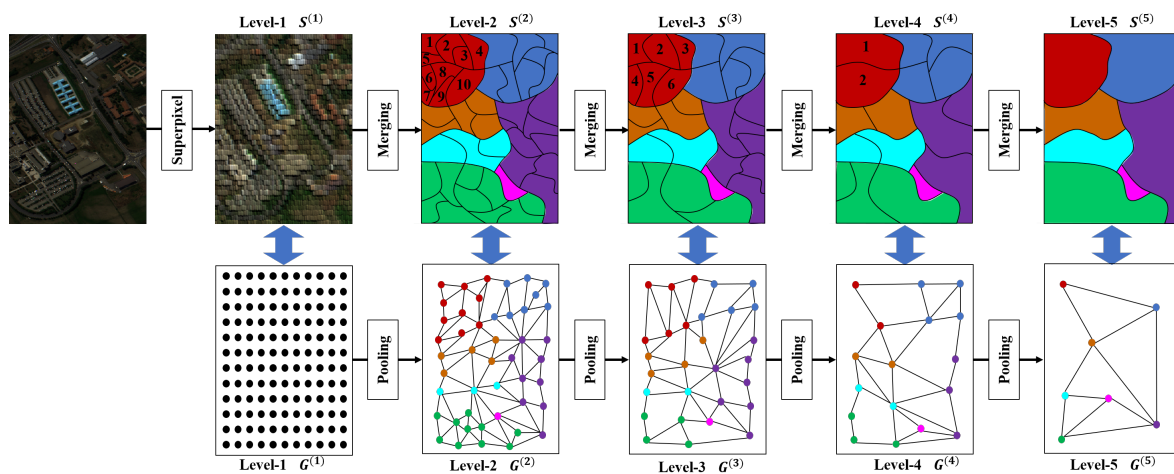


**Figure 2.** Progressive merging of superpixels and evolution of graph structure, where the top half is the superpixel merging process. Different colors represent different classes. We take the red region as an example to encode the superpixels and intuitively show the process of superpixel merging. Correspondingly, the bottom half is the graph structure evolution process inspired by the idea of superpixel merging.

Specifically, the pooling process involves merging several graph nodes in graph  $\mathcal{G}^l$  with merging probability to construct a new graph structure  $\mathcal{G}^{l+1}$ . The process of graph pooling lacks a definite evolutionary path, making it extremely challenging to enumerate all possible evolutionary graphs for evaluation. Hence, we employ a random mechanism to discover an optimal graph pooling result. The graph evolution process of our spatial convolution module is shown in Figure 3.

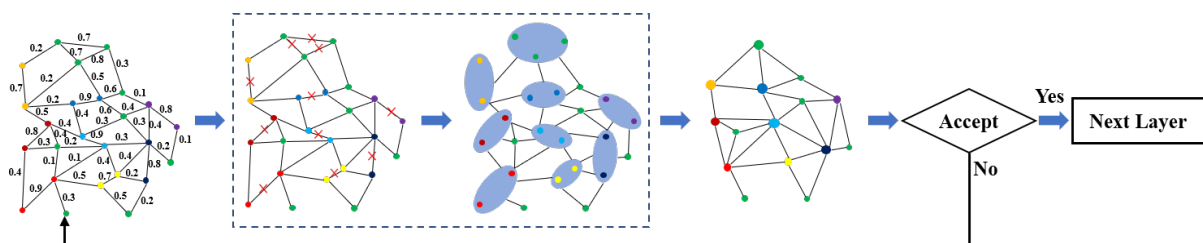


**Figure 3.** Illustration of the spatial pooling process. Given the calculated probabilities for merging nodes, the space pooling process involves multiple attempts to merge nodes and generate a new graph until an accepted graph is obtained. The generation of a new graph entails randomly merging some nodes with high merging probability, resulting in the creation of new edges.

First, we randomly select  $N$  edges. The edge elimination probability (EEP) between node  $i$  and node  $j$  can be calculated by

$$EEP_{ij} = \begin{cases} 1, & \text{if } label_i = label_j, \\ a_{ij}, & \text{otherwise.} \end{cases} \quad (9)$$

where we define the EEP of nodes with the same label as 1 and the EEP of nodes with different labels as the similarity of two nodes, which can be calculated by Formula (6). Further, the merging probability (MP) of the whole graph can be obtained as

$$MP = \prod_{\langle i,j \rangle \in \mathcal{E}^l \setminus \mathcal{E}^{l+1}} EEP_{ij} \quad (10)$$

This random merging method can also effectively avoid getting trapped in a local optimum. To evaluate the evolution between two graphs, we define the acceptance probability (AP) from graph  $\mathcal{G}^l$  to graph  $\mathcal{G}^{l+1}$  as

$$AP = \min(1, MP \cdot \varepsilon) \quad (11)$$

$$\varepsilon = \frac{P(\mathcal{G}^{l+1} | \mathbf{X}; \mathbf{H}, \mathbf{A})}{P(\mathcal{G}^l | \mathbf{X}; \mathbf{H}, \mathbf{A})}, \quad (12)$$

where  $\varepsilon$  is the merging dynamic coefficient, and  $P(\mathcal{G}^{l+1} | \mathbf{X}; \mathbf{H}, \mathbf{A})$  and  $P(\mathcal{G}^l | \mathbf{X}; \mathbf{H}, \mathbf{A})$  denote the posterior probabilities of the graph structures  $\mathcal{G}^{l+1}$  and  $\mathcal{G}^l$ , respectively. The dynamic coefficient can play a certain supervision role in the spatial pooling process and promote the evolution of a more accurate graph structure. To save time, cost, and computing resources and to ensure efficient completion of the exploration of pooling nodes, we can empirically set the upper bound of the sampling trials, for example, 20 in our experiment. After eliminating the edges with the above determination, the operations of pooling and unpooling can be further defined. First, we define  $\mathbf{T}^l \in \mathbb{R}^{H \times W}$  as an evolutionary matrix between  $\mathcal{G}^l$  and  $\mathcal{G}^{l+1}$ , computed as

$$\mathbf{T}_{i,j}^l = \begin{cases} 1, & \text{if } S_i^{l-1} \subseteq S_j^l, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Since node information is merged in the process of space pooling, the normalized association matrix  $\hat{\mathbf{T}}^l$  is defined as

$$\hat{\mathbf{T}}_{i,j}^l = \frac{\mathbf{T}_{i,j}^l}{\sum_m \mathbf{T}_{m,j}^l} \quad (14)$$

Next, the spatial pooling function can be denoted as

$$\mathbf{H}^l = \text{Pooling}(\mathbf{H}^{l-1}) = (\hat{\mathbf{T}}^l)^T \mathbf{H}^{l-1} \quad (15)$$

The unpooling function is expressed as

$$\mathbf{H}^{l-1} = \text{UpPooling}(\mathbf{H}^l) = \mathbf{T}^l \mathbf{H}^l \quad (16)$$

As shown in Figure 1, after multiple convolution-pooling modules, the final output  $\mathbf{H}^L$  passes through a fully connected layer with the softmax classifier to obtain the final classification result:

$$\mathbf{Z} = \text{Softmax}(\mathbf{H}^L \mathbf{W} + b), \quad (17)$$

where  $\mathbf{Z}$  is the classification result matrix. To optimize the proposed SPGAU, we use the cross-entropy loss function as

$$\mathcal{L} = - \sum_{g \in \mathbf{y}_G} \sum_{f=1}^C \mathbf{Y}_{gf} \ln \mathbf{Z}_{gf}, \quad (18)$$

where  $\mathbf{y}_G$  denotes the index set of labeled pixels,  $C$  represents the number of classes, and  $\mathbf{Y}$  represents the label matrix. Similarly to [59], our network parameters are acquired through training by the Adam optimizer.

#### 4. Experiments

This section summarizes three publicly available HSI datasets to validate the effectiveness of the proposed SPGAU. Then, the experimental settings are introduced. SPGAU was compared to seven effective HSI classification approaches: 3-D CNN [60], GCN [46], MDGCN [61], SSRN [62], SSGCN [63], HybridSN [64], and SVM [9]. Finally, the classification performance was evaluated using metrics such as the overall accuracy (OA), the average accuracy (AA), and the Kappa statistic (Kapp).

##### 4.1. Dataset Description

Three HSI classification datasets were selected for our experiments: the Indian Pines dataset (IP) [60], the Pavia University dataset (PU) [65], and the WHU-HiLongKou dataset (WHU) [66]. We summarize the datasets in Table 1. Specifically, Tables 2–4 show the number of training and testing samples for each class of the three datasets, respectively.

**Table 1.** Details of the three HSI classification datasets used for experiments.

| Dataset | Spectral Bands | Resolution | Wavelength Range      | Size             | Samples (Train/Test) | Class | Sensor         |
|---------|----------------|------------|-----------------------|------------------|----------------------|-------|----------------|
| IP      | 200            | 20 m       | 0.4–2.5 $\mu\text{m}$ | $145 \times 145$ | 567/9790             | 16    | AVIRIS         |
| PU      | 103            | 1.3 m      | 0.4–0.9 $\mu\text{m}$ | $610 \times 340$ | 2135/38,505          | 9     | ROSIS          |
| WHU     | 270            | 0.463 m    | 400–1000 nm           | $550 \times 400$ | 10,231/193,283       | 9     | Nano-Hyperspec |

**Table 2.** Category information of the Indian Pines dataset.

| Class                         | Training | Testing |
|-------------------------------|----------|---------|
| Alfafa                        | 4        | 55      |
| Corn-notill                   | 80       | 1344    |
| Corn-mintill                  | 45       | 799     |
| Corn                          | 12       | 220     |
| Grass-pasture                 | 28       | 476     |
| Grass-trees                   | 44       | 721     |
| Grass-pasture-mowed           | 7        | 25      |
| Hay-windrowed                 | 25       | 455     |
| Oats                          | 3        | 22      |
| Soybean-notill                | 50       | 910     |
| Soybean-mintill               | 122      | 2323    |
| Soybean-clean                 | 33       | 578     |
| Wheat                         | 16       | 199     |
| Woods                         | 66       | 1219    |
| Buildings-Grass-Trees-Drivers | 23       | 355     |
| Stone-Steel-Towers            | 9        | 89      |

**Table 3.** Category information of the Pavia University dataset.

| Class                | Training | Testing |
|----------------------|----------|---------|
| Water                | 332      | 5970    |
| Trees                | 940      | 16,775  |
| Asphalt              | 110      | 1892    |
| Self-blocking bricks | 155      | 2766    |
| Bitumen              | 68       | 1209    |
| Tiles                | 225      | 4527    |
| Shadows              | 67       | 1196    |
| Meadows              | 188      | 3317    |
| Bare soil            | 50       | 853     |

**Table 4.** Category information of the WHU-Hi-LongKou dataset.

| Class               | Training | Testing |
|---------------------|----------|---------|
| Corn                | 1726     | 32,612  |
| Cotton              | 419      | 7913    |
| Sesame              | 152      | 2863    |
| Broad-leaf soybean  | 3161     | 59,734  |
| Narrow-leaf soybean | 208      | 3922    |
| Rice                | 593      | 11,201  |
| Water               | 3353     | 63,367  |
| Roads and houses    | 357      | 6731    |
| Mixed weed          | 262      | 4940    |

#### 4.2. Experimental Settings

As illustrated in Section 3, the initial level of superpixel segmentation involves image convolutions, while graph convolutions are implemented in subsequent levels. The learning rate is set to  $5 \times 10^{-5}$  with 500 epochs, and the graph depth is 5. Specifically, Table 5 describes the details of SPGAU's network configuration. The experiments are conducted on a computer equipped with a Nvidia GeForce GTX 2080Ti 11G GPU and an i9-10900K CPU, boasting 64 GB of memory (NVIDIA, Santa Clara, CA, USA). The software environment comprises Python 3.8.5 and PyTorch 1.7.1.

**Table 5.** Network configuration of the SPGAU.

| Level | Convolution | Encoder                 | Decoder                 | Nodes |
|-------|-------------|-------------------------|-------------------------|-------|
| 1     | 1D-CNN      | $5 \times 5 \times 128$ | $5 \times 5 \times 128$ | –     |
| 2     | GCN         | 64                      | 64                      | 2048  |
| 3     | GCN         | 32                      | 32                      | 1024  |
| 4     | GCN         | 16                      | 16                      | 512   |
| 5     | GCN         | 8                       | –                       | 256   |

#### 4.3. Classification Results

We employed three widely recognized datasets to evaluate the research methods for HSI classification performance. We compared SPGAU with seven HSI classification methods across these benchmark datasets. We set each experiment to be repeated ten times and report the mean and standard deviation for each metric. Tables 6–8 present the per-class accuracy, OA, AA, and Kappa coefficients obtained by eight methods on the three datasets, highlighting the best results in bold. Figures 4–6 illustrate the HSI classification results obtained by these methods.

Among the various network models, our proposed SPGAU emerges as the top-performing method across all three datasets. Notably, the accuracy for all classes exceeds 90%, with most being optimal. Particularly on the IP and WHU datasets, SPGAU attains an OA, AA, and Kappa coefficient each exceeding 98%. As Table 7 reveals, CNN-based methods outperform GCN-based approaches. Compared with 3D-CNN, SSRN, featuring

the residual module, improves OA by 3.06%, while HybridSN achieves an impressive OA of 96%, attributed to its attention mechanism. However, not all GCN-based methods perform poorly. As depicted in Table 7, MDGCN also demonstrates substantial promise, with an OA of 95.24%, positioning it as the second-best performer, just behind SPGAU. The incorporation of multi-scale graph concepts in both SPGAU and MDGCN underscores the importance of multi-scale feature extraction for learning additional spatial features from HSI data. Moreover, because of the relatively large size of the WHU-Hi-LongKou dataset, MDGCN faces challenges in scaling to such data, which is why it is not included in the comparison presented in Table 7. Further insights from the experimental results of SVM in Table 6–8 indicate that machine learning methods can achieve the performance of the deep learning methods. It is worth noting that in Table 6, SVM achieves the optimal accuracy in four categories, second only to the 6-item optimality achieved by our method. However, because the classification accuracy of “Corn-mintill” and “Soybean-mintill” of SVM is less than 80%, the OA of SVM is only 88.59%, which is nearly 10% lower than the 98.56% of SPGAU. This not only affects OA, but the poor Kappa coefficient also reflects the weak spatial consistency between the prediction results of SVM and the actual situation. From the perspective of the classification results in Figures 4–6, SPGAU’s classification maps exhibit smoother visual effects with fewer classification errors. For instance, consider the red box section in Figure 4, where “Soybean-min-till” and “Corn-no-till” share similar spectral characteristics. This results in some pixels from “Soybean-min-till” being misclassified as “Corn-no-till” across all classification maps. Additionally, CNN-based methods produce pepper-like errors in certain regions, owing to the absence of spatial context information. In comparison, the classification maps generated by MDGCN and SPGAU, which leverage global information, exhibit stronger spatial correlations. In summary, SPGAU significantly outperforms other algorithms across the three datasets, owing to two key factors as follows:

1. Attention Mechanism Module: Adding the attention mechanism to graph convolution improves the efficiency of feature aggregation.
2. Spatial Pooling Module: The spatial pooling module combines similar nodes to generate graph structures at different levels. Building upon this foundation, it extracts more comprehensive global spatial context and spatial details by learning feature information at various scales.

These two modules enrich the feature space from both local and global perspectives, yielding multi-scale features that are well-suited for diverse land cover types. Consequently, SPGAU achieves classification results that are remarkably consistent with the ground truth labels.

**Table 6.** Comparison results on the Indian Pines d.

| Class   | 3-DCNN       | GCN                 | MDGCN               | SSRN                | SSGCN               | HybridSN            | SVM                 | SPGAU               |
|---------|--------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| 1       | 92.80 ± 7.53 | 76.43 ± 2.38        | <b>100.0 ± 0.00</b> | 86.22 ± 9.37        | <b>100.0 ± 0.00</b> | <b>100.0 ± 0.00</b> | 99.23 ± 0.98        | <b>100.0 ± 0.00</b> |
| 2       | 88.24 ± 3.12 | 86.74 ± 3.12        | 80.18 ± 0.84        | <b>96.49 ± 1.54</b> | 84.23 ± 2.65        | 95.19 ± 1.42        | 90.77 ± 3.81        | 95.46 ± 1.89        |
| 3       | 92.66 ± 3.03 | <b>100.0 ± 0.00</b> | 98.26 ± 0.00        | 99.12 ± 0.78        | 82.47 ± 5.97        | 97.92 ± 2.30        | 77.85 ± 3.34        | 98.33 ± 0.25        |
| 4       | 91.96 ± 6.98 | 92.11 ± 2.67        | 98.57 ± 0.00        | 96.78 ± 3.42        | 93.15 ± 1.73        | 98.92 ± 1.14        | <b>99.20 ± 0.97</b> | 98.93 ± 0.78        |
| 5       | 95.55 ± 1.09 | 91.48 ± 0.82        | 95.14 ± 0.33        | 98.31 ± 1.06        | <b>97.24 ± 1.46</b> | 96.57 ± 1.86        | 87.54 ± 2.28        | 96.21 ± 0.42        |
| 6       | 96.47 ± 1.51 | 93.21 ± 0.98        | 97.16 ± 0.57        | 93.27 ± 4.46        | 97.81 ± 1.10        | 93.22 ± 3.87        | <b>98.80 ± 0.53</b> | 97.32 ± 0.55        |
| 7       | 98.70 ± 2.56 | 88.20 ± 1.66        | 100.0 ± 0.00        | 98.49 ± 2.97        | 92.13 ± 0.74        | 98.88 ± 2.57        | <b>100.0 ± 0.00</b> | <b>100.0 ± 0.00</b> |
| 8       | 98.38 ± 1.55 | 79.94 ± 4.38        | 98.89 ± 0.00        | 98.38 ± 2.92        | 98.71 ± 1.12        | 99.13 ± 0.72        | <b>99.70 ± 0.28</b> | 98.12 ± 0.34        |
| 9       | 97.62 ± 2.14 | 82.53 ± 2.17        | <b>100.0 ± 0.00</b> | 80.21 ± 20.34       | 99.56 ± 0.68        | 76.47 ± 20.60       | 99.12 ± 0.31        | <b>100.0 ± 0.00</b> |
| 10      | 91.69 ± 3.22 | <b>98.77 ± 2.62</b> | 90.02 ± 1.02        | 95.27 ± 3.11        | 93.85 ± 3.81        | 94.58 ± 1.48        | 90.21 ± 4.23        | 91.03 ± 0.77        |
| 11      | 96.98 ± 1.94 | 91.68 ± 0.78        | 93.35 ± 1.47        | <b>99.29 ± 0.58</b> | 84.74 ± 3.01        | 98.07 ± 0.76        | 76.10 ± 5.82        | 92.96 ± 1.58        |
| 12      | 84.47 ± 3.11 | 93.83 ± 2.51        | 93.05 ± 2.30        | 98.68 ± 1.20        | 80.11 ± 5.07        | 95.92 ± 2.75        | 87.89 ± 2.36        | <b>95.52 ± 1.43</b> |
| 13      | 96.77 ± 3.32 | 91.80 ± 1.23        | <b>100.0 ± 0.00</b> | 92.79 ± 5.16        | 99.25 ± 0.43        | 95.38 ± 8.03        | 99.68 ± 0.47        | 99.68 ± 0.42        |
| 14      | 98.68 ± 1.31 | 93.34 ± 1.77        | 99.72 ± 0.05        | 99.27 ± 0.77        | 96.62 ± 0.99        | 98.27 ± 3.13        | 97.52 ± 2.68        | <b>99.78 ± 0.38</b> |
| 15      | 92.14 ± 3.47 | 95.66 ± 4.52        | 99.72 ± 0.00        | 95.19 ± 1.64        | 87.20 ± 3.27        | 93.94 ± 2.79        | 99.41 ± 0.63        | <b>99.82 ± 0.33</b> |
| 16      | 95.26 ± 3.64 | 96.78 ± 1.92        | 95.71 ± 0.00        | 96.73 ± 3.22        | <b>99.37 ± 0.69</b> | 97.68 ± 1.25        | 99.29 ± 0.88        | 97.34 ± 0.75        |
| OA(%)   | 94.17 ± 1.03 | 86.37 ± 0.88        | 93.47 ± 0.38        | 97.23 ± 0.72        | 89.58 ± 1.21        | 96.47 ± 0.93        | 88.59 ± 1.62        | <b>98.56 ± 0.39</b> |
| AA(%)   | 94.27 ± 0.92 | 90.12 ± 1.09        | 96.24 ± 0.21        | 94.69 ± 1.64        | 91.93 ± 1.52        | 96.58 ± 1.88        | 94.22 ± 0.71        | <b>98.95 ± 0.68</b> |
| Kapp(%) | 93.09 ± 1.10 | 88.59 ± 1.21        | 92.55 ± 0.43        | 92.01 ± 0.83        | 88.27 ± 1.34        | 96.49 ± 0.93        | 86.55 ± 1.78        | <b>98.79 ± 0.54</b> |

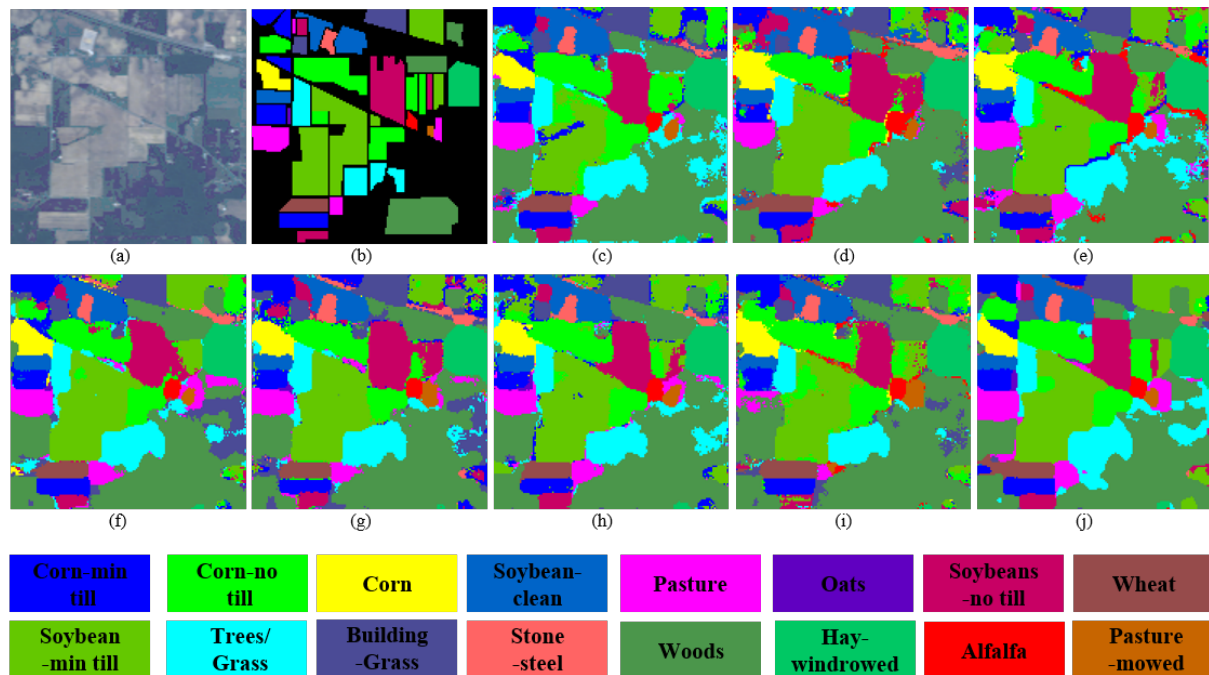


**Figure 4.** Classification maps obtained by eight different methods on the Indian Pines dataset. (a) False-color image. (b) Ground-truth map. (c) 3-DCNN. (d) GCN. (e) MDGCN. (f) SSRN. (g) SSGCN. (h) HybridSN. (i) SVM. (j) SPGAU.

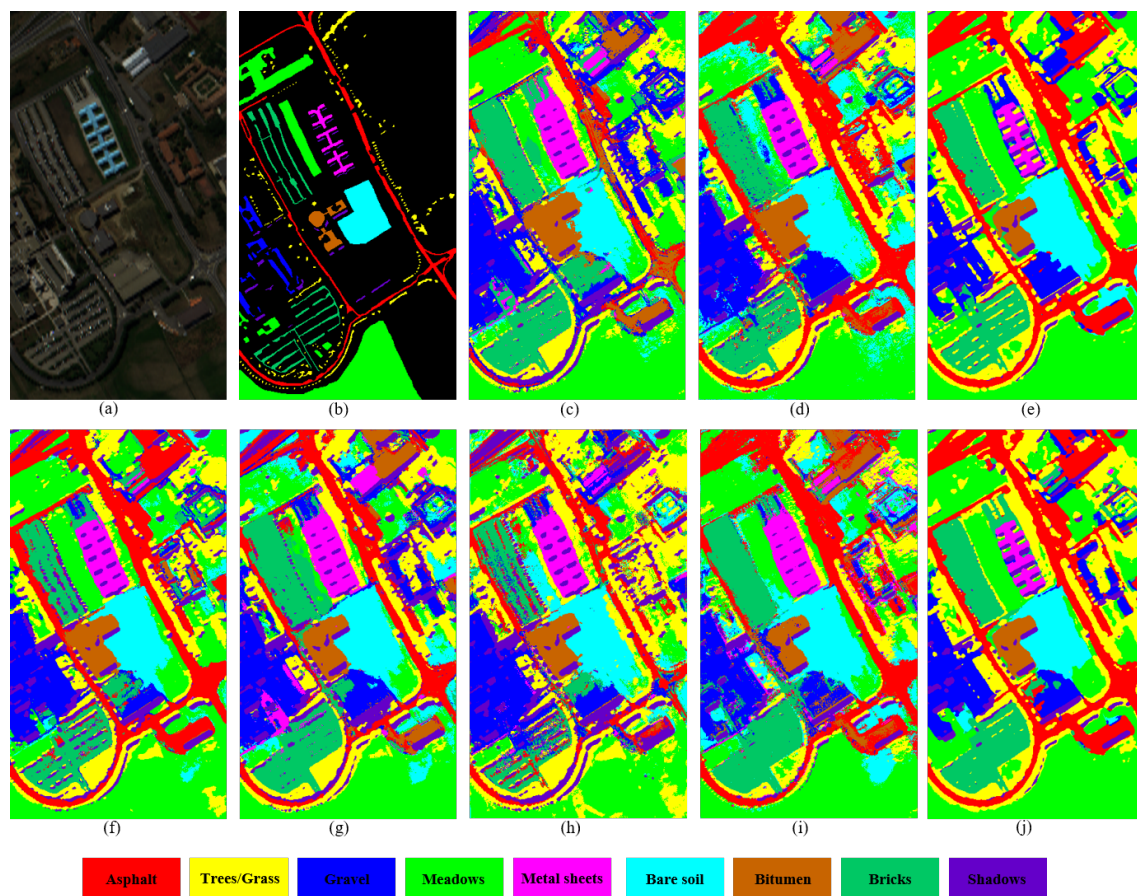
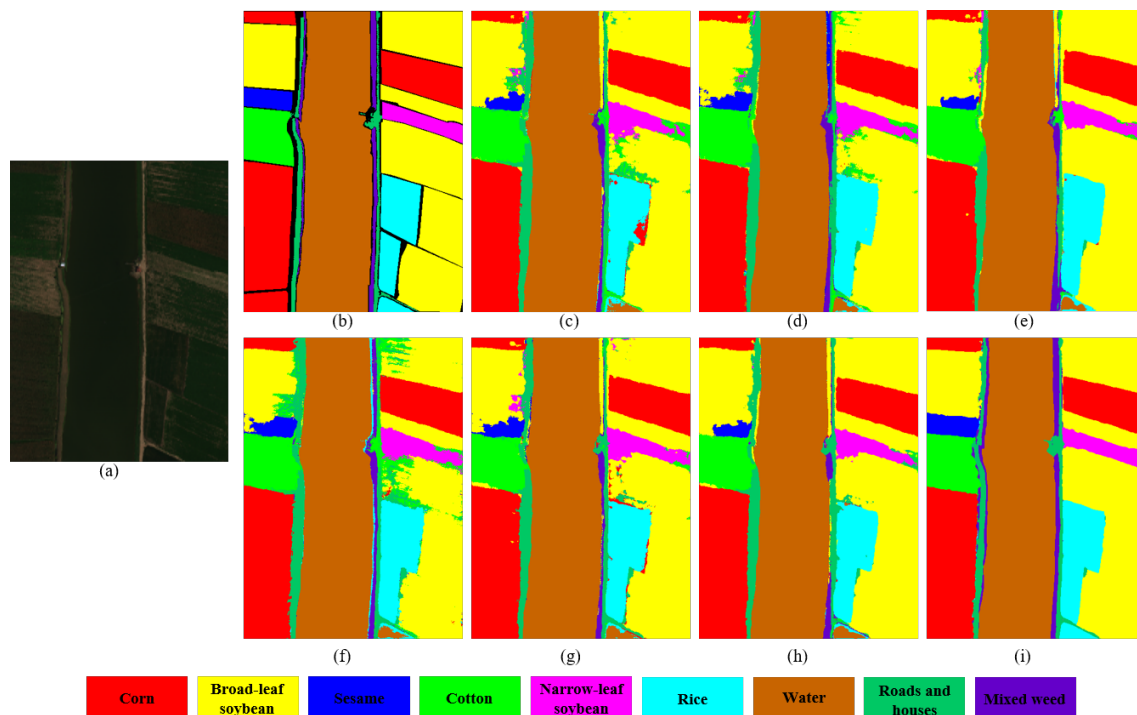


**Figure 5.** Classification maps obtained by eight different methods on the Pavia University dataset. (a) False-color image. (b) Ground-truth map. (c) 3-DCNN. (d) GCN. (e) MDGCN. (f) SSRN. (g) SSGCN. (h) HybridSN. (i) SVM. (j) SPGAU.

**Table 7.** Comparison results on the Pavia University dataset.

| Class   | 3-DCNN       | GCN                 | MDGCN               | SSRN                | SSGCN               | HybridSN     | SVM                 | SPGAU               |
|---------|--------------|---------------------|---------------------|---------------------|---------------------|--------------|---------------------|---------------------|
| 1       | 86.84 ± 1.23 | 79.88 ± 5.85        | 93.55 ± 0.37        | 92.47 ± 1.23        | 92.77 ± 3.98        | 75.44 ± 7.43 | 82.41 ± 3.07        | <b>100.0 ± 0.00</b> |
| 2       | 91.65 ± 0.42 | 90.70 ± 3.27        | 99.25 ± 0.23        | 97.08 ± 2.13        | 87.13 ± 4.46        | 84.78 ± 3.49 | 90.65 ± 3.07        | <b>99.76 ± 0.43</b> |
| 3       | 86.72 ± 1.98 | 81.70 ± 1.74        | 92.03 ± 0.24        | <b>97.12 ± 6.98</b> | 87.74 ± 4.88        | 73.09 ± 6.37 | 86.88 ± 4.27        | 94.30 ± 0.41        |
| 4       | 90.23 ± 0.83 | <b>98.67 ± 0.97</b> | 83.78 ± 1.55        | 91.56 ± 8.02        | 90.21 ± 1.08        | 73.77 ± 2.48 | 92.73 ± 2.13        | 90.28 ± 2.11        |
| 5       | 90.66 ± 1.08 | 96.39 ± 1.08        | 99.47 ± 0.09        | 97.57 ± 0.32        | <b>100.0 ± 0.00</b> | 82.66 ± 5.36 | <b>100.0 ± 0.00</b> | <b>100.0 ± 0.00</b> |
| 6       | 89.14 ± 0.72 | 79.89 ± 2.32        | 95.26 ± 0.50        | 95.19 ± 6.80        | 88.96 ± 2.08        | 76.92 ± 4.71 | 94.61 ± 4.26        | <b>95.78 ± 0.67</b> |
| 7       | 88.35 ± 1.56 | 81.25 ± 0.67        | <b>98.92 ± 1.04</b> | 97.27 ± 0.63        | 98.11 ± 1.64        | 87.53 ± 3.74 | 96.68 ± 1.73        | 98.89 ± 0.99        |
| 8       | 82.79 ± 4.51 | 92.82 ± 1.48        | 94.99 ± 1.33        | 92.45 ± 4.97        | 89.98 ± 3.52        | 88.27 ± 1.70 | 94.61 ± 3.38        | <b>96.26 ± 1.35</b> |
| 9       | 89.95 ± 0.27 | <b>97.74 ± 1.23</b> | 81.03 ± 0.49        | 97.05 ± 0.94        | 98.69 ± 0.57        | 85.93 ± 2.84 | 94.65 ± 0.34        | 95.03 ± 0.94        |
| OA(%)   | 88.72 ± 1.32 | 89.32 ± 1.71        | 95.68 ± 0.22        | 95.24 ± 2.06        | 89.77 ± 1.72        | 86.60 ± 1.44 | 90.72 ± 1.39        | <b>97.67 ± 0.24</b> |
| AA(%)   | 89.01 ± 1.43 | 89.06 ± 1.58        | 93.15 ± 0.28        | 95.41 ± 2.33        | 92.81 ± 0.64        | 84.81 ± 1.25 | 93.05 ± 0.77        | <b>96.21 ± 0.29</b> |
| Kapp(%) | 87.95 ± 1.61 | 89.17 ± 1.29        | 94.25 ± 0.29        | 96.62 ± 2.99        | 86.72 ± 2.09        | 84.26 ± 1.58 | 88.18 ± 1.74        | <b>96.89 ± 0.28</b> |

**Figure 6.** Classification maps obtained by seven different methods on the WHU-Hi-LongKou dataset. (a) False-color image. (b) Ground-truth map. (c) 3-DCNN. (d) GCN. (e) SSRN. (f) SSGCN. (g) HybridSN. (h) SVM. (i) SPGAU.**Table 8.** Comparison results on the WHU-Hi-LongKou dataset.

| Class   | 3-DCNN       | GCN          | MDGCN | SSRN                | SSGCN        | HybridSN     | SVM                 | SPGAU               |
|---------|--------------|--------------|-------|---------------------|--------------|--------------|---------------------|---------------------|
| 1       | 91.68 ± 0.87 | 95.86 ± 0.87 | –     | 95.68 ± 1.09        | 92.17 ± 1.20 | 96.17 ± 0.62 | 96.67 ± 1.33        | <b>100.0 ± 0.00</b> |
| 2       | 96.76 ± 0.35 | 98.97 ± 0.26 | –     | 99.61 ± 0.13        | 94.61 ± 0.16 | 99.87 ± 0.05 | 97.09 ± 2.04        | <b>99.54 ± 0.87</b> |
| 3       | 83.57 ± 3.42 | 84.48 ± 2.73 | –     | 82.33 ± 4.45        | 86.67 ± 2.99 | 95.82 ± 1.49 | <b>97.01 ± 9.09</b> | 95.89 ± 2.17        |
| 4       | 95.44 ± 0.52 | 97.02 ± 1.28 | –     | 98.12 ± 0.65        | 92.53 ± 0.55 | 93.66 ± 1.53 | 91.48 ± 4.96        | <b>100.0 ± 0.00</b> |
| 5       | 97.71 ± 0.33 | 99.01 ± 1.32 | –     | 99.36 ± 0.84        | 94.67 ± 0.54 | 95.01 ± 1.47 | 97.83 ± 0.17        | <b>99.92 ± 0.09</b> |
| 6       | 91.21 ± 1.98 | 95.66 ± 1.38 | –     | 95.82 ± 1.24        | 94.03 ± 1.37 | 96.65 ± 0.19 | 95.16 ± 3.09        | <b>98.68 ± 1.12</b> |
| 7       | 93.61 ± 0.99 | 89.79 ± 2.11 | –     | 83.89 ± 1.17        | 84.27 ± 3.65 | 95.93 ± 0.22 | 97.22 ± 0.19        | <b>98.87 ± 1.23</b> |
| 8       | 95.02 ± 1.73 | 92.69 ± 2.35 | –     | 88.81 ± 4.55        | 88.77 ± 2.03 | 94.49 ± 1.51 | 92.45 ± 1.98        | <b>100.0 ± 0.00</b> |
| 9       | 99.16 ± 0.92 | 98.08 ± 0.62 | –     | <b>99.96 ± 0.07</b> | 94.72 ± 0.46 | 94.07 ± 0.89 | 97.63 ± 0.51        | 99.07 ± 0.86        |
| OA(%)   | 94.27 ± 0.37 | 95.18 ± 0.39 | –     | 96.19 ± 0.39        | 92.88 ± 0.34 | 96.13 ± 0.19 | 95.41 ± 2.01        | <b>98.77 ± 0.19</b> |
| AA(%)   | 91.68 ± 0.52 | 94.51 ± 0.62 | –     | 93.87 ± 0.29        | 91.44 ± 0.41 | 95.37 ± 0.28 | 95.49 ± 2.19        | <b>98.16 ± 0.26</b> |
| Kapp(%) | 91.44 ± 0.48 | 95.08 ± 0.43 | –     | 95.04 ± 0.51        | 92.13 ± 0.38 | 95.86 ± 0.26 | 94.18 ± 2.47        | <b>98.38 ± 0.30</b> |

## 5. Discussion

In this section, we first discuss the influence of network depth on the proposed SPGAU, then analyze whether the classification performance of SPGAU is competitive with different numbers of labeled samples, and finally, we briefly compare and analyze the computational efficiency of SPGAU.

### 5.1. Influence of Network Depth

Graph convolutional networks typically employ no more than three layers to prevent over-smoothing. Our SPGAU overcomes this limitation by incorporating a spatial pooling module, allowing for increased network depth and simplified spatial topology. This enhancement forms a strong basis for extracting deep feature information. To evaluate the impact of network depth on SPGAU's classification performance, we select network depths ranging from 1 to 6 for the experiments. As depicted in Figure 7, increasing the network depth leads to improved classification accuracy, particularly when the depth reaches 2. This suggests that combining graph convolution with image convolution effectively enhances overall performance. On the PU and WHU datasets, deeper SPGAU leads to improved classification performance. However, for the Pavia University dataset, minimal improvement is observed beyond a depth of 2, mainly due to the presence of discontinuous and fragmented land covers, which limits the efficacy of advanced graph structures.

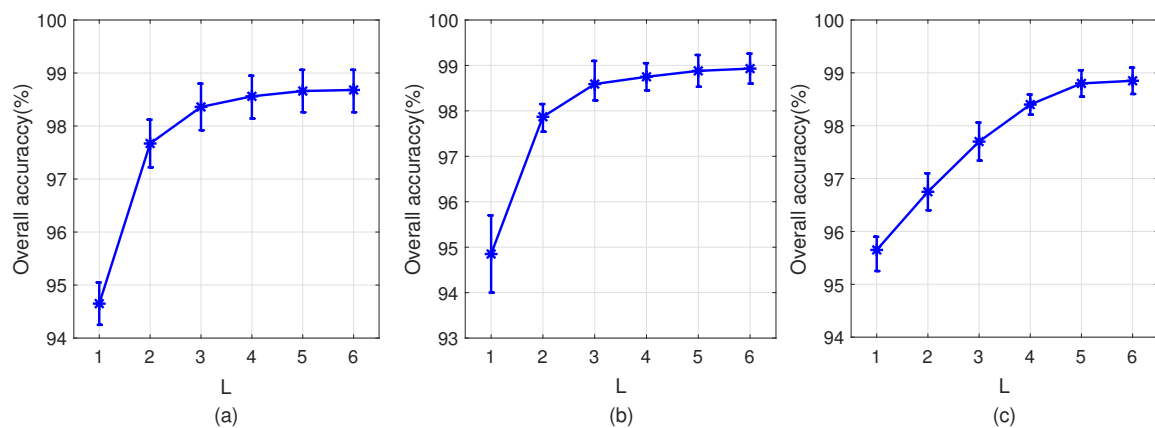


**Figure 7.** Classification accuracies of SPAGUN with different network depths. (a) Indian Pines. (b) Pavia University. (c) WHU-Hi-LongKou.

Beyond a network depth of five layers, the improvements in network classification performance become statistically insignificant, leading to the conclusion that the optimal number of network layers is five.

### 5.2. Impact of the Number of Labeled Examples

Given the high pixel resolution of HSI, manual annotation imposes substantial costs. In this experiment, we examine whether SPGAU maintains a robust classification capability with limited samples. Specifically, we set the training sample proportion for the IP dataset at between 1% and 6% per class and for the PU and WHU datasets at between 0.1% and 0.6% per class. Each experiment is repeated 10 times, and we employ the average OA index to assess the performance of both our proposed SPGAU and the comparative algorithm. The experimental results for the three datasets are presented in Figure 8.

Our SPGAU algorithm attains the highest OAs across all datasets. We notice an improvement in OA as the number of labeled instances increases. This suggests that incorporating more labeled data allows the model to acquire more practical information and enhances its training effectiveness. While the GCN-based method outperforms other models in the initial training stage, its OA does not rise significantly with the increase in samples in the later stage. Conversely, although the CNN-based methods exhibit lower

OAs with fewer training samples, they consistently exhibit a growth trend over time. This phenomenon underscores the distinct learning characteristics between these two models. By combining their strengths, our SPGAU model achieves superior classification performance, further validating its effectiveness. HybridSN, an attention mechanism-based method, witnesses a significant increase in OAs with increase in the label proportion.

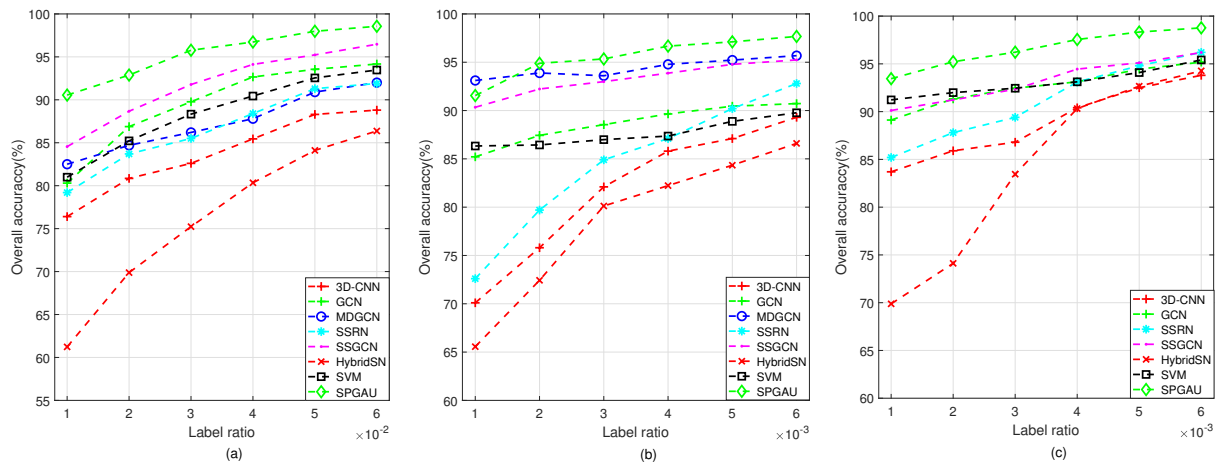


**Figure 8.** The classification performance of various methods with different training set ratios on three datasets. (a) Indian Pines. (b) Pavia University. (c) WHU-Hi-LongKou.

However, it does not reach the level of other methods, indicating that this method may not be suitable for scenarios with a few labeled samples. The MDGCN model secures the second-highest classification accuracy on both the IP and PU datasets. This can be attributed to its multi-scale mechanism, allowing it to learn diverse and informative features, demonstrating adaptability even with the limited labeled samples. Similarly, our SPGAU model, designed as a multi-scale feature extraction and fusion approach, highlights that incorporating diverse models and employing a multi-scale structure can enhance the effectiveness and robustness of HSI classification.

### 5.3. Running Time

We evaluated the model's time efficiency, a critical indicator for evaluating the algorithm's practicality. The experimental settings align with those outlined in Section 4.2. Table 9 reports the time cost (training and testing) of SPGAU on the three datasets. Specifically, machine learning-based SVM exhibits a characteristic of rapid training but slower testing, possibly attributed to the absence of a network structure suitable for GPU acceleration. CNNs are limited by their extensive parameter count, leading to significant time consumption. Notably, for CNN-based deep learning methods, network depth primarily influences processing speed. For instance, SSRN, an extension of the 3D-CNN model incorporating residual modules, results in more layers and longer training and testing times compared to 3D-CNN. Conversely, GCN-based methods do not exhibit an increase in time consumption with larger dataset scales. Interestingly, our proposed SPGAU shows higher time consumption than GCN and SSGCN on the Indian Pines dataset. Nevertheless, SPGAU delivers significantly improved classification performance compared to these two methods. In addition, on large-scale datasets, such as the PU dataset, SPGAU requires less time than GCN and SSGCN while achieving satisfactory results. Thus, our proposed spatial pooling module and graph attention convolution module effectively reduce redundant computations. Given that superpixel preprocessing substantially reduces the graph's size, SPGAU performs efficiently in terms of time consumption on large-scale datasets. In sum, our proposed SPGAU employs superpixel preprocessing technology to control the graph's scale, effectively reducing computational complexity.

**Table 9.** Comparison of running time between different methods.

| Dataset        | Time (s) | 3-DCNN | GCN    | MDGCN  | SSRN   | SSGCN  | HybridSN | SVM   | SPGAU  |
|----------------|----------|--------|--------|--------|--------|--------|----------|-------|--------|
| Indian Pines   | Train.   | 46.28  | 23.10  | 283.51 | 103.47 | 35.11  | 70.16    | 0.47  | 48.32  |
|                | Test.    | 1.87   | 1.94   | 3.87   | 2.55   | 2.26   | 12.03    | 1.54  | 2.19   |
| Pavia          | Train.   | 25.59  | 71.26  | 737.13 | 51.99  | 211.27 | 49.68    | 0.25  | 69.56  |
|                | Test.    | 5.12   | 9.31   | 22.78  | 8.67   | 16.18  | 19.73    | 4.01  | 7.35   |
| WHU-Hi-LongKou | Train.   | 59.54  | 129.36 | –      | 125.03 | 354.96 | 188.78   | 0.53  | 262.81 |
|                | Test.    | 26.62  | 17.93  | –      | 37.28  | 23.46  | 93.80    | 35.19 | 13.98  |

## 6. Conclusions

In this article, we established a novel SPGAU for HSI classification. Unlike conventional GCN models that operate on fixed graphs, SPGAU generates multi-scale graphs by iteratively merging adjacent and similar nodes in the graph. This progressive fusion of features of different scales, from rough to fine, enables the generation of more refined fusion features tailored to HSI classification tasks. Our spatial pooling method facilitates obtaining diverse scale graph structures through the merging of similar and adjacent graph nodes. This operation serves as subsampling, reducing computational complexity while acquiring abstract high-level graph structures. Subsequently, we employed a graph convolution operation with an attention mechanism to enhance the model's learning capability. The experimental results obtained on three extensively used HSI datasets indicated that SPGAU performs well on small sample experiments in addition to its superior classification performance. Moreover, our method balances the classification accuracy well with the running time. While our method has achieved promising results in HSI classification, the network training process utilized a single-scene HSI dataset, without attempting more complex cross-scene HSI classification research. In future research, we will focus on dynamic graphs and explore the fusion of GNNs with transfer models to cope with cross-scene HSI classification.

**Author Contributions:** Methodology, Q.D.; conceptualization, Y.D.; formal analysis, Q.D. and X.F.; software, Q.D.; validation, J.W. and Q.D.; writing—original draft preparation, Q.D.; writing—review and editing, C.Z. and Y.D.; supervision, C.Z. and F.P.; project administration, F.P. and X.F.; funding acquisition, F.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the National Natural Science Foundation of China under Grant 62261160575, Grant 61991414, and Grant 61973036; and in part by the Technical Field Foundation of the National Defense Science and Technology 173 Program under Grant 20220601053, Grant 20220601030.

**Data Availability Statement:** In this paper, three public hyperspectral datasets are selected. The Indian Pines dataset and the Pavia University dataset are available at [http://www.ehu.es/ccwintco/index.php/Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes), accessed on 2 August 2022. The WHU-Hi-LongKou dataset is available at [http://rsidea.whu.edu.cn/resource\\_WHUHi\\_sharing.htm](http://rsidea.whu.edu.cn/resource_WHUHi_sharing.htm), accessed on 2 August 2022.

**Acknowledgments:** The authors would like to thank Grupo de Inteligencia Computacional (GIC) and Intelligent Data Extraction, Analysis and Applications of Remote Sensing (RSIDEA). These two organizations have made many excellent hyperspectral datasets publicly available, which provided a lot of choices for our study. We would also like to thank the editors and reviewers for their valuable comments and help with this article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Hu, H.; Yao, M.; He, F.; Zhang, F. Graph neural network via edge convolution for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
2. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral-spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [\[CrossRef\]](#)

3. Ma, X.; Wang, H.; Wang, J. Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. *ISPRS J. Photogramm. Remote Sens.* **2016**, *120*, 99–107. [\[CrossRef\]](#)
4. Yang, G.; Huang, K.; Sun, W.; Meng, X.; Mao, D.; Ge, Y. Enhanced mangrove vegetation index based on hyperspectral images for mapping mangrove. *ISPRS J. Photogramm. Remote Sens.* **2022**, *189*, 236–254. [\[CrossRef\]](#)
5. Shitharth, S.; Manoharan, H.; Alshareef, A.M.; Yafaz, A.; Alkhiri, H.; Mirza, O.M. Hyper spectral image classifications for monitoring harvests in agriculture using fly optimization algorithm. *Comput. Electr. Eng.* **2022**, *103*, 108400.
6. Shimoni, M.; Haelterman, R.; Perneel, C. Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 101–117. [\[CrossRef\]](#)
7. Bai, J.; Huang, S.; Xiao, Z.; Li, X.; Zhu, Y.; Regan, A.C.; Jiao, L. Few-shot hyperspectral image classification based on adaptive subspaces and feature transformation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [\[CrossRef\]](#)
8. Rasti, B.; Hong, D.; Hang, R.; Ghamisi, P.; Kang, X.; Chanussot, J.; Benediktsson, J.A. Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 60–88. [\[CrossRef\]](#)
9. Bo, C.; Lu, H.; Wang, D. Hyperspectral image classification via JCR and SVM models with decision fusion. *IEEE Geosci. Remote Sens. Lett.* **2015**, *13*, 177–181.
10. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4085–4098. [\[CrossRef\]](#)
11. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [\[CrossRef\]](#)
12. Rodarmel, C.; Shan, J. Principal component analysis for hyperspectral image classification. *Surv. Land Inform. Sci.* **2002**, *62*, 115–122.
13. Zhang, Y.; Ma, Y.; Dai, X.; Li, H.; Mei, X.; Ma, J. Locality-constrained sparse representation for hyperspectral image classification. *Inform. Sci.* **2021**, *546*, 858–870. [\[CrossRef\]](#)
14. Ghamisi, P.; Benediktsson, J.A.; Ulfarsson, M.O. Spectral-spatial classification of hyperspectral images based on hidden Markov random fields. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2565–2574. [\[CrossRef\]](#)
15. Ye, Z.; Prasad, S.; Li, W.; Fowler, J.E.; He, M. Classification based on 3-D DWT and decision fusion for hyperspectral image analysis. *IEEE Geosci. Remote Sens. Lett.* **2013**, *11*, 173–177. [\[CrossRef\]](#)
16. Jia, S.; Zhuang, J.; Deng, L.; Zhu, J.; Xu, M.; Zhou, J.; Jia, X. 3-D Gaussian-Gabor feature extraction and selection for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8813–8826. [\[CrossRef\]](#)
17. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [\[CrossRef\]](#)
18. Fang, L.; He, N.; Li, S.; Ghamisi, P.; Benediktsson, J.A. Extinction profiles fusion for hyperspectral images classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1803–1815. [\[CrossRef\]](#)
19. Huang, H.; Shi, G.; He, H.; Duan, Y.; Luo, F. Dimensionality reduction of hyperspectral imagery based on spatial-spectral manifold learning. *IEEE Trans. Cybern.* **2019**, *50*, 2604–2616. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Fang, L.; Li, S.; Kang, X.; Benediktsson, J.A. Spectral-spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4186–4201. [\[CrossRef\]](#)
21. Ma, L.; Crawford, M.M.; Yang, X.; Guo, Y. Local-manifold-learning-based graph construction for semisupervised hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 2832–2844. [\[CrossRef\]](#)
22. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Du, Q.; Zhang, B. More diverse means better: Multimodal deep learning meets remote-sensing imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4340–4354. [\[CrossRef\]](#)
23. Fu, J.; Li, F.; Zhao, J. Real-Time Infrared Horizon Detection in Maritime and Land Environments Based on Hyper-Laplace Filter and Convolutional Neural Network. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–13. [\[CrossRef\]](#)
24. Aralikatti, R.C.; Pawan, S.; Rajan, J. A Dual-Stage Semi-Supervised Pre-Training Approach for Medical Image Segmentation. *IEEE Trans. Artif. Intell.* **2023**, *5*, 556–565 [\[CrossRef\]](#)
25. Zhang, Y.; Xu, S.; Hong, D.; Gao, H.; Zhang, C.; Bi, M.; Li, C. Multimodal Transformer Network for Hyperspectral and LiDAR Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–17. [\[CrossRef\]](#)
26. Yang, J.; Du, B.; Zhang, L. Overcoming the Barrier of Incompleteness: A Hyperspectral Image Classification Full Model. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–15. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *7*, 2094–2107. [\[CrossRef\]](#)
28. Dong, Y.; Liu, Q.; Du, B.; Zhang, L. Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification. *IEEE Trans. Image Process.* **2022**, *31*, 1559–1572. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Liang, L.; Zhang, S.; Li, J. Multiscale DenseNet meets with bi-RNN for hyperspectral image classification. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *15*, 5401–5415. [\[CrossRef\]](#)
30. Sellami, A.; Tabbone, S. Deep neural networks-based relevant latent representation learning for hyperspectral image classification. *Pattern Recognit.* **2022**, *121*, 108224. [\[CrossRef\]](#)
31. Yang, B.; Hu, S.; Guo, Q.; Hong, D. Multisource domain transfer learning based on spectral projections for hyperspectral image classification. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *15*, 3730–3739. [\[CrossRef\]](#)

32. Li, X.; Cao, Z.; Zhao, L.; Jiang, J. ALPN: Active-learning-based prototypical network for few-shot hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
33. Gao, K.; Liu, B.; Yu, X.; Yu, A. Unsupervised meta learning with multiview constraints for hyperspectral image small sample set classification. *IEEE Trans. Image Process.* **2022**, *31*, 3449–3462. [\[CrossRef\]](#)
34. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 1–12. [\[CrossRef\]](#)
35. Li, Y.; Zhang, H.; Shen, Q. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [\[CrossRef\]](#)
36. Yang, J.; Zhao, Y.Q.; Chan, J.C.W. Learning and transferring deep joint spectral–spatial features for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4729–4742. [\[CrossRef\]](#)
37. Chen, C.; Zhang, J.J.; Zheng, C.H.; Yan, Q.; Xun, L.N. Classification of hyperspectral data using a multi-channel convolutional neural network. In Proceedings of the the 14th International Conference on Intelligent Computing (ICIC), Wuhan, China, 15–18 August 2018; pp. 81–92.
38. Fang, J.; Zhu, Z.; He, G.; Wang, N.; Cao, X. Spatial-spectral Decoupling Framework for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [\[CrossRef\]](#)
39. Shi, H.; Cao, G.; Zhang, Y.; Ge, Z.; Liu, Y.; Yang, D. F3Net: Fast Fourier Filter Network for Hyperspectral Image Classification. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–18. [\[CrossRef\]](#)
40. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. Multilevel superpixel structured graph U-Nets for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [\[CrossRef\]](#)
41. Mou, L.; Lu, X.; Li, X.; Zhu, X.X. Nonlocal graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8246–8257. [\[CrossRef\]](#)
42. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5966–5978. [\[CrossRef\]](#)
43. Ren, S.; Zhou, F. Semi-supervised classification for PolSAR data with multi-scale evolving weighted graph convolutional network. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *14*, 2911–2927. [\[CrossRef\]](#)
44. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. CNN-enhanced graph convolutional network with pixel-and superpixel-level feature fusion for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 8657–8671. [\[CrossRef\]](#)
45. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852.
46. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
47. Xu, B.; Shen, H.; Cao, Q.; Qiu, Y.; Cheng, X. Graph wavelet neural network. *arXiv* **2019**, arXiv:1904.07785.
48. Hechtlinger, Y.; Chakravarti, P.; Qin, J. A generalization of convolutional neural networks to graph-structured data. *arXiv* **2017**, arXiv:1704.08165.
49. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
50. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
51. Li, Z.P.; Su, H.L.; Zhu, X.B.; Wei, X.M.; Jiang, X.S.; Gribova, V.; Filaretov, V.F.; Huang, D.S. Hierarchical graph pooling with self-adaptive cluster aggregation. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *14*, 1198–1207. [\[CrossRef\]](#)
52. Lee, J.; Lee, I.; Kang, J. Self-attention graph pooling. In Proceedings of the the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 3734–3743.
53. Zhang, M.; Cui, Z.; Neumann, M.; Chen, Y. An end-to-end deep learning architecture for graph classification. In Proceedings of the the 32th AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 1–5.
54. Yuan, H.; Ji, S. Structpool: Structured graph pooling via conditional random fields. In Proceedings of the the 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–12.
55. Ranjan, E.; Sanyal, S.; Talukdar, P. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In Proceedings of the the 34th AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5470–5477.
56. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Susstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [\[CrossRef\]](#) [\[PubMed\]](#)
57. Gao, H.; Ji, S. Graph u-nets. In Proceedings of the the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 2083–2092.
58. Maas, A.L.; Hannun, A.Y.; Ng, A.Y.; et al. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the the 30th International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
59. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
60. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [\[CrossRef\]](#)
61. Wan, S.; Gong, C.; Zhong, P.; Du, B.; Zhang, L.; Yang, J. Multiscale dynamic graph convolutional network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3162–3177. [\[CrossRef\]](#)

62. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [[CrossRef](#)]
63. Qin, A.; Shang, Z.; Tian, J.; Wang, Y.; Zhang, T.; Tang, Y.Y. Spectral-spatial graph convolutional networks for semisupervised hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 241–245. [[CrossRef](#)]
64. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3D-2D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [[CrossRef](#)]
65. Wang, Z.; Du, B.; Guo, Y. Domain adaptation with neural embedding matching. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 2387–2397. [[CrossRef](#)]
66. Zhong, Y.; Hu, X.; Luo, C.; Wang, X.; Zhao, J.; Zhang, L. WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF. *Remote Sens. Environ.* **2020**, *250*, 112012. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.