




Article

DCG-Net: Enhanced Hyperspectral Image Classification with Dual-Branch Convolutional Neural Network and Graph Convolutional Neural Network Integration

Wenkai Zhu ¹ , Xueying Sun ^{1,2,*}  and Qiang Zhang ^{1,2} 

¹ College of Automation, Jiangsu University of Science and Technology, No. 666 Changhui Road, Zhenjiang 212100, China; 212210104229@stu.just.edu.cn (W.Z.); qzhang@just.edu.cn (Q.Z.)

² Systems Science Laboratory, Jiangsu University of Science and Technology, No. 666 Changhui Road, Zhenjiang 212100, China

* Correspondence: sunxueying@just.edu.cn

Abstract: In recent years, graph convolutional neural networks (GCNs) and convolutional neural networks (CNNs) have made significant strides in hyperspectral image (HSI) classification. However, existing models often encounter information redundancy and feature mismatch during feature fusion, and they struggle with small-scale refined features. To address these issues, we propose DCG-Net, an innovative classification network integrating CNN and GCN architectures. Our approach includes the development of a double-branch expanding network (E-Net) to enhance spectral features and efficiently extract high-level features. Additionally, we incorporate a GCN with an attention mechanism to facilitate the integration of multi-space scale superpixel-level and pixel-level features. To further improve feature fusion, we introduce a feature aggregation module (FAM) that adaptively learns channel features, enhancing classification robustness and accuracy. Comprehensive experiments on three widely used datasets show that DCG-Net achieves superior classification results compared to other state-of-the-art methods.

Keywords: hyperspectral image classification; graph neural network; superpixel segmentation; attention mechanism; multi-space scale



Citation: Zhu, W.; Sun, X.; Zhang, Q. DCG-Net: Enhanced Hyperspectral Image Classification with Dual-Branch Convolutional Neural Network and Graph Convolutional Neural Network Integration. *Electronics* **2024**, *13*, 3271. <https://doi.org/10.3390/electronics13163271>

Academic Editor: Silvia Liberata Ullo

Received: 11 July 2024

Revised: 11 August 2024

Accepted: 16 August 2024

Published: 18 August 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/>).

1. Introduction

Hyperspectral images (HSIs) collected from satellites or aircrafts comprise hundreds of contiguous bands and contain abundant spectral–spatial information [1]. HSIs provide detailed spectral and spatial data, essential for precise classification. Consequently, HSIs are crucial in fields such as target detection, ocean exploration, crop monitoring, and environmental investigation [2–8]. HSI classification, a key tool in HSI processing, aims to assign category labels to each pixel based on spectral and spatial information, facilitating detailed analysis of target areas for precise management and research.

Initially, HSI classification relied on machine learning methods such as random forests [9], support vector machines (SVMs) [10], principal component analysis (PCA) [11], linear discriminant analysis (LDA) [12], and sparse representation classification (SRC) [13]. These traditional supervised methods focused solely on spectral data, ignoring spatial information, which limited their classification effectiveness. Additionally, these methods lacked robustness and accuracy.

Compared to traditional machine learning, deep learning enables end-to-end, automatic, advanced feature extraction. Consequently, numerous deep learning-based models have been proposed [14–16]. Convolutional neural networks (CNNs) have significant advantages in HSI processing and have proven to be particularly effective [17,18]. Specifically, 1D-CNNs [19], 2D-CNNs [20–22], and 3D-CNNs [23,24] extract spectral, spatial, and spectral–spatial features, respectively. Zhong et al. [25] introduced residual blocks

in their SSRN model, while Wang et al. [26] (FDSSC) incorporated dense connections, enhancing network training and representation. Additionally, Yang et al. [27] developed a two-channel CNN (TCCNN) for separate spectral and spatial feature extraction. The attention mechanism [28–30] can extract more discriminative features and reduce the interference of irrelevant features in classification, making it widely used in HSI classification. For instance, Li et al. [31] (DBDA) added channel attention and spatial attention blocks to their network.

Recently, graph neural networks (GNNs) [32] have become crucial in HSI classification. GNNs handle graph structures, overcoming CNN limitations with non-Euclidean data and better extracting internal HSI relationships. For instance, Mou et al. [33] proposed a nonlocal graph convolutional network to classify HSIs using both labeled and unlabeled data. However, this method is computationally intensive because it directly uses each pixel of the entire image as a graph node. To reduce computational demand, Wan et al. [34] proposed multi-scale dynamic graph convolution (MDGCN), using superpixels instead of individual pixels for graph construction. Liu et al. [35] introduced a classification method based on multi-level superpixel structured graphs (MSSGU) to learn spectral features at different scales. Ding et al. [36] proposed a multi-feature fusion network (MFGCN) incorporating multi-scale GCNs and multi-scale CNNs to combine hyperpixel-level and pixel-level features. Additionally, Liu et al. [37] developed a CNN-enhanced GCN (CEGCN), and Dong et al. [38] proposed a convolutional neural and graph attention network weighted feature fusion model (WFCG).

Although these models have achieved good classification results, challenges remain. First, feature extraction for small-scale refined samples remains a prominent issue. Second, CNN and GCN feature extraction and fusion occur only at a single spatial scale, ignoring features at different scales. Finally, multiple features cannot be accurately fused, often resulting in information redundancy. To address these issues, we propose an innovative classification network (DCG-Net) that integrates CNN and GCN architectures, merging superpixel-level graph convolutional networks with pixel-level convolutional networks [39]. Our contributions include designing an expanding network (E-Net) for feature extraction, integrating a GCN with an attention mechanism, and developing a feature aggregation module (FAM) for adaptive channel feature learning. We demonstrate our model's effectiveness through comprehensive experiments on three widely used datasets, achieving superior classification results compared to state-of-the-art methods.

The main contributions of this study are as follows:

- Innovative classification network (DCG-Net): We introduced DCG-Net, a novel classification network that integrates convolutional neural networks (CNNs) and graph convolutional networks (GCNs). This hybrid architecture is designed to effectively capture both large-scale regular features and small-scale fine features in HSIs, addressing issues of information redundancy and feature mismatch commonly encountered in existing models.
- Expanding network (E-Net) for enhanced feature extraction: We developed a double-branch expanding network (E-Net) based on a CNN architecture. E-Net enhances the spectral features of HSIs and efficiently extracts high-level features by projecting the image information into higher spatial dimensions. This approach balances the extraction of both high-level and fine-grained features.
- Feature aggregation module (FAM): We designed a feature aggregation module (FAM) that adaptively learns channel features. The FAM dynamically calibrates channel responses within the network, enhancing the model's feature representation and extraction capabilities.

2. Related Work

2.1. HSI Classification Based on Convolutional Neural Network

Convolutional Neural Networks (CNNs) can automatically extract both spatial and spectral features from HSIs. They support an end-to-end learning framework, enabling direct learning and classification of features from the original HSIs.

Lee et al. [40] (CDCNN) achieved feature mapping of HSIs by integrating local spatial-spectral information of neighboring pixels through multi-scale dynamic convolution. To alleviate the training difficulties of deep networks, Zhong et al. [25] incorporated residual blocks into the spectral and spatial network using the 3D cube approach as training samples to reduce network parameters. Wang et al. [26] employed end-to-end fast and dense spectral-spatial convolution to facilitate the rapid classification of HSIs. Additionally, Yang et al. [27] designed a dual-channel CNN for separate spectral and spatial feature extraction. To overcome the limitations of fixed kernel shapes in conventional convolution, Liu et al. [41] used dynamic convolution for classification, allowing the kernel shape to adapt according to the spatial distribution of the HSI. This deformable convolution helps suppress irregularities and accidental features at category boundaries, enhancing the ability to learn features in cross-category regions. Furthermore, the attention mechanism has been applied to CNN networks. For instance, Li et al. [31] designed two branches to capture a large number of spectral and spatial features in the HSI, applying channel attention blocks and spatial attention blocks to improve feature extraction accuracy. Given that CNN networks often require a large number of computational parameters and are computationally expensive during the training phase, Wang et al. [42] (LMAFN) designed a lightweight spectral-spatial attention feature fusion network based on the network search framework (NAS), which reduces computational parameters by adjusting the weights of different channels through adaptive passbands.

Although various structural CNN networks have achieved good results in addressing HSI classification problems, significant challenges remain in handling the internal connections within HSIs. This difficulty arises from the complex nature of hyperspectral data, which often involves non-Euclidean relationships.

2.2. HSI Classification Based on Hyperpixel-Based Graph Convolutional Network

Using a superpixel segmentation [43] strategy to create the graphical structure accurately captures the core information of the image and significantly reduces the parameter requirements of the graph convolutional network when processing HSIs. This approach reduces model complexity while ensuring a sensitive and accurate interpretation of image details.

Recently, various superpixel-based graph convolutional models for HSI classification have been proposed. For example, Wan et al. [34] (MDGCN) used a multi-scale simple linear iterative clustering (SLIC) approach to obtain superpixels, capturing multi-scale feature information of the HSI as input nodes for the GCN. To address the problem of superpixel segmentation accuracy, GAT-AGSM [44] designed the SFS mechanism, which is able to assign different channel weights to the superpixel features. This can make the superpixel segmentation generate finer classification results. To fuse pixel-level and superpixel features, Liu et al. [37] first used SLIC to segment and encode the HSI into a graph structure, then they constructed an association matrix between pixels and superpixels to capture superpixel-level features with the GCN and combined them with the pixel-level features extracted by the CNN. To learn spectral features at different scales, Liu et al. [35] (MSSGU) constructed a multilevel graph structure by obtaining superpixels using a region merging-based superpixel segmentation method, then they built graph structures at different scales. This method effectively fuses multi-scale feature information, thereby improving the classification accuracy of HSIs. In superpixel segmentation, pixels with similar attributes in an image are organized into separate units, enhancing model processing speed, the accuracy of feature extraction, and overall model expressiveness. This demonstrates the significant value of superpixels in image processing and analysis.

Superpixel segmentation organizes pixels with similar attributes into distinct units. This approach improves processing speed, enhances feature extraction accuracy, and increases the model's expressiveness. This demonstrates the significant value of superpixels in image processing and analysis.

Although these methods have achieved good classification results, they still face challenges in processing fine edge features. To address these issues, we constructed superpixel graph structures at various spatial scales using superpixel-based graph convolution. Simultaneously, we fused GCNs and CNNs at multiple spatial scales to effectively enhance texture feature extraction.

3. Proposed Method

In this paper, we propose DCG-Net, a novel HSI classification method. As shown in Figure 1, HSIs exhibit complex edge features and unbalanced sample distributions. To enhance the extraction of detailed and small-scale sample features, we propose an expanding network (E-Net) based on a double-branch encoder–decoder architecture. The E-Net effectively extracts regular spatial–spectral features and amplifies spatial features to refine details and edge features through high-dimensional mapping. To enhance the extraction of large-scale sample features, we incorporate a superpixel-based GCN. This integration enables the fusion of pixel-level local spatial features with superpixel-level large-scale spatial features. The graph convolution module is embedded in the encoder and decoder of E-Net, facilitating a more effective fusion of local and global features. Additionally, to enhance dual-channel feature aggregation, we designed a feature aggregation module based on the attention mechanism. This module further optimizes feature extraction, enhancing the accuracy and robustness of the classification process.

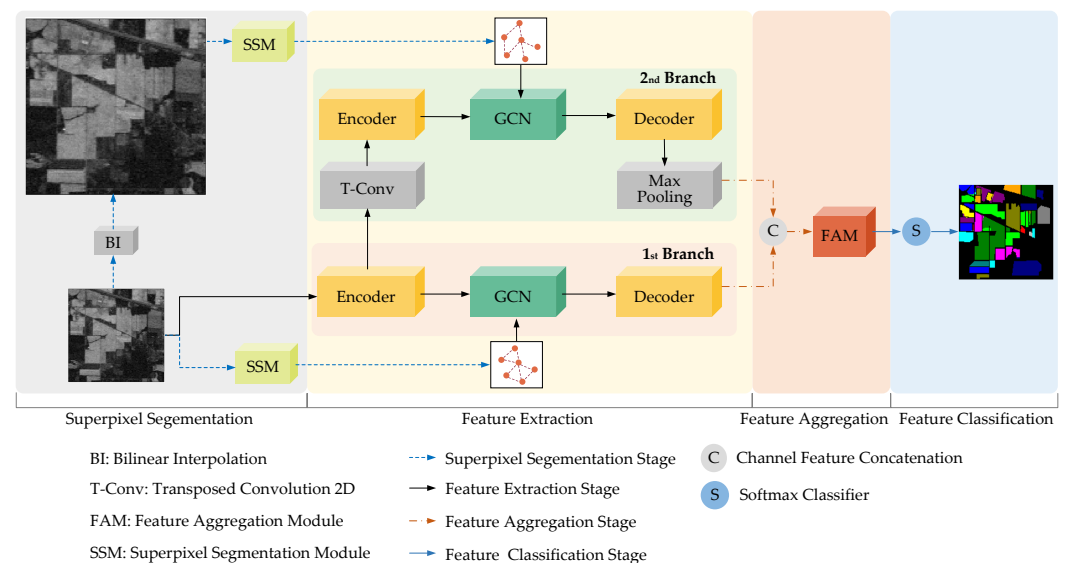


Figure 1. DCG-Net architecture diagram. Our model is structured into four primary stages: superpixel segmentation, feature extraction, feature aggregation, and feature classification. We employ a dual-branch architecture, where each branch processes both pixel-level and superpixel-level features. The 1st branch retains the original image resolution for feature processing, whereas the 2nd branch processes upscaled feature images to facilitate multi-scale feature analysis.

We first use the superpixel segmentation module to construct the spatial topology of HSIs and preserve their large-scale structural information. Then, we incorporate the attention mechanism into GCN to enhance the correlation between graph nodes. Next, to fully extract small-scale sample features, we integrate E-Net and GCN. This enables the extraction of both pixel-level and superpixel spectral features, along with multi-spatial-scale feature extraction. Then, we use the dynamically integrated feature aggregation module at different scales. Finally, we predict the obtained features using a Softmax classifier.

3.1. Expanding Network

HSIs exhibit complex edge features and unbalanced sample distributions. To enhance the extraction of advanced features and small-scale samples, we designed a double-branch encoding and decoding expanding network (E-Net), as shown in Figure 2. E-Net is capable of projecting HSI to spatially higher dimensions and effectively combining GCN, enhancing the extraction of pixel-level and superpixel features for multi-spatial-scale feature extraction. We will now introduce E-Net in detail.

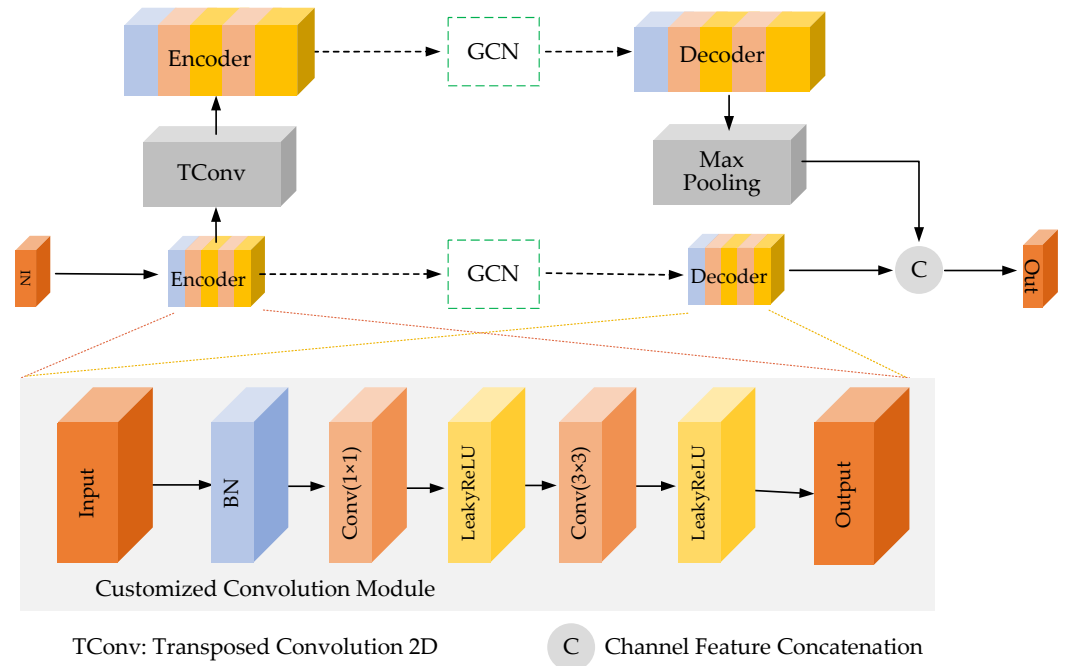


Figure 2. E-Net structure diagram. Both the encoder and decoder use a customized convolution module. E-Net can effectively combine GCN in the dual-branch encoding and decoding processes to achieve the effective combination of different spatial features.

In E-Net, the first branch network consists of an encoder–decoder structure operating at the original image size. The encoder of the second branch network upscales the feature output from the first branch encoding to a higher spatial scale and performs feature extraction. The decoder of the second branch network downsamples these features and performs convolution operations, enabling efficient extraction of small-scale sample features at this level.

Specifically, the encoder and decoder use a customized convolution module built upon the CNN architecture. In this module, input features undergo batch normalization first, followed by downsampling with a (1×1) pointwise convolution. Subsequently, nonlinear transformation is performed using the Leaky ReLU activation function [45]. Then, the features are extracted by a (3×3) depth-separable convolution, followed by another nonlinear transformation using the Leaky ReLUs [45] activation function. After the first branch of the encoder, the input features are up-sampled, and the resulting output features are then up-sampled as follows:

$$Y_u = TConv(Y), \quad (1)$$

where Y represents the feature output from the first branch encoder, $TConv$ denotes the transpose convolution that doubles the spatial size of the input features, and Y_u is the resulting feature with doubled spatial dimensions. The feature output Y from the first branch encoder is also fed into the first branch GCN network. The specific operation of the GCN network will be described below. After extracting the superpixel-level features using the GCN network, we obtain the new feature Y_g . At this point, Y_g is input into the

customized convolution module for decoding, and the superpixel-level features are fused with the pixel-level features to produce the feature Y_c .

Simultaneously, the upsampled feature Y_u passes through the second branch encoder for additional pixel-level feature extraction. The extracted feature is also input into the GCN network to obtain the upsampled superpixel-level feature output Y_{ug} , which is subsequently decoded to yield Y_{uc} . This decoded feature is then downsampled, as follows:

$$Y_p = \text{MaxPooling}(Y_{uc}), \quad (2)$$

where *MaxPooling* refers to the MaxPool2d function and Y_p represents the feature output after downsampling. At this point, the features Y_p and Y_c have the same spatial size. They are then concatenated to generate the output features of E-Net.

3.2. Superpixel Structured Graph

Graph convolutional networks accept only graph-structured data as input, necessitating the construction of a graph structure. We use a superpixel segmentation method to construct the graph structure. This method groups pixels with similar features into units, simplifying the graph representation while preserving crucial image information.

As shown in Figure 3, to reduce information redundancy and noise pollution of the HSI, we first apply a principal component analysis (PCA) to reduce the dimensionality of the original HSI. Subsequently, we apply superpixel segmentation to obtain a specified number of superpixels [43]. Each superpixel is treated as a node, and we construct the graph structure using the K-nearest neighbor (KNN) method [46]. Specifically, assuming the number of superpixels is Z , we define $G = (\mathcal{V}, \mathcal{E})$ as an unweighted graph, where \mathcal{V} and \mathcal{E} represent the nodes and edges of the graph, respectively. Here, the edges \mathcal{E} are represented by the adjacency matrix $A \in \mathbb{R}^{Z \times Z}$, where $A_{i,j}$ denotes the edge of the node in the j -th column of the i -th row. The adjacency matrix A is constructed using the K-nearest neighbor approach and is computed as follows:

$$A_{i,j} = \begin{cases} 1, & \text{if } p_i \text{ is one of the K-Nearest-Neighbors of } p_j \\ 0, & \text{otherwise} \end{cases}, \quad (3)$$

where p_i and p_j denote the i -th and j -th superpixel in the superpixel, respectively.

Meanwhile, to establish the relationship between pixel features and superpixel features, influenced by MSSGU [35], we construct the association matrix $M \in \mathbb{R}^{(HW) \times Z}$. As shown in Figure 3, we define $G^k = (\mathcal{V}^k, \mathcal{E}^k)$ as the original HSI 4-connected graph. We take the first three principal components extracted by the PCA as the node features in G^k and define x_i as the i -th pixel of the original HSI after dimensionality reduction via PCA. The final correlation matrix $M_{i,j}$ is computed using the following formula:

$$M_{i,j} = \begin{cases} 1, & \text{if } x_i \subseteq p_j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Finally, node \mathcal{V} can be represented by the node matrix H . The computation can be expressed as follows:

$$H = (\hat{M}_{i,j})^T \mathcal{V}^k, \quad (5)$$

where $\hat{M}_{i,j}$ denotes the association matrix normalized by columns, $\hat{M}_{i,j} = M_{i,j} / \sum_m M_{m,j}$, and \mathcal{V}^k is the node matrix of the 4-connectivity graph of the original HSI. Specifically, we construct the 4-connectivity map of the original HSI to obtain its node matrix \mathcal{V}^k , which is directly constructed from the original pixels. At this stage, \mathcal{V}^k represents the pixel-level node information. Consequently, the constructed H-matrix allows the node features to be transferred from the pixel level to the superpixel level. At this point, in order to transfer

from superpixel-level features to pixel-level features, we construct the adjacency matrix \hat{H} , which can be calculated as follows:

$$\hat{H} = \hat{M}_{i,j} \mathcal{V}^k \tag{6}$$

For the 2nd step of DCG-Net, we perform bilinear interpolation on the original HSI to construct a new HSI graph. We then repeat the process to construct a graph structure that aligns with the spatial scale.

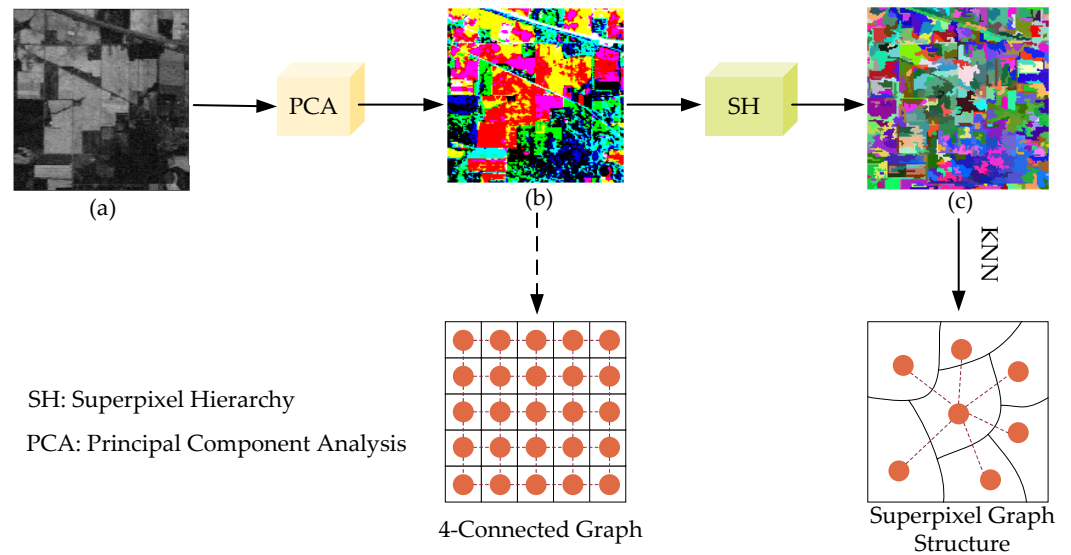


Figure 3. Flowchart of superpixel segmentation and graph structure construction. (a) Original hyperspectral image. (b) Hyperspectral image after PCA downscaling, with the 4-connected graph as a feature graph constructed using the pixels of (b) as nodes. (c) Hyperspectral image after superpixel segmentation, with the superpixel graph structure as a feature graph constructed using the superpixels of (c) as nodes. The orange dots represent the graph’s nodes, and the orange dotted lines represent the graph’s edges.

3.3. Graph Convolutional Network

We used superpixel segmentation to divide the image into multiple superpixels, which forms the basis for constructing the graph structure. We then employed a graph convolutional network (GCN) [47] to focus on extracting features at the superpixel level, improving the model’s feature extraction capabilities. Additionally, we incorporated an attention mechanism into the GCN to enhance the correlation between graph nodes.

We will outline the specific implementation method, beginning with the calculation of the attention coefficient between a node and its neighbors:

$$A_{i,j} = \text{Sigmoid}(HM \cdot (HM)^T), \tag{7}$$

$A_{i,j}$ denotes the attention coefficient between the i -th node and the j -th neighbor, where M is the learnable parameter and H is the adjacency matrix. Next, the attention coefficient matrix is obtained using normalization $\hat{A}_{i,j}$:

$$\hat{A}_{i,j} = \text{Softmax}(A_{i,j}), \tag{8}$$

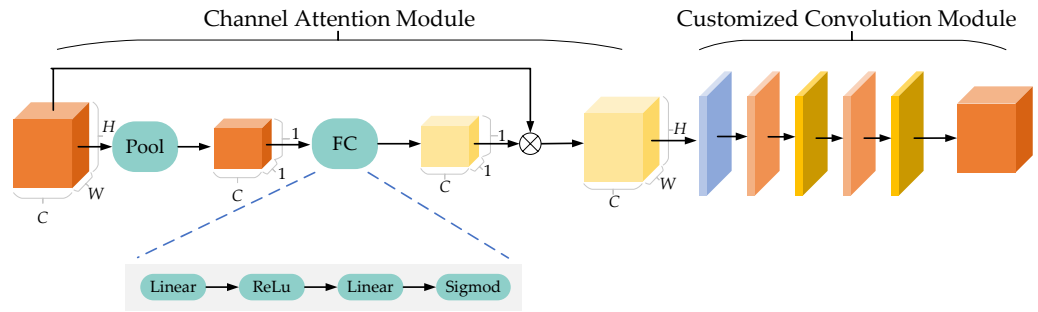
Finally, the attention coefficient matrix $\hat{A}_{i,j}$ is weighted and averaged over the node features, and Leaky ReLU [45] is used as the activation function to perform the graph convolution:

$$fg(H, A) = \text{LeakyReLU}\left(\sum_{j=1}^W \hat{A}_{i,j} \cdot out_{i,j}\right), \tag{9}$$

where H represents the node matrix, A is the adjacency matrix, W is the number of neighbors, and $out_{i,j}$ denotes the feature representation between the j -th neighbor and the i -th node.

3.4. Feature Aggregation Module

After obtaining the dual-branch features, we need to perform feature fusion to dynamically calibrate the channel response within the network and enhance the model’s feature perception. As shown in Figure 4, we designed a feature aggregation module (FAM) that includes a channel attention module and a customized convolution module. Next, these two modules will be introduced in detail.



Pool: AdaptiveAvgPool2d ⊗ : Attention operates on each channel

Figure 4. Feature aggregation module structure diagram. The module begins with channel feature learning through the channel attention module, then it continues with feature extraction using the customized convolution module.

Channel attention module: We use the channel attention module to dynamically reweight the fused features. As shown in Figure 4, we apply global adaptive pooling to the input features, compressing their spatial scale from $BC \times H \times W$ to $B \times C \times 1 \times 1$ to extract global information, expressed as follows:

$$Y = AvgPool(X), \tag{10}$$

where X is the input feature and Y is the feature output. Then, the features pass through two MLP layers. The first MLP layer conducts a fully connected operation followed by ReLU activation, and the second MLP layer also performs a fully connected operation followed by sigmoid activation to constrain the attention weights within the range of $(0, 1)$. This process is specifically illustrated as follows:

$$\begin{aligned} Y_{s1} &= \text{Linear}\left(Y, \frac{ch_{in}}{reduction}\right) \\ Y_{s2} &= \text{ReLU}(Y_{s1}) \\ Y_{s3} &= \text{Linear}(Y_{s2}, ch_{in}) \\ Y_{out} &= \text{Sigmoid}(Y_{s3}) \end{aligned}, \tag{11}$$

where Y_{s1}, Y_{s2}, Y_{s3} are the intermediate features, Y_{out} is the normalized attention feature, ch_{in} is the number of input feature channels, and reduction is the feature reduction factor. Finally, the attention features are applied to each channel.

Customized convolution module: After extracting features using the channel attention module, we input these features into the customized convolution module for further extraction of pixel-level features. In this module, features are first batch normalized, then downsampled using (1×1) point-wise convolution and subjected to a nonlinear transformation using the Leaky ReLU activation function [45]. Next, features are processed

with (3×3) depthwise separable convolution for extraction, followed by another nonlinear transformation using Leaky ReLU [45], resulting in the final output features.

Through these processes, we globally and adaptively learn the weights of each channel, assigning varying degrees of importance. We then use the customized convolution module to enhance the model's representation capability and feature extraction, thus enhancing the robustness and accuracy of classification.

4. Experimental Results and Analysis


4.1. Experiment Design

To comprehensively evaluate DCG-Net, we used three well-known datasets: Indian Pines, Salinas, and Kennedy Space Center. We compared DCG-Net with six state-of-the-art methods: contextual CNN-based deep networks (CDCNNs) [40], dual-branch dual attention networks (DBDAs) [31], dual-branch multi-attention networks (DBMAs) [48], fast dense spectral-spatial convolutional networks (FDSSCs) [26], spectral-spatial residual networks (SSRNs) [25], and CNN-enhanced graph convolutional networks (CEGCNs) [37]. The overall accuracy (OA), average accuracy (AA), and Kappa statistics were used as the performance evaluation criteria.

4.1.1. HSI Datasets

To validate the robustness and effectiveness of DCG-Net, we used three datasets: Indian Pines (IN) [49], Salinas (SA) [49], and Kennedy Space Center (KSC) [49], as shown in Figures 5–7. In Figures 5–7, 'Train' indicates the number of training samples, 'Val' represents the number of validation samples, and 'Test' represents the number of test samples.

Indian Pines dataset: This dataset was captured by the US AVIRIS sensor in the Indian remote sensing experiment (IRSE) area, with an image size of 145×145 and a spatial resolution of 20 m. The 20 bands covering the water vapor absorption region are removed, and the remaining 200 spectral bands are taken. In addition, there are a total of 16 unevenly distributed classes containing 10,366 labeled samples. Figure 5 shows a pseudo-color visualization of this dataset, where each color denotes a distinct label category.



















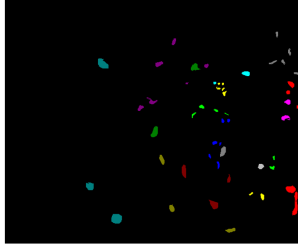
Class	Legend	Name	Train	Val	Test
1		Alfalfa	3	3	48
2		Corn_notill	72	72	1290
3		Corn_mintill	42	42	750
4		Corn	12	12	210
5		Pasture	25	25	447
6		Trees/Grass	38	38	671
7		Pasture_mowed	2	2	22
8		Hay_windrowed	25	25	439
9		Oats	1	1	18
10		Soybeans_notill	49	49	870
11		Soybeans_mintill	124	124	2220
12		Soybeans_cleantill	31	31	552
13		Wheat	11	11	190
14		Woods	65	65	1164
15		Building_grass	19	19	342
16		Stone_steel_towers	5	5	85
Total			524	524	9318

Figure 5. Description of Indian Pines dataset.

Kennedy Space Center dataset: This dataset comprises images captured by the U.S. at the Kennedy Space Center, Florida using the AVIRIS sensor with a size of 512×614 pixels and a resolution of 18 m. The water vapor absorption and noise bands were removed, and the remaining 176 spectral bands were taken. This HSI has 13 categories, with a total of

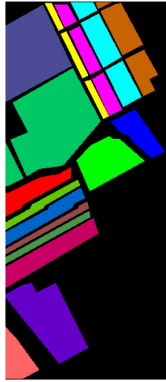
5211 labeled samples. Figure 6 shows a pseudo-color visualization of this dataset, where each color denotes a distinct label category.



Class	Legend	Name	Train	Val	Test
1		Srub	23	23	715
2		Willow_swamp	8	8	227
3		CP_hammock	8	8	240
4		Slash_pine	8	8	236
5		Oak/Broadleaf	5	5	151
6		Hardwood	7	7	215
7		Swamp	4	4	97
8		Graminoid	13	13	405
9		Spartina_marsh	16	16	488
10		Cattail_marsh	13	13	378
11		Salt_marsh	13	13	393
12		Mud_flats	16	16	471
13		Water	28	28	871
Total			162	162	4887

Figure 6. Description of Kennedy Space Center dataset.

Salinas dataset: This dataset was captured by the United States in Salinas Valley in California using the AVIRIS sensor. Its image size is 512 × 217, and it has a spatial resolution of 3.7 m. Twenty water vapor absorption bands were removed, and the remaining 204 spectral bands were taken. This HSI has 16 categories, totaling 54,129 labeled samples. Figure 7 shows a pseudo-color visualization of this dataset, where each color denotes a distinct label category.



Class	Legend	Name	Train	Val	Test
1		Brocoli_green_weeds_1	11	11	1987
2		Brocoli_green_weeds_2	19	19	3688
3		Fallow	10	10	1956
4		Fallow_rough_plow	7	7	1380
5		Fallow_smooth	14	14	2650
6		Stubble	20	20	3919
7		Celery	18	18	3543
8		Grapes_untrained	57	57	11,157
9		Soil_vinyard_develop	32	32	6139
10		Corn_senesced_green_weeds	17	17	3244
11		Lettuce_romaine_4wk	6	6	1056
12		Lettuce_romaine_5wk	10	10	1907
13		Lettuce_romaine_6wk	5	5	906
14		Lettuce_romaine_7wk	6	6	1058
15		Vinyard_untrained	37	37	7194
16		Vinyard_vertical_trellies	10	10	1787
Total			279	279	53,571

Figure 7. Description of the Salinas dataset.

4.1.2. Evaluation Indices

Four performance metrics were used in this experiment to evaluate the classification ability of the model: per-class accuracy (PA), overall accuracy (OA), average accuracy (AA), and statistics (Kappa). The formulas for each metric are as follows:

$$PA = \frac{T_i}{F_i} \tag{12}$$

$$OA = \frac{\sum_i T_i}{\sum_i F_i}, \quad (13)$$

$$AA = \frac{1}{I} \sum_i PA, \quad (14)$$

$$Kappa = \frac{OA - \frac{\sum_i T_i F_i}{(\sum_i F_i)^2}}{1 - \frac{\sum_i T_i F_i}{(\sum_i F_i)^2}}, \quad (15)$$

In Equations (12)–(15), I indicates that the dataset has I categories of samples, i represents the i category of samples, T_i indicates the number of samples in the i -th category, and F_i indicates the number of samples correctly categorized in the i category.

4.1.3. Environment Configuration

All experiments were conducted in PyCharm using Python 3.11, PyTorch 2.0.1, and an NVIDIA GeForce RTX 3090 Ti GPU server manufactured by Kuankes Co., Shanghai, China. The loss function used in this experiment is category-weighted cross entropy. All activation functions are Leaky ReLU. The network parameters were updated using the Adam optimizer with a learning rate of 0.0005, and the total number of iterations was set to 600. The number of superpixels was set to 512, and the number of neighbors for KNN was set to 15.

4.2. Experiment Results

4.2.1. Comparative Analysis of Classification Performance

In this section, we compare the proposed DCG-Net with six recent methods. For a fair comparison, we used the hyperparameter settings reported in the respective references. Each experiment was repeated ten times to calculate the mean and standard deviation for each metric. The statistical classification results of all the methods are shown in Tables 1–3. Additionally, to further analyze the classification performance of our proposed network, the classification results for all the methods are visualized in Figures 8–10.

(a) Results of comparative experiments on the IN dataset. For the IN dataset, as shown in Figure 5, the training, validation, and test sets comprised 5%, 5%, and 90%, respectively. The proposed DCG-Net achieved an overall accuracy (OA) of 98.37%, an average accuracy (AA) of 97.64%, and a Kappa coefficient of 98.15; the highest among all the compared models, as shown in Table 1. Compared to CDCNN, DBMA, FDSSC, SSRN, DBDA, and CEGCN, DCG-Net improved the OA by 22.59%, 4.45%, 2.11%, 3.65%, 1.57%, and 1.16%, respectively. The IN dataset has an uneven sample distribution across its classes, especially with very few training samples in classes 1, 7, 9, and 16. In CDCNN, DBMA, SSRN, DBDA, and FDSSC, these classes exhibit poor classification performance due to limited sample sizes. For instance, despite CEGCN's combined double-branch structure of convolutional and graph networks, it still lacks the ability to effectively extract features from small-scale samples. Conversely, DCG-Net achieved classification accuracies of over 93% in these four challenging classes, demonstrating its strong ability to extract features from small-scale samples.

Figure 8 shows the predicted labeled images for the seven methods for the IN dataset. CDCNN displays significant noise in its classification results, while DBMA, SSRN, DBDA, FDSSC, and CEGCN exhibit varying levels of noise and misclassification. In contrast, the proposed DCG-Net generates smoother and more accurate classification results with fewer noise artifacts and errors. This visualization result clearly demonstrates the superiority of our proposed method.

(b) Results of comparative experiments on the KSC dataset. The KSC dataset, known for its high spatial resolution and low noise levels, is particularly suitable for classification tasks. As shown in Figure 6, due to the dataset's limited sample size, only 5% of the samples were used as training data, resulting in fewer than 30 training samples per category. As shown in Table 2, DCG-Net achieved exceptional performance, with an overall accuracy

(OA) of 99.49%, an average accuracy (AA) of 99.43%, and a Kappa coefficient of 99.44; the highest among all the compared models. Specifically, DCG-Net improved the OA by 11.30%, 5.07%, 1.82%, 4.19%, 2.67%, and 0.31% compared to CDCNN, DBMA, FDSSC, SSRN, DBDA, and CEGCN, respectively. Our model achieved 100% classification accuracy in six categories: 2, 4, 6, 7, 8, and 13, highlighting its robust classification abilities.

Table 1. Classification results of different methods in terms per-class accuracy, OA (%), AA (%), and Kappa for the Indian Pines dataset.

Class	CDCNN	DBMA	FDSSC	SSRN	DBDA	CEGCN	DCG-Net
1	31.86	90.69	76.02	93.68	90.08	69.79	98.95
2	68.65	90.68	76.02	93.36	96.14	96.71	98.03
3	62.57	95.53	93.44	92.64	97.45	98.05	98.42
4	58.52	94.16	96.18	88.95	97.11	91.61	98.09
5	96.37	97.02	98.77	99.42	97.93	93.02	94.94
6	90.16	98.30	99.08	98.31	78.73	99.40	99.35
7	39.54	70.19	76.15	88.56	78.39	90.90	94.09
8	88.17	98.14	96.86	98.04	99.86	99.97	99.97
9	74.32	75.43	77.12	84.63	79.96	23.33	93.12
10	66.52	89.32	93.97	91.81	94.27	94.71	96.88
11	75.81	95.02	97.30	94.28	97.65	98.81	98.61
12	49.91	92.64	95.78	94.21	95.78	96.99	97.60
13	93.39	98.39	99.84	99.33	96.04	99.36	99.26
14	88.69	97.29	96.80	98.25	98.80	99.67	99.99
15	78.82	87.86	96.80	93.61	97.03	94.18	98.73
16	95.32	94.19	97.42	91.83	86.62	93.52	96.23
OA (%)	75.78 ± 5.02	93.92 ± 1.54	96.26 ± 2.49	94.72 ± 0.72	96.80 ± 0.59	97.21 ± 0.51	98.37 ± 0.58
AA (%)	72.41 ± 7.97	91.55 ± 1.56	92.93 ± 6.73	93.81 ± 1.74	93.78 ± 2.25	90.00 ± 2.33	97.64 ± 0.86
Kappa	72.22 ± 5.97	93.07 ± 1.76	95.73 ± 2.86	93.98 ± 0.82	96.35 ± 0.68	96.82 ± 0.58	98.15 ± 0.66

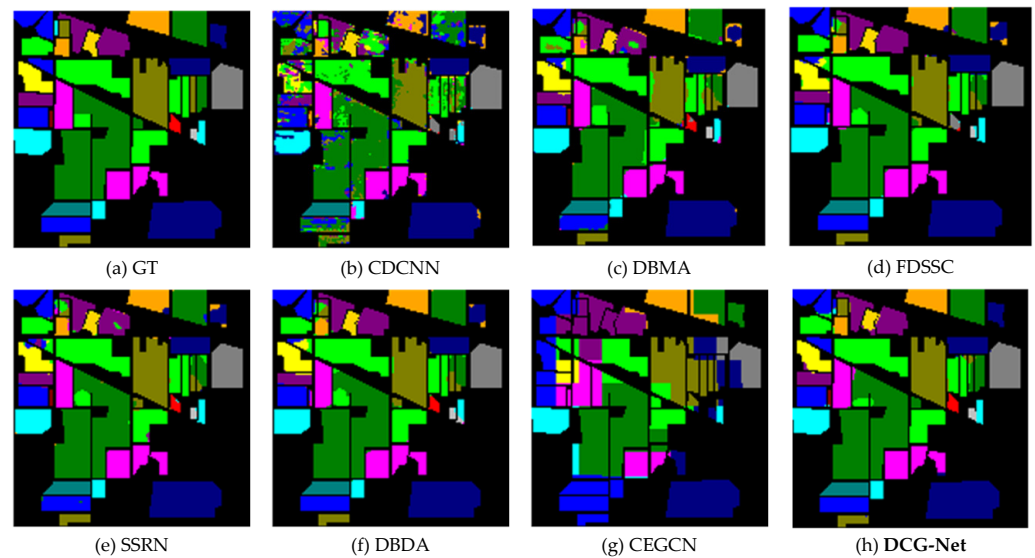
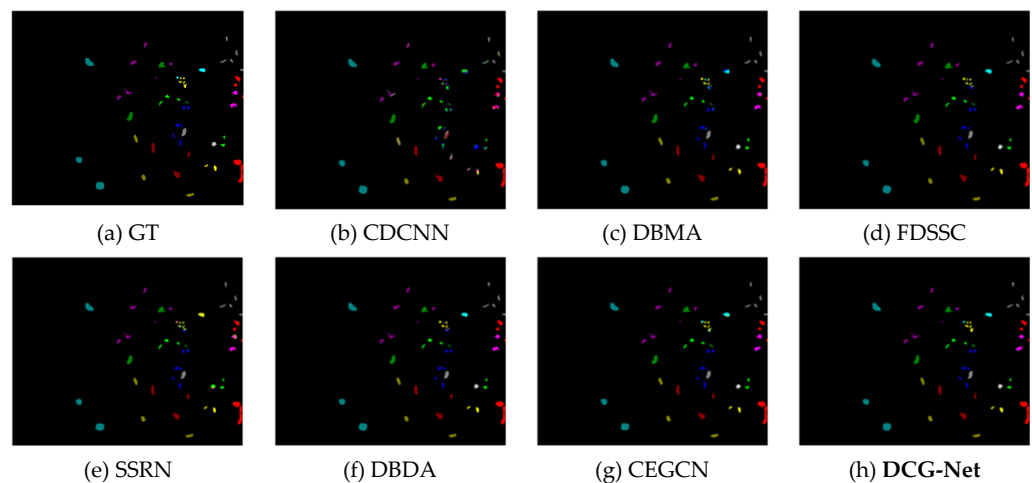


Figure 8. Classification maps of different methods for the Indian Pines dataset.

Figure 9 shows the predicted labeled images of these seven methods on the KSC dataset. CDCNN’s predicted images exhibit more errors, while DBMA, SSRN, DBDA, FDSSC, and CEGCN show fewer errors but still some noise. In contrast, DCG-Net demonstrates a superior classification performance with minimal noise, closely aligning with the true labels.

Table 2. Classification results of different methods in terms of per-class accuracy, OA (%), AA (%), and Kappa for the Kennedy Space Center dataset.

Class	CDCNN	DBMA	FDSSC	SSRN	DBDA	CEGCN	DCG-Net
1	94.71	100.00	98.64	96.54	99.94	99.56	99.61
2	85.41	93.38	97.38	90.87	96.46	98.41	100.00
3	79.28	79.58	90.82	87.57	83.27	99.58	99.43
4	53.57	75.11	90.56	75.62	83.32	95.08	100.00
5	47.40	65.96	89.85	69.62	90.85	90.53	97.76
6	71.22	91.48	99.14	93.07	98.44	99.48	100.00
7	72.16	87.74	93.69	71.67	88.58	98.76	100.00
8	84.10	95.38	98.71	98.13	99.64	99.77	100.00
9	91.09	96.07	99.76	98.49	99.91	100.00	99.89
10	92.85	97.02	99.86	99.24	100.00	100.00	98.67
11	98.24	99.89	98.77	98.96	99.26	100.00	98.43
12	96.25	98.17	98.12	99.52	99.43	99.29	98.86
13	99.91	100.00	100.00	99.89	99.91	100.00	100.00
OA (%)	88.14 ± 2.57	94.42 ± 1.91	97.67 ± 1.12	95.30 ± 2.65	96.82 ± 1.31	99.18 ± 0.51	99.49 ± 0.34
AA (%)	82.02 ± 3.61	90.75 ± 2.93	96.65 ± 1.13	90.71 ± 8.00	95.31 ± 1.51	98.50 ± 0.89	99.43 ± 0.55
Kappa	86.78 ± 2.88	93.78 ± 2.13	97.41 ± 1.25	94.76 ± 2.96	96.46 ± 1.45	99.08 ± 0.57	99.44 ± 0.38

**Figure 9.** Classification maps of different methods for the Kennedy Space Center dataset.

(c) Results of comparative experiments on the SA dataset. The SA dataset, characterized by simpler scenarios, explicit feature information, and a more continuous distribution, is relatively easier to train for classification purposes. Consequently, as shown in Figure 7, the dataset was divided into a training set of 0.5%, a validation set of 0.5%, and a test set of 99%. As indicated in Table 3, DCG-Net achieved 99.53% OA, 99.22% AA, and a Kappa coefficient of 99.49; the highest among all the methods. Compared to CDCNN, DBMA, FDSSC, SSRN, DBDA, and CEGCN, DCG-Net improved OA by 10.69%, 3.01%, 3.51%, 6.36%, 2.78%, and 0.79%, respectively. In individual categories 1, 2, 3, and 9, DCG-Net achieved a 100% accuracy, while CEGCN achieved a 100% accuracy only in categories 2 and 9. DBMA, FDSSC, and SSRN achieved a 100% accuracy only in category 1, while CDCNN and DBDA did not achieve a 100% accuracy in any category. This highlights DCG-Net's superior classification performance across all the categories.

Figure 10 shows the predicted labeled images of these seven methods on the SA dataset. Compared with the other six algorithms, DCG-Net demonstrates superior classification results with minimal noise, closely aligning with the ground truth and showcasing excellent visual classification performance.

Table 3. Classification results of different methods in terms of per-class accuracy, OA (%), AA (%), and Kappa for the Salinas dataset.

Class	CDCNN	DBMA	FDSSC	SSRN	DBDA	CEGCN	DCG-Net
1	97.34	100.00	100.00	100.00	100.00	99.95	100.00
2	97.25	99.92	99.92	99.75	99.96	100.00	100.00
3	90.37	97.79	97.75	94.21	98.16	99.86	100.00
4	98.08	93.55	97.32	97.87	94.61	99.52	95.99
5	94.60	98.54	99.38	98.89	98.98	98.10	98.28
6	96.79	99.56	99.95	99.85	99.83	99.83	99.82
7	97.09	99.41	99.51	99.32	98.90	99.98	99.77
8	78.56	94.18	90.51	88.45	93.04	97.64	99.70
9	99.00	99.60	99.50	99.36	99.35	100.00	100.00
10	88.21	97.13	97.19	96.93	98.29	96.97	98.56
11	84.98	95.35	94.84	95.68	96.10	99.75	97.48
12	95.94	99.18	98.21	98.69	99.16	100.00	99.84
13	96.92	99.29	99.44	97.91	99.64	99.80	99.86
14	93.76	93.49	96.99	96.74	94.96	98.63	97.91
15	73.75	90.90	90.75	81.16	93.17	97.11	99.92
16	94.27	99.04	99.93	99.21	99.91	98.97	99.93
OA (%)	88.80 ± 1.03	96.52 ± 0.89	96.02 ± 1.15	93.17 ± 3.50	96.75 ± 0.64	98.74 ± 0.50	99.53 ± 0.29
AA (%)	92.31 ± 1.26	97.31 ± 0.66	97.58 ± 0.52	96.50 ± 0.74	97.75 ± 0.41	99.13 ± 0.18	99.22 ± 0.48
Kappa	87.52 ± 1.14	96.12 ± 0.99	95.56 ± 1.29	92.42 ± 3.82	96.38 ± 0.72	99.13 ± 0.55	99.49 ± 0.29

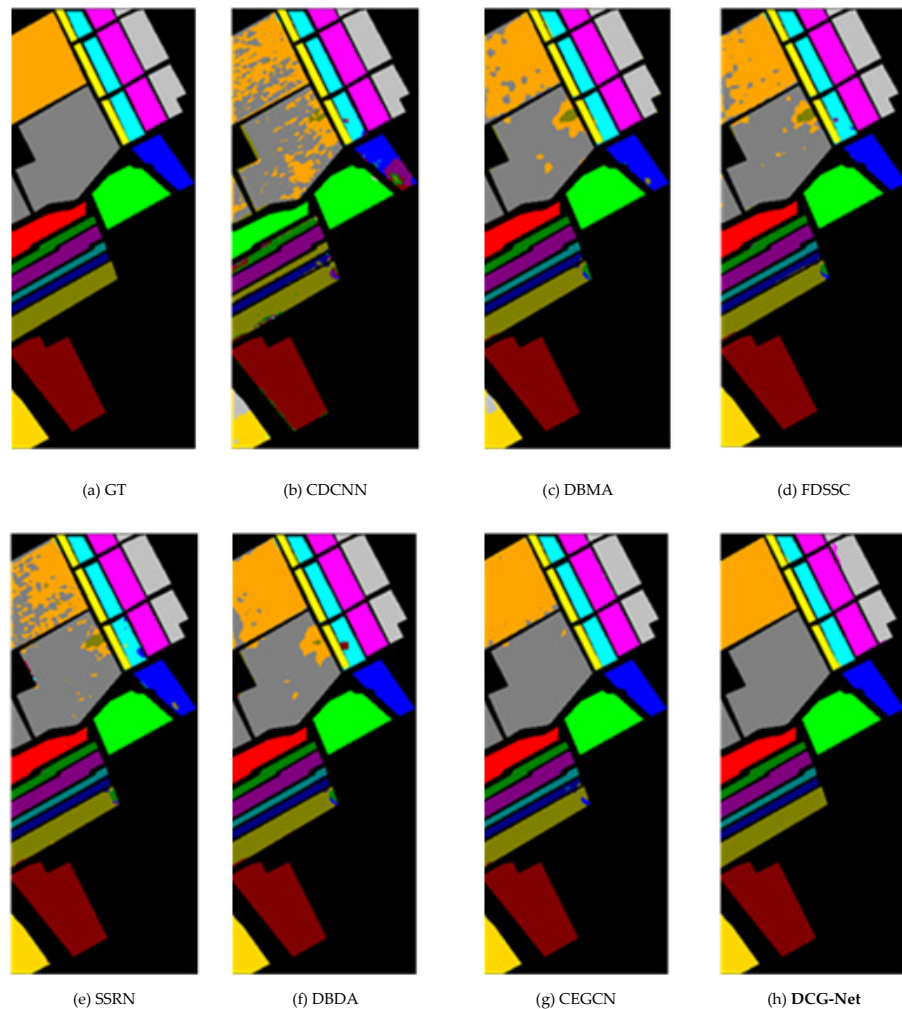


Figure 10. Classification maps of different methods for the Salinas dataset.

DCG-Net achieved the highest OA, AA, and Kappa scores across all three datasets, demonstrating superior classification effectiveness. DCG-Net’s double-branch structure enhances the extraction of features from small sample ranges, while the graph network excels at capturing global features from larger ranges. This dual-branch network structure enhances the complementarity between features at different scales, enabling the model to effectively adapt to diverse target characteristics and achieve superior classification performance. In regions with large-scale samples, DCG-Net produces smoother classification maps, effectively suppressing noise. Conversely, in regions with small-scale samples, DCG-Net delivers finer-grained classification results with a higher accuracy, highlighting its robustness.

4.2.2. Comparison of Running Time between Different Methods

To compare the direct running efficiency for each model, this section records the running time of the different methods. The experimental setup is the same as in Section 4.2.1. Table 4 shows the time cost of each model for the three datasets.

Table 4 shows that the running efficiency of DBMA, FDSSC, SSRN, and DBDA across the three datasets is generally low. This is because these models are based on 3D cubes, which contain a large number of parameters and require longer running and testing times. CDCNN is a 2D-CNN-based model with a comparatively shorter training time. However, all five models utilize local blocks of HSIs as inputs, resulting in long computation times. CEGCN and DCG-Net employ the whole HSIs as input, facilitating parallel computation, and the GCN-based approach enhances the processing speed. Given that the 2nd branch of DCG-Net processes super-resolution hyperspectral data, its structure is more complex, leading to longer processing times than CEGCN. Nonetheless, among all the compared methods, DCG-Net achieved the best classification results, underscoring the superiority of our model.

Table 4. Comparison of running times between different methods.

Dateset	Time (s)	CDCNN	DBMA	FDSSC	SSRN	DBDA	CEGCN	DCG-Net
Indian Pines	Train	47.10	451.78	584.42	407.85	436.5	4.74	17.52
	Test	0.49	3.74	2.18	1.68	3.41	1.52	0.60
Salinas	Train	25.40	237.50	312.99	213.95	229.71	17.92	79.95
	Test	2.82	21.87	12.67	9.69	19.69	1.74	2.92
Kennedy Space Center	Train	23.30	175.87	230.31	164.04	180.00	60.46	278.95
	Test	0.24	1.67	0.99	0.76	1.53	2.43	8.39

4.2.3. Comparison of Classification Performance with Different Proportions of Training Samples

The proportion of training samples significantly impacts model classification effectiveness. In this study, we investigate how varying proportions of training samples influence classification performance across different models. We assess training sample proportions of 1%, 3%, 5%, and 7% for the IN and KSC datasets and 0.1%, 0.3%, 0.5%, and 0.7% for the SA dataset. As show in Figure 11, it is evident that for the SA dataset, characterized by distinct data boundaries, our model achieves a 98.78% classification accuracy using just 0.3% of the samples for training. This demonstrates our model’s reduced dependency on training sample size compared to the other models. For both the KSC and IN datasets, significant classification results were achieved using just 3% of the samples for training.

In contrast, the classification performance of the other networks varied significantly across different training sample sizes, whereas DCG-Net maintained a stable performance. This highlights DCG-Net’s robust generalization ability, consistently delivering excellent classification performance, even with limited training samples.

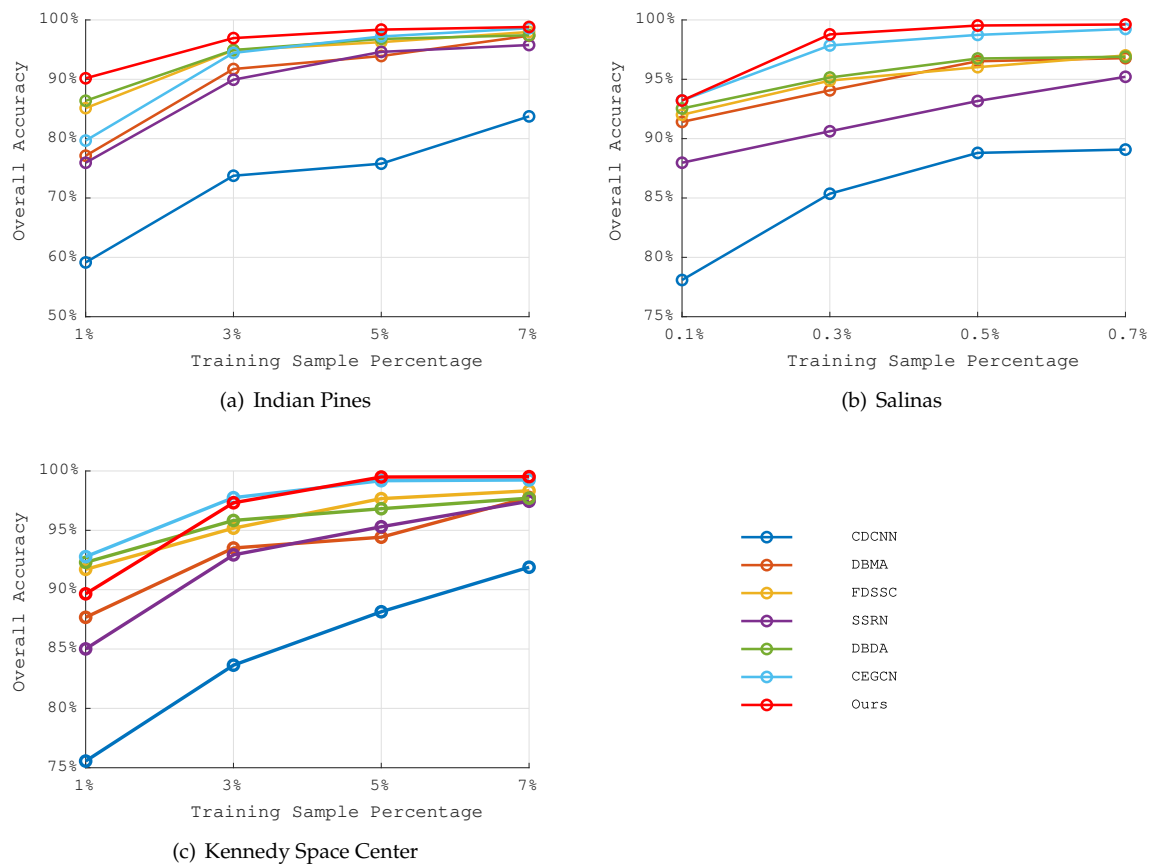


Figure 11. Comparison of classification performance of seven methods with different training set ratios for three datasets.

5. Discussion

The classification performance of the proposed DCG-Net is heavily influenced by the characteristics of the constructed graph, particularly the number of nodes and their connectivity properties. The number of superpixels dictates the number of nodes in the graph, while the value of K in the KNN algorithm determines its connectivity characteristics. We conducted two extended experiments to investigate the impact of these factors on DCG-Net's classification performance.

Furthermore, to assess the effectiveness of each module in our proposed DCG-Net, we performed ablation experiments focusing on the two branches of DCG-Net and the feature aggregation module.

5.1. Effects of the Number of Superpixels

The number of superpixels is proportional to the number of nodes in the graph, affecting the classification results of DCG-Net. In this section, the same number of superpixels is set for both the 1st and 2nd branches, with values of 64, 128, 256, 512, and 768. Figure 12 illustrates the impact of varying numbers of superpixels on the classification results across the three datasets. Specifically, for the IN dataset, Figure 12 shows that increasing the number of superpixels leads to a higher number of graph nodes and improves classification accuracy. When the number of superpixels reaches or exceeds 512, the segmentation results are satisfactory, with little difference in the classification accuracy compared to using 768 superpixels. For the SA dataset, changing the number of superpixels has a minimal effect on the classification accuracy. For the KSC dataset, the segmentation effect is poor, with 64 superpixels, but it improves as the number of superpixels increases.

Therefore, the number of superpixels determines the graph's node count. More superpixels result in more nodes. Consequently, as each node represents less sample information, more details are retained, leading to improved classification performance.



Figure 12. Comparison results of different superpixel numbers on three datasets.

5.2. Effects of the Value of K in the KNN Algorithm

In our study, the structure of a graph, particularly its edges, is crucial for defining its connectivity characteristics. Utilizing the K -nearest neighbors (KNN) algorithm, we designate a predetermined number of neighboring edges for each node within the graph. To examine how altering the number of neighboring edges influences the classification outcomes, we experimentally set the parameter ' K ' in the KNN algorithm to various values—specifically, 5, 10, 15, 20, and 25. It is important to note that the quantity of superpixels directly correlates with the total number of nodes in the graph; for consistency across experiments, we maintain this number at 512.

Figure 13 shows the classification performance across three distinct datasets under different configurations of ' K '. It can be observed that setting the number of neighboring edges too low results in poor node correlations and decreases the network's adaptive learning ability. Conversely, having too many neighboring edges causes the nodes to cover different categories, which adversely affects training effectiveness. In the analysis of Figure 13, when ' K ' is set to 15, the classification results for all the datasets are the best.



Figure 13. Comparison results of different K values on three datasets.

5.3. Ablation Study

In order to fully evaluate the algorithms we propose in this paper, we conducted a comprehensive test of the double-branch DCG-Net and feature aggregation module using ablation experiments. First, we evaluated the 1st branch and the 2nd branch of DCG-Net on three datasets separately, then we tested the impact of the feature aggregation module on the network. The experimental results are shown in Table 5.

Table 5. Comparison of different modules (× indicates that the module was not used in these experiments, ✓ indicates that the module was used).

Indian Pines						
No.	1st Branch	2nd Branch	FAM	OA (%)	AA (%)	Kappa
1	✓	×	×	97.78 ± 0.34	97.52 ± 0.89	97.47 ± 0.39
2	×	✓	×	97.29 ± 0.44	97.12 ± 1.20	96.19 ± 0.50
3	✓	✓	×	98.33 ± 0.57	96.60 ± 0.19	98.10 ± 0.66
4	✓	✓	✓	98.37 ± 0.47	97.64 ± 0.86	98.15 ± 0.66
Salinas						
No.	1st Branch	2nd Branch	FAM	OA (%)	AA (%)	Kappa
1	✓	×	×	99.40 ± 0.20	98.96 ± 0.50	99.38 ± 0.25
2	×	✓	×	94.37 ± 0.46	93.83 ± 1.22	93.74 ± 0.52
3	✓	✓	×	99.40 ± 0.20	98.96 ± 0.50	99.38 ± 0.25
4	✓	✓	✓	99.53 ± 0.29	99.22 ± 0.48	99.49 ± 0.29
Kennedy Space Center						
No.	1st Branch	2nd Branch	FAM	OA (%)	AA (%)	Kappa
1	✓	×	×	98.65 ± 0.58	98.50 ± 0.86	98.59 ± 0.64
2	×	✓	×	94.37 ± 0.46	93.83 ± 1.22	93.74 ± 0.52
3	✓	✓	×	98.89 ± 0.67	98.47 ± 1.03	98.76 ± 0.75
4	✓	✓	✓	99.49 ± 0.34	99.43 ± 0.55	99.44 ± 0.38

As shown in Table 5, the absence of any branch or module in DCG-Net impacts the classification results. For the Indian Pines and Salinas datasets, FAM improves all three metrics, particularly the average accuracy (AA). For the Kennedy Space Center dataset, FAM significantly boosts all three metrics. For the 2nd branch, on the Kennedy Space Center and Salinas datasets, the improvement in classification accuracy was smaller due to the distinct class boundaries. However, for the Indian Pines dataset, the classification accuracy showed a significant improvement. The 2nd branch refines the classification, enhancing the extraction of fine and edge features. It addresses the 1st branch's limitations in classifying small samples. The feature aggregation module enhances the overall classification accuracy of the network by dynamically learning channel features and combining them with the CNN, thus improving the model's robustness and adaptability.

6. Conclusions

In this study, we proposed DCG-Net, an innovative classification network that combines convolutional neural networks (CNNs) and graph convolutional networks (GCNs) to enhance hyperspectral image (HSI) classification. Our approach overcame the limitations of traditional methods by combining pixel-level and superpixel-level features, effectively handling both large-scale regular features and small-scale fine features. The key contributions of our work include the development of the expanding network (E-Net) for enhanced feature extraction, the integration of GCNs with an attention mechanism, and the creation of a feature aggregation module (FAM) for adaptive channel feature learning.

The comprehensive experiments conducted on three widely used hyperspectral datasets, Indian Pines, Salinas, and Kennedy Space Center, demonstrate the superiority of DCG-Net. Our model consistently outperformed state-of-the-art methods in terms of overall accuracy (OA), average accuracy (AA), and the kappa coefficient. Specifically, DCG-Net achieved an OA of 98.37% on the Indian Pines dataset, 99.53% on the Salinas dataset, and 99.49% on the Kennedy Space Center dataset. These results highlight the robustness and effectiveness of our proposed network in various HSI classification scenarios. Our findings indicate that the combination of CNN and GCN architectures, along with the use of attention mechanisms and feature aggregation techniques, significantly improves the classification performance of HSI.

However, our model has shortcomings in memory usage and running speed, and it requires further optimization to improve its performance and efficiency. Future research can explore further enhancements to this hybrid approach, including the incorporation of additional data augmentation techniques and the investigation of alternative graph structures to further optimize feature extraction and classification accuracy.

Author Contributions: Conceptualization, W.Z. and X.S.; methodology, W.Z.; software, W.Z. and Q.Z.; writing—original draft preparation, W.Z. and Q.Z.; writing—review and editing, W.Z. and X.S.; visualization, W.Z. and Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Data Availability Statement: All the datasets can be obtained at: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 20 May 2011). The code corresponding to the proposed method <https://github.com/229Kevin/DCG-Net> (accessed on 10 July 2024).

Acknowledgments: The authors would like to thank the editor and reviewers for their insights and comments.

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

References

1. Zhou, C.; He, Z.; Lou, A.; Plaza, A. RGB-to-HSV: A Frequency-Spectrum Unfolding Network for Spectral Super-Resolution of RGB Videos. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–18. [[CrossRef](#)]
2. Transon, J.; D'Andrimont, R.; Maignard, A.; Defourny, P. Survey of Hyperspectral Earth Observation Applications from Space in the Sentinel-2 Context. *Remote Sens.* **2018**, *10*, 157. [[CrossRef](#)]
3. Guo, T.; Luo, F.; Zhang, L.; Tan, X.; Liu, J.; Zhou, X. Target Detection in Hyperspectral Imagery via Sparse and Dense Hybrid Representation. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 716–720. [[CrossRef](#)]
4. Su, Y.; Gao, L.; Jiang, M.; Plaza, A.; Sun, X.; Zhang, B. NSCKL: Normalized Spectral Clustering with Kernel-Based Learning for Semisupervised Hyperspectral Image Classification. *IEEE Trans. Cybern.* **2023**, *53*, 6649–6662. [[CrossRef](#)] [[PubMed](#)]
5. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking Hyperspectral Image Classification with Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [[CrossRef](#)]
6. Gu, Y.; Chanussot, J.; Jia, X.; Benediktsson, J.A. Multiple Kernel Learning for Hyperspectral Image Classification: A Review. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6547–6565. [[CrossRef](#)]
7. Gerhards, M.; Schlerf, M.; Mallick, K.; Udelhoven, T. Challenges and Future Perspectives of Multi-/Hyperspectral Thermal Infrared Remote Sensing for Crop Water-Stress Detection: A Review. *Remote Sens.* **2019**, *11*, 1240. [[CrossRef](#)]
8. Lv, Z.; Wang, F.; Sun, W.; You, Z.; Falco, N.; Benediktsson, J.A. Landslide Inventory Mapping on VHR Images via Adaptive Region Shape Similarity. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5630211. [[CrossRef](#)]
9. Ham, J.; Chen, Y.; Crawford, M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [[CrossRef](#)]
10. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [[CrossRef](#)]
11. Prasad, S.; Bruce, L.M. Limitations of Principal Components Analysis for Hyperspectral Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 625–629. [[CrossRef](#)]
12. 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](#)]
13. Tang, Y.Y.; Yuan, H.; Li, L. Manifold-Based Sparse Representation for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7606–7618. [[CrossRef](#)]
14. Bai, J.; Yuan, A.; Xiao, Z.; Zhou, H.; Wang, D.; Jiang, H.; Jiao, L. Class Incremental Learning With Few-Shots Based on Linear Programming for Hyperspectral Image Classification. *IEEE Trans. Cybern.* **2022**, *52*, 5474–5485. [[CrossRef](#)] [[PubMed](#)]
15. Wu, Z.; Sun, J.; Zhang, Y.; Zhu, Y.; Li, J.; Plaza, A.; Benediktsson, J.A.; Wei, Z. Scheduling-Guided Automatic Processing of Massive Hyperspectral Image Classification on Cloud Computing Architectures. *IEEE Trans. Cybern.* **2021**, *51*, 3588–3601. [[CrossRef](#)]
16. Liu, Q.; Zhou, F.; Hang, R.; Yuan, X. Bidirectional-Convolutional LSTM Based Spectral-Spatial Feature Learning for Hyperspectral Image Classification. *Remote Sens.* **2017**, *9*, 1330. [[CrossRef](#)]
17. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317. [[CrossRef](#)]
18. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [[CrossRef](#)]
19. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral Image Classification Using Deep Pixel-Pair Features. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 844–853. [[CrossRef](#)]
20. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962. [[CrossRef](#)]
21. Zhang, S.; Xu, M.; Zhou, J.; Jia, S. Unsupervised Spatial-Spectral CNN-Based Feature Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [[CrossRef](#)]
22. Huang, L.; Chen, Y. Dual-Path Siamese CNN for Hyperspectral Image Classification with Limited Training Samples. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 518–522. [[CrossRef](#)]
23. Li, Y.; Zhang, H.; Shen, Q. Spectral-Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67. [[CrossRef](#)]
24. Ghaderizadeh, S.; Abbasi-Moghadam, D.; Sharifi, A.; Zhao, N.; Tariq, A. Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7570–7588. [[CrossRef](#)]
25. 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.* **2018**, *56*, 847–858. [[CrossRef](#)]
26. Wang, W.; Dou, S.; Jiang, Z.; Sun, L. A Fast Dense Spectral-Spatial Convolution Network Framework for Hyperspectral Images Classification. *Remote Sens.* **2018**, *10*, 1068. [[CrossRef](#)]
27. 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](#)]
28. Zhao, C.; Qin, B.; Feng, S.; Zhu, W.; Sun, W.; Li, W.; Jia, X. Hyperspectral Image Classification with Multi-Attention Transformer and Adaptive Superpixel Segmentation-Based Active Learning. *IEEE Trans. Image Process.* **2023**, *32*, 3606–3621. [[CrossRef](#)] [[PubMed](#)]

29. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Li, J. Visual Attention-Driven Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8065–8080. [[CrossRef](#)]
30. Pu, C.; Huang, H.; Yang, L. An attention-driven convolutional neural network-based multi-level spectral–spatial feature learning for hyperspectral image classification. *Expert Syst. Appl.* **2021**, *185*, 115663. [[CrossRef](#)]
31. Li, R.; Zheng, S.; Duan, C.; Yang, Y.; Wang, X. Classification of Hyperspectral Image Based on Double-Branch Dual-Attention Mechanism Network. *Remote Sens.* **2020**, *12*, 582. [[CrossRef](#)]
32. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2017**, [[CrossRef](#)]
33. 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](#)]
34. 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.* **2020**, *58*, 3162–3177. [[CrossRef](#)]
35. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. Multilevel Superpixel Structured Graph U-Nets for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [[CrossRef](#)]
36. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Cai, W.; Yu, C.; Yang, N.; Cai, W. Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification. *Neurocomputing* **2022**, *501*, 246–257. [[CrossRef](#)]
37. 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.* **2021**, *59*, 8657–8671. [[CrossRef](#)]
38. 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](#)]
39. Lu, T.; Li, S.; Fang, L.; Jia, X.; Benediktsson, J.A. From Subpixel to Superpixel: A Novel Fusion Framework for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4398–4411. [[CrossRef](#)]
40. 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](#)]
41. Liu, Q.; Xiao, L.; Yang, J.; Chan, J.C.W. Content-Guided Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 6124–6137. [[CrossRef](#)]
42. Wang, J.; Huang, R.; Guo, S.; Li, L.; Zhu, M.; Yang, S.; Jiao, L. NAS-Guided Lightweight Multiscale Attention Fusion Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 8754–8767. [[CrossRef](#)]
43. Wei, X.; Yang, Q.; Gong, Y.; Ahuja, N.; Yang, M.H. Superpixel Hierarchy. *IEEE Trans. Image Process.* **2018**, *27*, 4838–4849. [[CrossRef](#)]
44. Bai, J.; Shi, W.; Xiao, Z.; Regan, A.C.; Ali, T.A.A.; Zhu, Y.; Zhang, R.; Jiao, L. Hyperspectral Image Classification Based on Superpixel Feature Subdivision and Adaptive Graph Structure. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [[CrossRef](#)]
45. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; Volume 30, p. 3.
46. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5966–5978. [[CrossRef](#)]
47. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems NIPS'16, Barcelona, Spain, 5–10 December 2016; pp. 3844–3852.
48. Ma, W.; Yang, Q.; Wu, Y.; Zhao, W.; Zhang, X. Double-Branch Multi-Attention Mechanism Network for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 1307. [[CrossRef](#)]
49. Maveganzones. Hyperspectral Remote Sensing Scenes. Available online: https://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 20 May 2011).

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.