remote sensing

MDPI

Article

A New Dual-Branch Embedded Multivariate Attention Network for Hyperspectral Remote Sensing Classification

Yuyi Chen [1], Xiaopeng Wang [1,*], Jiahua Zhang [2,3], Xiaodi Shang [1], Yabin Hu [4], Shichao Zhang [1] and Jiajie Wang [1]

[1] College of Computer Science & Technology, Qingdao University, Qingdao 266071, China; chenyuyi@qdu.edu.cn (Y.C.); shangxd@qdu.edu.cn (X.S.); 2021010034@qdu.edu.cn (S.Z.); wangjiajie@qdu.edu.cn (J.W.)
[2] Key Laboratory of Earth Observation of Hainan Province, Hainan Aerospace Information Research Institute, Sanya 572029, China; zhangjh@radi.ac.cn
[3] Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[4] Lab of Marine Physics and Remote Sensing, First Institute of Oceanography, Ministry of Natural Resources, Qingdao 266061, China; huyabin@fio.org.cn
* Correspondence: wxp@qdu.edu.cn; Tel.: +86-0532-8595-0208

Abstract: With the continuous maturity of hyperspectral remote sensing imaging technology, it has been widely adopted by scholars to improve the performance of feature classification. However, due to the challenges in acquiring hyperspectral images and producing training samples, the limited training sample is a common problem that researchers often face. Furthermore, efficient algorithms are necessary to excavate the spatial and spectral information from these images, and then, make full use of this information with limited training samples. To solve this problem, a novel two-branch deep learning network model is proposed for extracting hyperspectral remote sensing features in this paper. In this model, one branch focuses on extracting spectral features using multi-scale convolution and a normalization-based attention module, while the other branch captures spatial features through small-scale dilation convolution and Euclidean Similarity Attention. Subsequently, pooling and layering techniques are employed to further extract abstract features after feature fusion. In the experiments conducted on two public datasets, namely, IP and UP, as well as our own labeled dataset, namely, YRE, the proposed DMAN achieves the best classification results, with overall accuracies of 96.74%, 97.4%, and 98.08%, respectively. Compared to the sub-optimal state-of-the-art methods, the overall accuracies are improved by 1.05, 0.42, and 0.51 percentage points, respectively. The advantage of this network structure is particularly evident in unbalanced sample environments. Additionally, we introduce a new strategy based on the RPNet, which utilizes a small number of principal components for feature classification after dimensionality reduction. The results demonstrate its effectiveness in uncovering compressed feature information, with an overall accuracy improvement of 0.68 percentage points. Consequently, our model helps mitigate the impact of data scarcity on model performance, thereby contributing positively to the advancement of hyperspectral remote sensing technology in practical applications.

Keywords: deep learning; two-branch network; feature classification; hyperspectral remote sensing; small-sample problem; Yellow River Estuary wetland

Citation: Chen, Y.; Wang, X.; Zhang, J.; Shang, X.; Hu, Y.; Zhang, S.; Wang, J. A New Dual-Branch Embedded Multivariate Attention Network for Hyperspectral Remote Sensing Classification. *Remote Sens.* **2024**, *16*, 2029. https://doi.org/10.3390/rs16112029

Academic Editors: Jie Wang, Xinxin Wang, Yongchao Liu and Xiaocui Wu

Received: 28 April 2024
Revised: 28 May 2024
Accepted: 3 June 2024
Published: 5 June 2024

1. Introduction

Hyperspectral remote sensing technology records both spectral and spatial data of the observed material in the form of hyperspectral images (HSIs), which contain rich information in narrow spectral bands, making the study of HSIs of great interest. It has thus been widely used in mineral surveys [1], agricultural evaluations [2], and urban and industrial infrastructure surveys [3]. However, despite the comprehensive number of features in HSIs,

their high dimensionality and nonlinearity are still quite challenging [4], and there is a need to provide effective dimensionality reduction and feature extraction methods. In addition, the problem of small sample sizes due to the difficulty of feature labeling likewise makes it a limitation in method selection. Thus, the attack and breakthrough of the difficulties of HSI data are of great significance to the research in the field.

Among the different aspects of HSIs, classification is one of the most researched tasks [5]. HSI classification, as a pixel-level classification task, can be classified into dimensionality reduction utilization and all-band utilization based on the amount of data utilized. In the early stage of HSI classification research, researchers mainly utilized all spectral feature design methods, where pixel vectors are directly classified with classifiers, such as Support Vector Machine (SVM) [6], Random Forest (RF) [7], and Polynomial Logistic Regression [8–10], and the features utilized are only shallow features [11,12]. However, continuous remote sensing band imaging leads to a large amount of redundancy in the raw spectral information, which poses a higher challenge for the classification task. Therefore, the combination of dimensionality reduction and feature extraction has been used to learn more discriminative features. Principal component analysis (PCA), as a typical dimensionality reduction method, can capture independent information as the principal components it retains exhibit orthogonal properties. In addition, it effectively reduces the effect of noise by focusing on the components that explain significant variance in the data. Therefore, it is widely used in HSIs. Kang et al. introduced the PCA-Based Edge-Preserving Features (PCA-EPFs) method, which comprehensively captures spatial information and substantially enhances SVM classification accuracy [13]. However, these machine learning methods can easily be limited in classification effectiveness due to the lack of fitting ability, even though they are robust to small-sample problems.

With the increasing maturity of deep learning applications in hyperspectral remote sensing, numerous deep learning methods have been employed for HSI classification in recent years, encompassing Deep Belief Neural Networks (DBNs) [14,15], Recurrent Neural Networks (RNNs) [16–18], Graph Convolutional Networks (GCNs) [19,20], and Convolutional Neural Networks (CNNs) [21–23], among others. Typically inspired by algorithms and techniques developed in the fields of computer vision and natural language processing, these methods demonstrate a wide variety of learning processes. Among them, CNNs, with their simple structural design and high processing efficiency, can be well applied to HSI classification tasks while achieving high accuracy, resulting in a breakthrough in the field [24]. For example, to extract both spectral and spatial features simultaneously, Chen et al. introduced Three-Dimensional CNN (3DCNN) into HSI classification, providing a new extraction method [25]. Subsequently, Zhong et al. also adopted this form of convolution, except that they flexibly adjusted the direction of convolution and divided it into spectral and spatial learning modules in cascade for feature extraction [26]. Gong et al. proposed that multi-scale convolution enriches the features of the image and integrates the feature information of interest in the image from a global perspective to achieve good results [27]. Recognizing the powerful fitting ability of CNNs, researchers began to consciously try to control the model complexity or use data augmentation to expand the number of samples to change the undesirable sample environment [28–30] to accommodate small-sample problems. Introducing the idea of separable convolutions is one of the more notable points in improving model complexity alone [5,31,32], which can be effective in improving the model complexity problem brought about by the increase in the number of parameters. From the aspect of improving model complexity, the introduction of separable convolutions as well as dilated convolutions has helped [5,31,32], while the latter reduces the depth of the network while achieving a higher sense of the field, realizing parameter reduction. Several excellent modules have been proposed [33–37], and the lightweight design they imply, while often losing a bit of accuracy, is still very desirable for reducing time loss.

In recent years, the Attention Mechanism (AM) has been introduced into the deep learning network structure, which can help HSIs to further break through the classification

challenge by assigning weights for the purpose of focusing on the features that affect the accuracy of the task. Among them, spatial attention, channel attention, spatial channel attention, and self-attention are currently the four most used AMs in HSI classification [38–41]. Scholars have integrated existing attention modules into the classification network of HSIs to enhance accuracy, but this approach often lacks specificity and may overlook unique relationships among neighborhood or global features [42,43]. To address this challenge, several novel attention mechanisms have begun to emerge, aiming to improve the inefficient characterization of features [44,45]. Among these, recent research has focused on the center pixel of HSIs, based on the prevailing input pattern of patch blocks [46,47]. This trend is driven by the presence of other classes within a block, making it challenging to effectively utilize or suppress these pixels. In a recent study, Li et al. introduced two self-similar modules to delve deeper into the spatial relationships guided by the center vectors inside patches in both input and feature spaces, significantly improving the ability to leverage information in subsequent feature extraction processes [48].

Although all these methods have obtained good results, the higher retention of the number of features after dimensionality reduction tends to increase the unnecessary computation. Utilizing the large amount of feature information contained in the first few principal components of the PCA dimensionality reduction for classification can theoretically achieve good results while avoiding excessive computational complexity. The Random Patches Network (RPNet) adopts random patches of the original data for feature extraction, combined with PCA to retain fewer principal components to reduce the amount of computation [49]. It adopts deep and shallow feature stacking to give it the advantage of being multi-scale, and it has better adaptability to the HSI classification of different objects, which often have different scales. Ultimately, it demonstrates higher results than unsupervised dimensionality reduction alone, which provides new ideas [50,51]. Deep learning models usually require many features to achieve better classification results, and the ability to directly utilize a small number of principal components for classification is limited. Therefore, it is natural to consider a way to explore the rich information contained in a small number of principal components, and the RPNet's stochastic convolution strategy can realize this very well.

Building on prior work, we introduce a Dual-Branch Embedded Multivariate Attention Network (DMAN) for HSI classification. This approach addresses the challenges of small sample sizes, including susceptibility to overfitting and misclassification in unbalanced environments. To address the issue of reduced bands after dimensionality reduction, we incorporate a hybrid stochastic convolution module to fully leverage potential information, enrich category features, and integrate them into the classification network. Recognizing the significance of feature refinement in multi-classification tasks, our network separately extracts spectral and spatial information to minimize feature interference [52–54]. Given the inherent limitations of small-sample problems, the strategic integration of multiple attention modules ensures a lightweight model that emphasizes critical feature information and intrinsic pixel connections. The primary contributions of this paper are outlined below.

1. To address the co-existence of insufficient feature extraction and overfitting problems, the proposed model incorporates multiple multi-scale strategies to enhance the information captured in each layer and furnish rich features for the fitted samples. Additionally, attention modules with fewer or even no parameters are integrated at appropriate locations in the model, facilitating the accurate and efficient identification of feature types.
2. Building upon the enhancements to the random convolution module, a hybrid random patch module (HRPM) is introduced. This module empowers the classification model to sustain high accuracy with fewer principal components. Subsequently, a deep learning model leveraging this module is proposed for application in classification problems involving a reduced number of bands after dimensionality reduction.
3. Our proposed DMAN method is validated on two public datasets and our own labeled datasets (IN, UP, and YRE, respectively), demonstrating markedly superior

classification results. In particular, in scenarios of sample scarcity, our method surpasses traditional deep learning methods in accuracy, offering robust validation of its effectiveness.

The rest of this paper is structured as follows. Section 2 specifies the design details of the DMAN. Section 3 describes the dataset and characteristics used and provides the necessary analysis of the experimental results. Section 4 provides a comprehensive discussion of the reasons for the advantages of the proposed method based on the experimental results, as well as the differences and commonalities with the comparative algorithms. Section 5 summarizes the core of the full paper and provides suggestions for future research.

2. Methods

This section provides a detailed description of the structure of the two-branch network model proposed in this paper, encompassing the hybrid random convolutional module, the composition of the two branches—spatial and spectral—and the structure of the pyramid pooling attention. The overall model structure is illustrated in Figure 1.

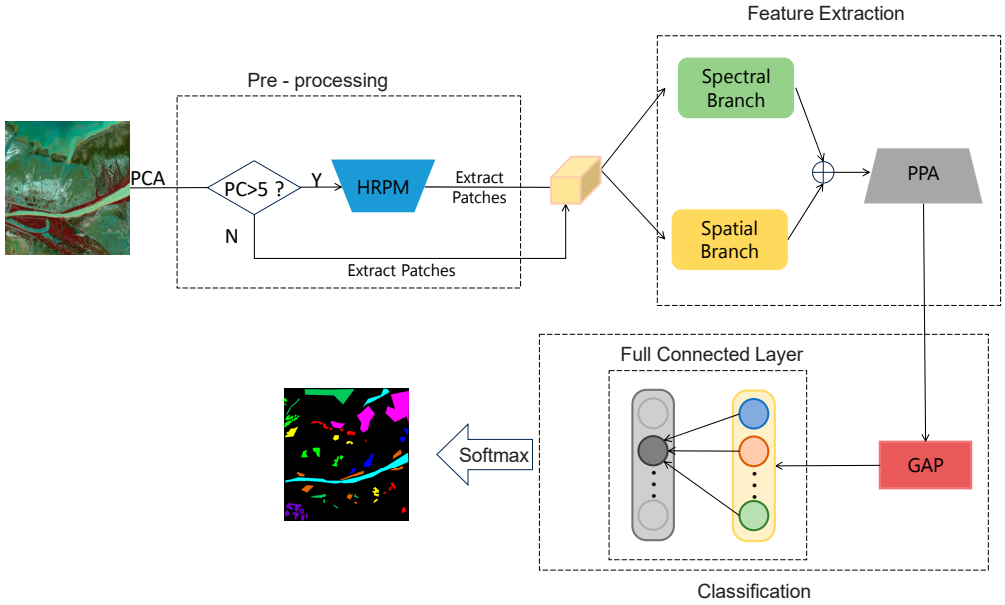


Figure 1. The overall architecture of the proposed DMAN model, where PC stands for principal component, PCA stands for principal component analysis, GAP stands for global average pooling, and HRPM stands for hybrid random patch model.

2.1. Hybrid Random Patch Model

As mentioned earlier, the RPNet already has good feature extraction capabilities, while this network and its current improvements are usually used to solve the problem that neural networks need to be trained, ignoring further research on classification in small-sample situations. In addition, the RPNet is often used for the classification process in combination with machine learning algorithms such as SVMs and graph-based learning methods. Deep learning methods are considered for applications less often, which may lead to limited classification results. In addition, since it is essentially a 2D convolution in terms of convolution, the spectral dimension information may not be well preserved. Moreover, in the application of deep learning methods, a small number of bands after PCA are retained for classification and are seldom considered by scholars because these bands are often not suitable for model construction even though they already contain a large amount of information.

To address the above problems, we designed a Hybrid Random Patch Module (HRPM) that combines multiple features. It is aimed at improving the extraction of classification information from fewer spectral features and replaces part of the convolutional layers to

reduce the computation. It combines shallow and deep convolutional layers and has the advantage of being multi-scale, which provides better adaptation to the HSI classification of different objects, often at different scales. This strategy solves the problem of information loss during hierarchical feature extraction and performs effective learning and inference at different scales. It is located in the data preprocessing part of the whole classification process, which can effectively map the original features into a contributing information flow, fully considering the randomness of spatial and spectral dimensions. The combination of the HPRM with the classification network can further explore the deep features in a small number of bands, effectively retain the information that is beneficial to the classification, and increase the inter-class separability.

As shown in Figure 2, the HRPM adopts a cascade structure containing PCA and constructs random patch blocks, a convolution-type selection operation module, and a Rectified Linear Unit (ReLU), described in detail, as follows. First, PCA is used for dimensionality reduction, the computational and information component contents are weighed, considering the depth needed for 3D convolution, and 5 principal components are retained. Then, the convolution block of $k \times k \times n$ is randomly selected in the dimensionality-reduced data cube for convolution operation. For example, the output channel is set to o, the padding is set to be equivalent to when the operation mode is selected as a 2D convolution, and the output size is (w, h, and o). When it is a 3D convolution, the output size is (w, h, n, and o), further reshaping the shape to (w, h, and n*o). Then, it is sent to the next random layer for convolution. Finally, we stack these feature blocks in the third dimension and feed them to the classifier for processing.

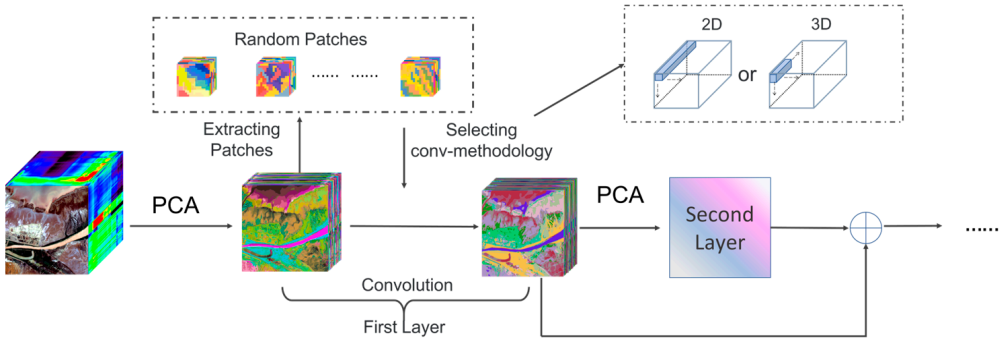


Figure 2. The proposed HRPM structure.

2.2. Spectral Branch

The spectral branching structure is shown in Figure 3. The spectral branch convolution method uses 3D convolution with a convolution kernel shape of $1 \times 1 \times k$, aiming to extract only spectral features. Specifically, a convolution kernel of size $1 \times 1 \times 5$ with a step size of 2 is first used to downscale the HRPM-extracted features. Immediately after that, the data stream is split into two; one part prepares the end-of-branch jump connections to ensure the smooth gradient conduction and acquisition of global features in the spectral dimension, and the second part is fed into the multi-scale convolution module, i.e., features are extracted at smaller scales of 3, 5, and 7 in the channel dimension, respectively, and padded to ensure that the output size of the feature map remains constant. This operation obtains rich spectral local feature information while better adapting to various details in the data. Immediately after splicing at the channel dimension and using convolution to migrate the feature dimensions to fit the next channel attention operation, the weights assigned to each feature map are computed and assigned. Finally, the feature mappings obtained from the previous branches are summed with the main channel.

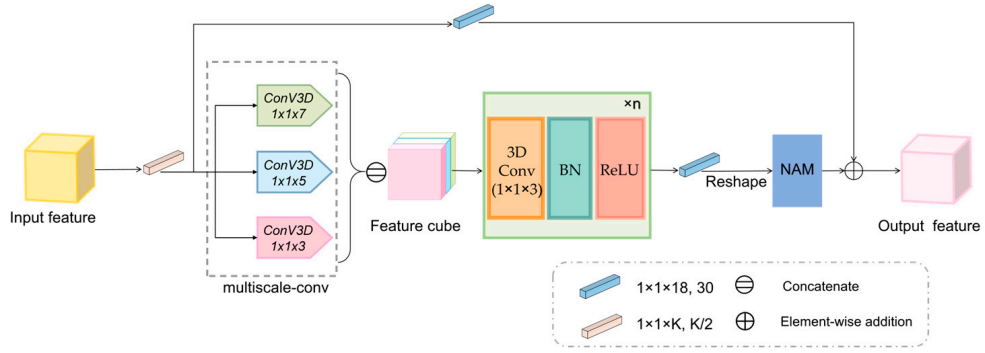


Figure 3. The proposed spectral branch structure.

2.3. NAM

In order to ensure a lightweight model, the channel attention uses the NAM [55]. This module does not require additional computational and parametric operations such as full connectivity, convolution, etc., but calculates the attention weights directly with the help of the scale factors in the batch normalization, and the whole process is shown in Figure 4 and Equation (1), where, at the beginning, the scale factor Y is learned through the batch normalization (BN) of the input features. Based on this, a weight factor $\omega_i = Y_i / \sum_{j=0} Y_j$ which represents the proportion of the individual channel scale factors in all the channels, is calculated. Then, the processed features are multiplied channel by channel with the weight factor using Equation (1), and finally, after the activation function *sigmoid* is nonlinearized, the weight of each channel attention module, M_c, can be obtained, as shown in Equation (1).

$$M_c = sigmoid(W_Y(BN(F_1))) \tag{1}$$

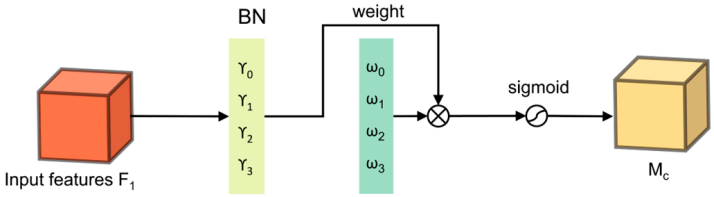


Figure 4. The proposed NAM structure.

The BN process is shown in Equation (2), where $\mu_\mathcal{B}$ and $\sigma_\mathcal{B}$ denote the mean and standard deviation of each batch, respectively, and β denotes the bias term. The BN layer needs to compute the mean and variance of all elements in a minibatch input feature, i.e., Bin, then divide the standard deviation by the subtracted mean, and finally, perform affine transformation using the learnable parameters γ and β to obtain the final BN output Bout.

$$B_{out} = BN(B_{in}) = Y\frac{B_{in} - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} + \beta \tag{2}$$

Therefore, the NAM makes clever use of the variability of the information deflation process, i.e., how much information is embedded in the channel to grasp the corresponding weights assigned to the channel, a process that is both lightweight and practical at the same time. The higher the normalized value, the more information the channel has and the more attention will be given to it for recalibrating the features of the original feature cube at a faster rate.

2.4. Spatial Branch

Most two-branch networks usually use 3D convolution to extract spatial information independently on spatial branch extraction, which indeed ensures feature purity in the spatial dimension. However, often the number of channels of the two branches is artificially

set to be equal before network fusion. In reality, the degree of contribution of the two features is often not equal, which is due to the inherent defect of low spatial resolution of HSIs. To bridge this gap and preserve this network design, our convolution approach uses 2D convolution, which instead aids in spatial branching to extract features because this convolution utilizes channel feature fusion computation. In this, we take the expansion of the convolution kernel and pooling to expand the sensory field to make the extracted information more globally characterized. The spectral branching structure is shown in Figure 5, and we describe the detailed operation below.

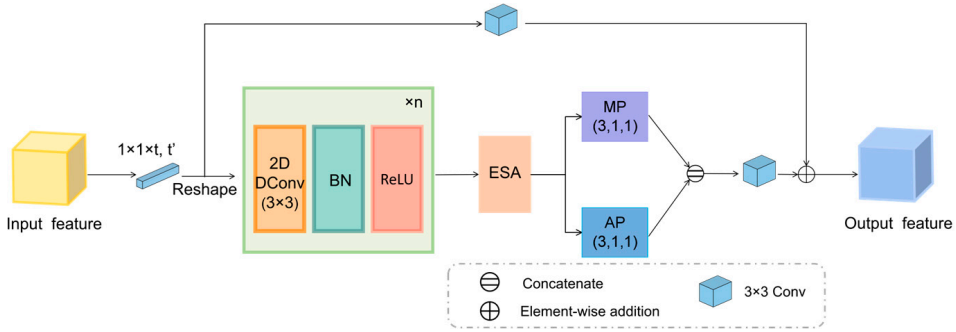


Figure 5. The proposed spatial branch structure.

The spatial branch first compresses the channel using 3D convolution, performs feature migration, and reshapes it into an applicable 2D convolution shape. Then, it passes through n consecutive null convolution modules, where the null rate is set to a consecutive natural number to increase the sensory field while avoiding the mesh effect. Subsequently, the data stream is divided into two to go through maximum pooling (MP) and average pooling (AP) to obtain spatial texture information and background information, respectively. The two pooling kernels have a size of 3, a step size of 1, and a padding of 1. This is to ensure that the degree of information loss is reduced. Then, the merged convolution is fed to the spatial attention module. Finally, the same jump join operation is performed to obtain the output features.

2.5. ESA

HSI classification methods based on block-predicted center pixels are increasingly focusing on the relationship and importance between the center vector and the rest of the block vectors as this is often a critical factor in determining the classification results. At the same time, the distribution of pixels in spatial locations can easily result in the classification of categories dominated by large areas into positive categories, which often turn out to be inaccurate. Building on the work of Li et al. [36], Euclidean Similarity Attention (ESA), which utilizes Euclidean similarity for center pixel-related texture feature extraction, was designed, with the architecture shown in Figure 6.

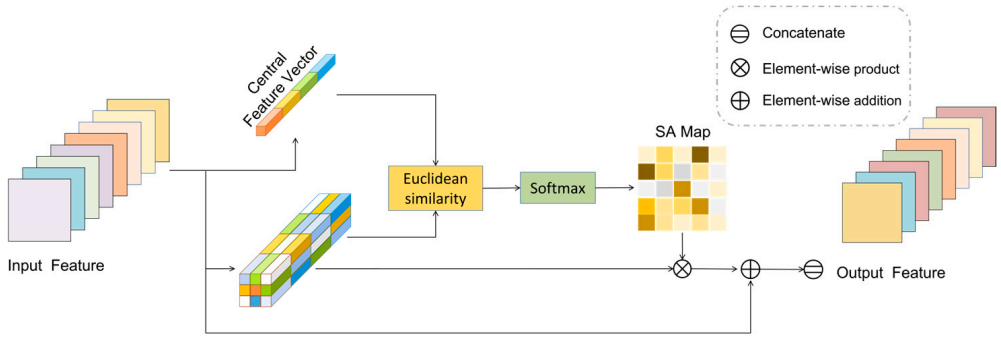


Figure 6. The proposed ESA structure.

Specifically, to make full use of the feature information of other classes in the patch to assist classification, we calculated the Euclidean similarity of the center pixel with other pixels; the process is shown in Equation (5). In addition, the weights of the spatial attention of the pixel points were obtained by using Softmax to multiply them and sum them with the original feature points.

$$E_{i,j} = EDSim(x_{o,o}, x_{i,j}) = \frac{1}{1+\|x_{o,o}-x_{i,j}\|} \tag{3}$$

where a represents the Euclidean similarity between the vectors, the block size is set to w × w, and b represents the center vector, where o = w/2, i = 1,..., w and j= 1,..., w.

2.6. PPA

Joint spatial–spectral features have a stronger characterization ability, and the appropriate extraction method can improve the robustness of the classifier. Moreover, it is reasonable to solve the limitation of single-size feature learning in the "small-sample problem" of HSI classification tasks to improve the classification accuracy. Therefore, we designed the PPA module from the perspective of expanding the learning feature surface and mitigating overfitting; the architecture is shown in Figure 7. Pooling is first learned using three different sizes and step sizes, where bilinear interpolation (BI) is used to smooth the feature loss. A shared convolutional layer immediately follows to resize the channels while reducing the computational effort. These three channel descriptors are then spliced, and a 3 × 1 convolution operation is used to fuse them. Next, the cascade features are weighted using weight coefficients. Finally, to avoid vanishing gradients, residual joins are introduced to sum the weighted feature maps with the original feature maps. The detailed operation is described below.

$$F_n = f^{s\times s}(p^{n\times n}(M_{ss})) \tag{4}$$

$$F_m = [F_{n_1}; F_{n_2}; F_{n_3}] \tag{5}$$

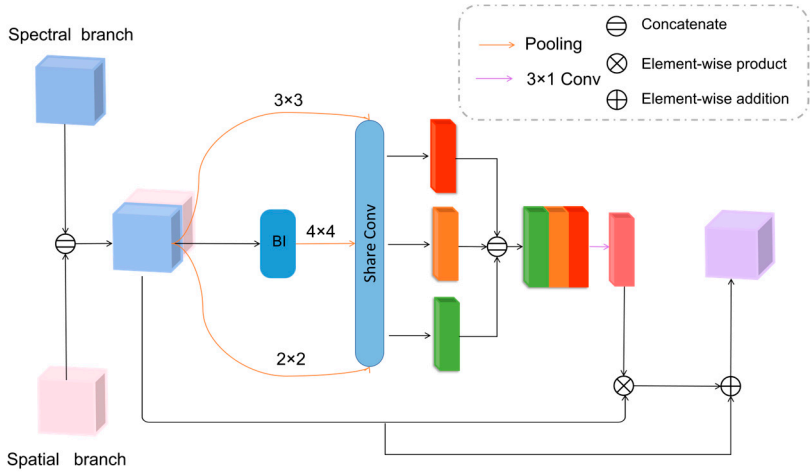


Figure 7. The proposed PPA structure.

In Equations (4) and (5), M_{ss} represents the spliced input of the two types of features, $P^{n\times n}$ represents the pooling of the pooling kernel of size n, where the bilinear interpolation operation of one of the taps is omitted, and $f^{s\times s}$ represents the shared convolutional layer, which will reduce the size of the three feature maps to 1. The three tap feature maps are then spliced together in the length dimension, which represents the integration of the information learned at the different scales, i.e., F_m.

$$M(F_m) = \sigma(f^{3\times 1}(F_m)) \tag{6}$$

$$F = M(F_m) \times M_{ss} + M_{ss} \tag{7}$$

Next, as shown in Equation (6), these multi-scale features are effectively fused. This fusion process aims to make the network more comprehensive in perceiving the information at different scales, thus improving the accuracy of the importance of the assignment to the original data. Finally, after obtaining higher-level learning weights for the multi-scale features, the original inputs are multiplied and summed with the constructed attention weights wave by wave and dotwise, as shown in Equation (7).

3. Results

The experimental part is organized as follows. First, the details of the three HSI datasets are presented in detail, which include two public datasets as well as our own labeled dataset. Second, the hyperparameter settings of the experiment are presented. Then, we compare our model to other methods to prove the advancement of the proposed method. The contribution of each module is explored through ablation experiments.

3.1. Datasets

In our experiments, two commonly used HSI datasets were used, namely, the Indian Pines (IP) and University of Pavia (UP) datasets. In addition, we preprocessed and annotated the data from the Yellow River Estuary wetland (YRE) in Dongying, Shandong Province, China, and we additionally performed image fusion in order to obtain pure image elements. Each dataset has its specification, which is described as follows.

Indian Pines (IP): The IP dataset is an important hyperspectral remote sensing image resource that was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in June 1992 at the Indian Pines Agricultural Experimental Range in northwestern Indiana. The dataset contains 145 × 145 pixels and has a spatial resolution of 20 m. During data preprocessing, 200 bands were screened from the raw data in the 400–2500 nm wavelength range by removing 20 absorbing and low signal-to-noise bands. In effect, 16 feature classes covering 10,249 labeled pixels were analyzed in detail, with a significant imbalance in the number of samples in some of these classes.

University of Pavia: The UP dataset was collected using the Reflectance Optical System Imaging Spectrometer (ROSIS) in the urban area surrounding the University of Pavia in northern Italy. The dataset has a pixel size of 610 × 340, covers a wavelength range of 0.43–0.86 μm, and contains 115 spectral bands with a spatial resolution of 1.3 m and a spectral resolution of 4 nm. A total of nine classes including 42,776 labeled pixels were covered, and after removing the 12 noisy bands, we based our classification analysis on the remaining 103 bands. Compared to the IP dataset, even though the UP dataset contains fewer bands, it still has a high dimensionality.

Yellow River Estuary wetland: The YRE was acquired by GF5_AHSI in 2019 at the mouth of the Yellow River in Dongying City, Shandong Province, China, covering the extent of the core area of the Yellow River Delta National Nature Reserve. The overall image contains 330 spectral bands in the wavelength ranges of 390–1029 nm (VNIR) and 1005–2513 nm (SWIR). A total of 50 bad bands were removed, and the remaining 280 bands were used for classification with a spatial resolution of 30 m. It is worth noting that to improve the spatial resolution and reduce the effect of mixed pixels, we fused the panchromatic bands of Landsat8 (Landsat8_Pan) satellite images with the preprocessed GF5 hyperspectral at the same time. The fusion algorithm used the Gram–Schmidt transform method to improve the spatial resolution to 15 m while ensuring less loss of spectral information. Both images before fusion were subjected to a series of preprocessing through radiometric calibration, atmospheric correction, and geometric fine correction and cropping, as shown in Figure 8. The classification data used in this paper were cropped in the center part of the fused image. The dataset size was 710 × 673 pixels, with nine feature classes covering 79,732 labeled pixels.

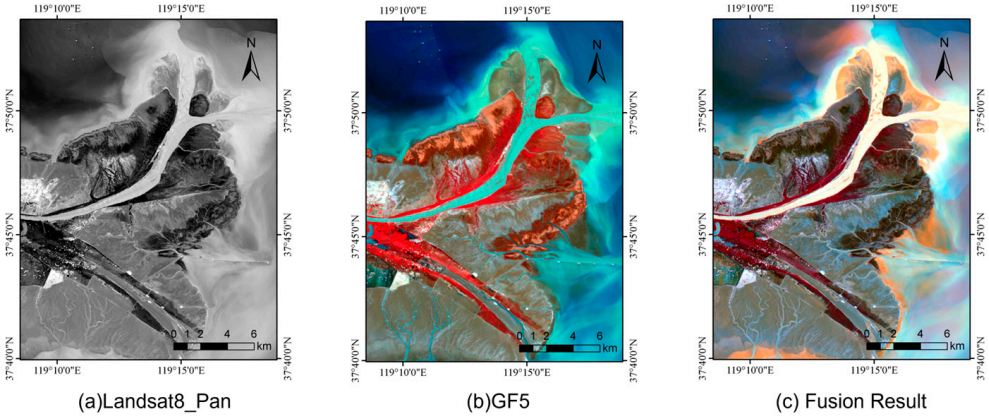


Figure 8. Comparison between pre- and post-fusion images of YRE data fusion.

The false-color band settings in Figure 8b,c are uniform, i.e., 110, 58, and 38, respectively. The spatial resolution of the fused image shows more details and richer color levels. The number of spectral bands corresponds to that before fusion.

Each dataset was divided into three parts, i.e., the training, validation, and test sets, where the training and validation sets were in equal proportions, the remainder was the test set, and all the samples were randomly selected. The training set was used to iteratively find the optimal parameters of the model, the validation set only shows the results of the interim model simulation measurements for each training phase, and the test set was used to evaluate the effect of the model that performs optimally in the validator. Due to the difference in the overall sample order of magnitude and the preservation of the original dataset number distribution, we used 5% of training samples for the Indian Pines (IP) dataset, which has fewer samples, while the University of Pavia (UP) dataset and Yellow River Estuary wetland (YRE) dataset had 0.5% and 0.1% training samples, respectively, with the specific number of pixels shown in Tables 1–3.

Table 1. Groundtruth classes for the IP scenes and their respective sample number.

No.	Class	Train/Val	Test	Total
1	Alfalfa	2	42	46
2	Corn-notill	71	1286	1428
3	Corn-mintill	41	747	830
4	Corn	12	213	237
5	Grass-pasture	24	435	483
6	Grass-trees	37	657	730
7	Grass-pasture-mowed	1	26	28
8	Hay-windrowed	24	430	478
9	Oats	1	18	20
10	Soybean-notill	49	874	972
11	Soybean-mintill	123	2209	2455
12	Soybean-clean	30	533	593
13	Wheat	10	185	205
14	Woods	63	1139	1265
15	Buildings-grass-trees-drives	19	348	386
16	Stone-steel-towers	5	83	93
	Total	512	9225	10,249

Table 2. Groundtruth classes for the UP scenes and their respective sample number.

No.	Class	Train/Val	Test	Total
1	Asphalt	33	6565	6631
2	Meadows	93	18,463	18,649
3	Gravel	10	2079	2099
4	Trees	15	3034	3064
5	Painted metal sheets	7	1331	1345
6	Bare soil	25	4979	5029
7	Bitumen	7	1316	1330
8	Self-blocking bricks	18	3646	3682
9	Shadows	5	937	947
	Total	213	42,350	42,776

Table 3. Groundtruth classes for the YRE scenes and their respective sample number.

No.	Class	Train/Val	Test	Total
1	Mudflat	4	3575	3583
2	Bare soil	5	5376	5386
3	Tamarix	4	3574	3582
4	Suaeda salsa	4	4235	4243
5	Spartina alterniflora	23	23,489	23,535
6	Turbid water	17	16,984	17,018
7	Tidal flat reed	6	5975	5987
8	Clear water	12	12,229	12,253
9	Bare lake beach	4	4137	4145
	Total	79	79,574	79,732

3.2. Experimental Configuration

For a fair comparison, all the experiments were conducted on a workstation running Windows 10 with an NVIDIA GeForce RTX3090 graphics card and 64 G of RAM. The workstations were manufactured by Dell in Qingdao, China. The deep learning framework used was Pytorch 1.10, and the programming language was Python 3.6. Both the model in this paper and the comparison model were trained with an initial learning rate of 0.001; Adam with a momentum term of 0.9 and weight decay of 0.01 was chosen as the optimizer to optimize the network model, and the cross-entropy loss function was used for the experiments. In addition, to speed up the training, we set the batch size to 32 and the number of epochs to 100. The rest of the parameters were consistent with those described in the respective original papers.

In order to evaluate the performance of the model classification, the reliability of the predicted classification maps compared to the ground truth maps was obtained. In the experiment, three common evaluation indexes were used, namely, Overall Accuracy (OA), Average Accuracy (AA), and Kappa Coefficient, to evaluate it. OA (Overall Accuracy) represents the ratio of correctly classified samples to the total test samples, i.e., the overall effect of classification can be visualized. AA (Average Accuracy) represents the average of all category Recalls, and based on the common situation that there is an imbalance of samples in multi-categorization, this metric can reflect the model's ability to assess the accuracy of classification for fine-grained evaluation, reflecting the robustness of the model. The Kappa Coefficient measures the degree of agreement between the true values and the classification results. The values of these three evaluation indexes are positively correlated with the classification effect.

3.2.1. Effectiveness of the k-Value in the HRPM Structure

The HRPM was initially designed to ensure the robustness of the classification results, mainly in the case of fewer principal components, while other deep learning methods have

a greater demand for the number of principal components. Therefore, we only explore the effect of the number of kernels (k) it extracts per layer on the accuracy in this subsection, and subsequent comparison experiments will not be added to this module. In addition, we chose to keep five principal components to facilitate the smooth learning of deep features for 3D convolution. We first experimented with the original DMAN structure and then experimented with the DMAN structure combined with the HRPM for different values of the hyperparameter k (6, 8, 10, 12, 14, and 16). As shown in Table 4, the poor classification results for direct feature extraction for the five principal components occur because the information of a small number of principal components is not well captured by the network, and the difficulty of extraction is further exacerbated by the conditions of the samples that we chose. It is 7.65% lower compared to the HRPM (k = 12) after incorporation, which indicates that random extraction before HRPM training is beneficial in highlighting the discriminative features of the samples in the small-sample case. Thus, combining shallow and deep features like this gives the extracted features a multi-scale advantage, and the hybrid convolution further enriches the features so that the classification information is better utilized by the model.

Table 4. Effect of the number of convolution kernels (k) on accuracy.

IP (5%)	Without	HRPM (k-Value)					
		6	8	10	12	14	16
OA (%)	87.74	93.32	93.92	94.74	95.39	95.12	95.58
AA (%)	73.60	83.62	88.90	90.42	91.86	90.48	92.83
Kappa × 100	86.03	92.38	93.05	94.00	94.74	94.43	94.95

However, when the k-value is set to 6 or 8, the performance drops slightly, which occurs because fewer convolutional kernels are unable to carry the diversity of features, resulting in a loss of information. There is a general trend of increasing accuracy as the k-value increases. However, when the k-value exceeds 12, the classification accuracy starts to fluctuate as the information starts to be redundant and the training slows down due to the presence of more features in the input data. Therefore, based on the experimental results and time cost, we set the k-value to 12.

3.2.2. Effect of the Number of Training Samples

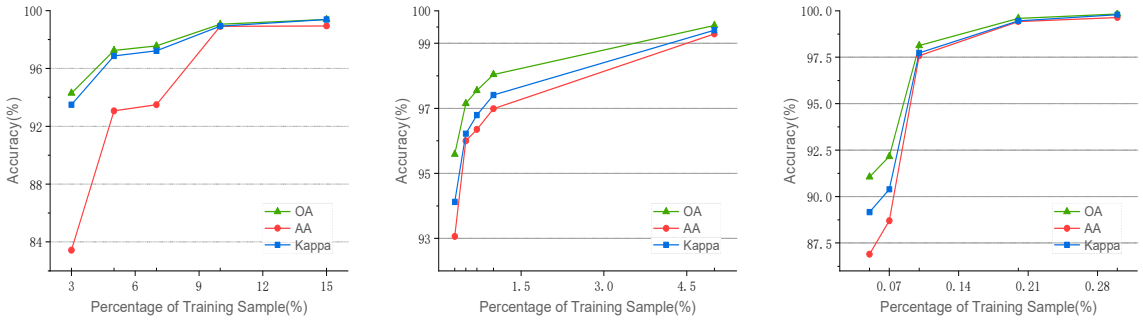
To further analyze the effect of the number of training samples on the proposed DMAN, we chose different proportions of training sets for the experiments based on the characteristics of the sample distribution and the number of samples in each dataset. The proportion of training samples for three of them for the three datasets is shown in Table 5. The validation set had the same number of data samples as the partitioned training set, and the remaining part was used as the test set.

Table 5. Percent of Training Samples.

Dataset Name	Percent of Training Samples (%)				
IP	3	5	7	10	15
UP	0.3	0.5	0.7	1	5
YRE	0.05	0.07	0.1	0.2	0.3

Figure 9 shows the classification results with different training sample ratios. The vertical axis is the accuracy of the evaluation metrics, and the horizontal axis is the training set ratio. As expected, the accuracy values for all three datasets increase with the number of training samples and stabilize after reaching a certain ratio. For the IP dataset, the gap between the AA value and the other two metrics is too large at the beginning, which is due to the more heterogeneous distribution of the data volume. The effect stabilizes after

the proportion reaches 10%. The spatial resolution of the UP and YRE datasets is higher. Therefore, the metrics are all superior to the IP dataset with the same number of training samples. The UP dataset has high spatial resolution, and the YRE dataset is rich in spectral features; so, they do not need a large sample size to achieve close to 100% accuracy.



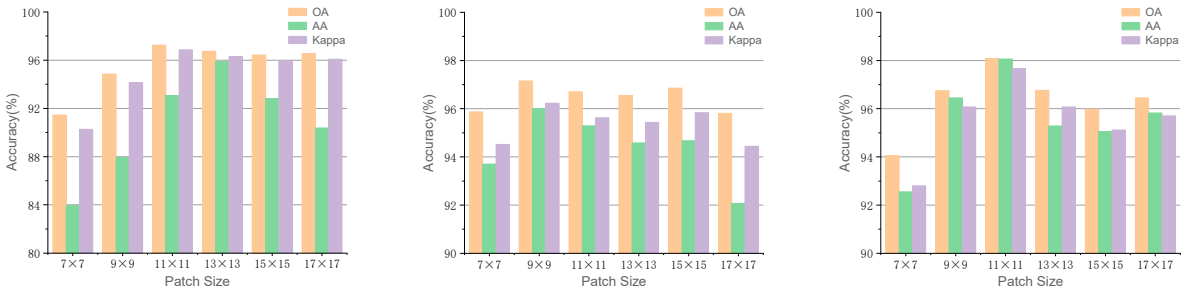
(a) IP dataset (b) UP dataset (c) YRE dataset

Figure 9. The classification results of the DMAN with different training samples.

3.2.3. Effect of Patch Size

Another important factor that affects the performance of the network is the size of the patch. Usually, patches that are too small contain less feature information and have difficulty fitting the target, and patches that are too large may be harmful to the classification process as they tend to introduce noise, especially when the categories are closely connected.

In this section, we analyze the effect of patch size on the proposed method based on the initially described dataset division ratio and parameters. Figure 10 presents the classification results of the proposed method under different patch sizes. For the IP dataset, there is a lag in the trend of AA compared to OA, which may be because larger patches have a greater enhancement of the classification effect of certain classes, while the overall tendency is stabilized. For the UP dataset, the three performance metrics are optimal at a patch size of 9. This is because the labels of the UP dataset are very detailed and are suitable for small patches. For the YRE dataset, there is a significant decrease in the size of the patch after finding the most suitable one, which may be because there are more edge pixels in the YRE dataset, which may contain more external padding for larger sizes, resulting in poor classification results.



(a) IP dataset (b) UP dataset (c) YRE dataset

Figure 10. The classification results of the DMAN with different patch sizes.

3.3. Results and Analysis

To validate the performance of our designed network, we selected the DMAN network for comparison to six classification networks, namely, the HybridSN, DFFN, SSRN, RSSAN, GSC-ViT, AMS-M2ESL, FDSSC, and SSFTT, which were classified into unsupervised dimensionality reduction and all-band utilization input according to the data utilization. For the fairness of the comparison, we replaced the AMS-M2ESL dimensionality reduction method

with PCA. In addition, we did not include the HRPM in this section because it manifests a significant effect when the number of bands after dimensionality reduction is small, while the accuracy does not increase significantly when more bands are retained. Therefore, we used pre-PCA downscaling of the data in this section to compare to other methods, in which 40 principal components are retained. The classification plots obtained for all methods are shown in Figures 10–12, and the classification accuracies of these methods on the three datasets are shown in Tables 6–8 where the highest accuracies in the classification results are shown in bold.

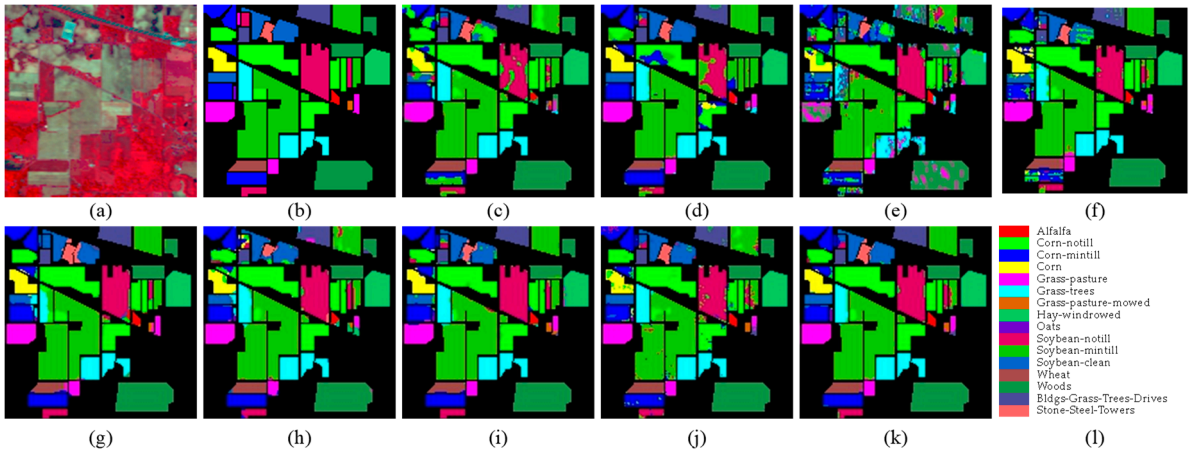


Figure 11. Classification maps of different methods on the IP dataset. (**a**) False-color; (**b**) ground truth map; (**c**) FDSSC; (**d**) SSRN; (**e**) RSSAN; (**f**) GSC-ViT; (**g**) DFFN; (**h**) HybridSN; (**i**) SSFTT; (**j**) AMS-M2ESL; (**k**) DMAN; and (**l**) color bar.

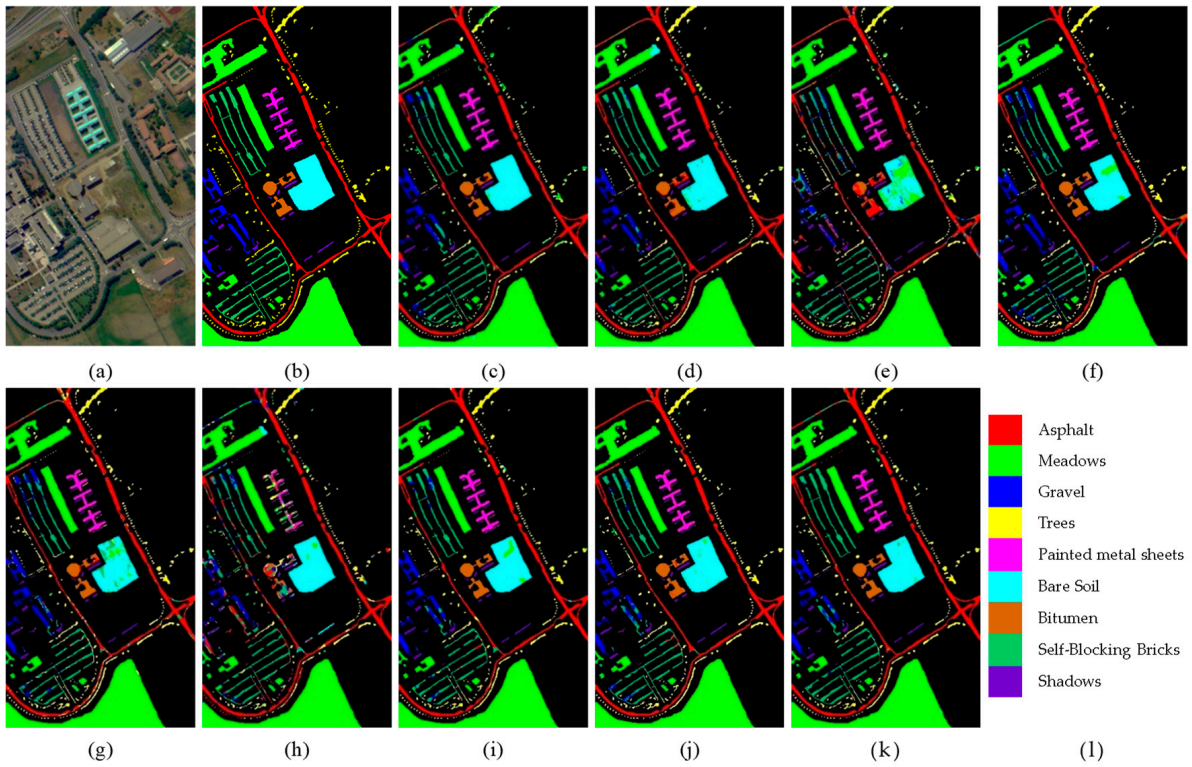


Figure 12. Classification maps of different methods on the UP dataset. (**a**) False-color; (**b**) ground truth map; (**c**) FDSSC; (**d**) SSRN; (**e**) RSSAN; (**f**) GSC-ViT; (**g**) DFFN; (**h**) HybridSN; (**i**) SSFTT; (**j**) AMS-M2ESL; (**k**) DMAN; and (**l**) color bar.

Table 6. The classification results (%) of all compared methods on the IP dataset.

Class	Name	All Bands				PCA				Proposed
		FDSSC	SSRN	RSSAN	GSC-ViT	DFFN	HybridSN	SSFTT	AMS-M2ESL	
1	Alfalfa	**100.0 ± 0.0**	91.89 ± 7.23	90.00 ± 6.93	**100.0 ± 0.0**	65.00 ± 11.22	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
2	Corn-notill	69.64 ± 3.45	**99.39 ± 0.36**	73.47 ± 1.03	82.92 ± 1.43	91.64 ± 2.01	88.07 ± 0.96	98.01 ± 1.07	87.97 ± 1.34	94.93 ± 1.35
3	Corn-mintill	**98.32 ± 0.98**	72.50 ± 2.03	73.14 ± 3.44	94.89 ± 0.67	95.77 ± 0.32	93.80 ± 1.42	92.45 ± 0.63	91.68 ± 1.22	96.23 ± 1.2
4	Corn	95.51 ± 2.34	81.15 ± 4.3	93.84 ± 4.45	75.56 ± 2.53	95.95 ± 2.7	85.71 ± 2.33	97.20 ± 2.06	97.18 ± 3.25	**98.57 ± 1.37**
5	Grass-pasture	**100.0 ± 0.0**	97.07 ± 2.34	49.62 ± 2.7	88.67 ± 3.92	94.33 ± 1.1	94.57 ± 3.7	95.89 ± 2.11	99.24 ± 0.42	96.43 ± 3.02
6	Grass-trees	95.33 ± 3.63	95.63 ± 0.57	71.73 ± 8.25	98.89 ± 0.25	91.30 ± 0.53	93.05 ± 1.04	97.75 ± 0.33	98.05 ± 0.52	**99.53 ± 0.19**
7	Grass-pasture-mowed	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	66.67 ± 8.22	86.36 ± 4.28	90.00 ± 6.21	72.73 ± 8.97	**100.0 ± 0.0**	**100.0 ± 0.0**
8	Hay-windrowed	96.39 ± 2.88	99.76 ± 0.19	98.71 ± 0.41	97.28 ± 2.45	99.54 ± 0.24	90.15 ± 2.2	98.81 ± 0.53	**100.0 ± 0.0**	**100.0 ± 0.0**
9	Oats	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	88.89 ± 10.03	83.33 ± 5.23	80.00 ± 6.72	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
10	Soybean-notill	92.25 ± 2.32	**98.78 ± 0.21**	81.42 ± 1.54	95.99 ± 1.09	98.01 ± 1.43	92.60 ± 2.29	96.50 ± 0.74	89.27 ± 1.1	96.72 ± 3.13
11	Soybean-mintill	87.05 ± 0.41	77.10 ± 0.25	86.76 ± 0.65	94.05 ± 0.22	95.17 ± 0.38	97.04 ± 0.61	93.74 ± 0.31	93.61 ± 0.25	**97.08 ± 0.16**
12	Soybean-clean	92.98 ± 3.22	**94.48 ± 2.33**	75.10 ± 5.2	85.10 ± 3.43	83.98 ± 3.21	91.57 ± 2.28	90.70 ± 1.99	88.39 ± 4.07	91.43 ± 2.45
13	Wheat	98.39 ± 0.78	**100.0 ± 0.0**	91.89 ± 3.13	99.41 ± 0.16	**100.0 ± 0.0**	91.28 ± 3.17	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**
14	Woods	93.67 ± 1.91	96.04 ± 2.31	70.66 ± 4.2	97.26 ± 0.32	98.52 ± 1.11	97.74 ± 1.16	98.94 ± 0.25	98.36 ± 0.42	**99.04 ± 0.35**
15	Bldgs-grass-trees-drives	87.07 ± 6.42	89.34 ± 7.21	48.64 ± 6.77	**98.11 ± 0.27**	93.05 ± 3.44	89.64 ± 2.31	96.87 ± 2.6	89.04 ± 3.54	95.07 ± 2.28
16	Stone-steel-towers	86.32 ± 3.01	91.76 ± 3.22	77.94 ± 7.1	93.67 ± 5.65	60.29 ± 6.09	72.46 ± 4.27	79.73 ± 6.44	**98.81 ± 1.13**	83.00 ± 4.85
	OA (%)	87.76 ± 0.66	87.40 ± 0.72	76.89 ± 1.22	92.28 ± 0.32	93.95 ± 0.49	93.18 ± 0.54	95.69 ± 0.29	93.46 ± 0.32	**96.74 ± 0.23**
	AA (%)	76.18 ± 1.78	78.85 ± 1.55	64.00 ± 2.44	83.82 ± 1.48	87.84 ± 1.33	83.16 ± 1.53	90.19 ± 1.21	90.59 ± 1.07	**95.88 ± 0.58**
	Kappa × 100	85.93 ± 0.53	85.52 ± 0.42	73.53 ± 1.28	91.17 ± 0.57	93.09 ± 0.67	92.23 ± 0.78	95.07 ± 0.49	92.55 ± 0.31	**96.28 ± 0.21**

Table 7. The classification results (%) of all compared methods on the UP dataset.

Class	Name	All Bands				PCA				Proposed
		FDSSC	SSRN	RSSAN	GSC-ViT	DFFN	HybridSN	SSFTT	AMS-M2ESL	
1	Asphalt	96.36 ± 1.14	92.26 ± 2.5	73.12 ± 2.2	**99.33 ± 0.83**	95.69 ± 2.32	75.39 ± 3.24	98.34 ± 0.38	96.71 ± 0.35	98.65 ± 0.12
2	Meadows	94.74 ± 0.71	98.35 ± 1.03	89.93 ± 2.19	97.15 ± 0.63	97.68 ± 0.65	93.24 ± 0.45	97.56 ± 0.22	98.69 ± 0.1	**99.19 ± 0.21**
3	Gravel	75.78 ± 4.91	91.97 ± 6.23	49.92 ± 4.12	64.34 ± 5.22	72.21 ± 5.49	50.80 ± 3.79	84.78 ± 2.42	83.97 ± 1.96	**96.30 ± 0.45**
4	Trees	98.80 ± 0.72	**99.03 ± 0.14**	97.78 ± 1.22	95.97 ± 1.33	74.22 ± 1.57	73.72 ± 0.89	91.92 ± 1.28	98.34 ± 0.76	97.53 ± 1.06
5	Painted metal sheets	99.70 ± 0.25	**99.85 ± 0.16**	93.75 ± 0.37	97.58 ± 0.35	99.70 ± 0.08	82.77 ± 1.21	92.37 ± 0.87	99.40 ± 0.17	99.33 ± 0.51
6	Bare soil	97.75 ± 1.19	93.90 ± 2.35	90.38 ± 1.38	98.72 ± 0.19	99.43 ± 0.13	90.63 ± 3.45	98.98 ± 0.22	**99.53 ± 0.21**	98.83 ± 0.13

Table 7. *Cont.*

Class	Name	All Bands				PCA				Proposed
		FDSSC	SSRN	RSSAN	GSC-ViT	DFFN	HybridSN	SSFTT	AMS-M2ESL	
7	Bitumen	90.42 ± 2.66	**99.80 ± 0.05**	36.13 ± 3.23	91.72 ± 3.54	94.47 ± 3.74	75.51 ± 2.32	97.38 ± 0.95	99.29 ± 0.01	96.36 ± 1.02
8	Self-blocking bricks	87.12 ± 5.03	**91.03 ± 3.56**	77.86 ± 4.88	80.42 ± 3.11	70.11 ± 5.29	70.13 ± 3.73	89.76 ± 1.43	88.92 ± 1.93	85.68 ± 3.21
9	Shadows	98.11 ± 1.21	99.04 ± 0.17	99.88 ± 0.06	94.56 ± 1.57	97.69 ± 1.03	81.57 ± 2.01	88.01 ± 1.02	**99.56 ± 0.29**	96.25 ± 0.75
	OA (%)	94.01 ± 0.62	96.02 ± 0.55	83.66 ± 0.68	93.76 ± 0.32	91.43 ± 0.87	84.02 ± 0.79	95.75 ± 0.42	96.98 ± 0.29	**97.40 ± 0.18**
	AA (%)	90.85 ± 1.12	93.52 ± 1.19	73.83 ± 1.62	92.33 ± 0.72	88.82 ± 1.32	64.22 ± 1.15	93.45 ± 0.64	94.89 ± 0.33	**95.97 ± 0.23**
	Kappa × 100	91.98 ± 0.66	94.72 ± 0.34	77.88 ± 0.87	91.72 ± 0.47	88.64 ± 0.52	78.42 ± 0.62	94.35 ± 0.31	95.98 ± 0.17	**96.54 ± 0.19**

Table 8. The classification results (%) of all compared methods on the YRE dataset.

Class	Name	All Bands				PCA				Proposed
		FDSSC	SSRN	RSSAN	GSC-ViT	DFFN	HybridSN	SSFTT	AMS-M2ESL	
1	Mudflat	95.00 ± 2.3	98.15 ± 1.32	93.72 ± 1.78	**98.67 ± 0.73**	66.98 ± 4.02	92.79 ± 1.13	92.69 ± 0.87	98.13 ± 0.95	94.93 ± 1.2
2	Bare soil	98.89 ± 0.44	**100.00 ± 0.0**	95.20 ± 0.83	**100.00 ± 0.0**	91.44 ± 1.02	83.37 ± 1.02	98.81 ± 0.97	**100.00 ± 0.0**	99.47 ± 0.44
3	Tamarix	**100.00 ± 0.0**	98.37 ± 0.66	68.36 ± 0.92	**100.00 ± 0.0**	96.82 ± 1.22	92.94 ± 2.01	83.82 ± 1.04	95.16 ± 1.87	**100.00 ± 0.0**
4	Suaeda salsa	98.05 ± 3.01	84.92 ± 2.03	68.96 ± 3.32	67.40 ± 3.17	80.47 ± 5.03	96.96 ± 0.89	98.08 ± 0.73	90.06 ± 2.21	**99.53 ± 0.61**
5	Spartina alterniflora	94.16 ± 0.2	95.02 ± 0.32	93.96 ± 0.49	**99.88 ± 0.3**	99.68 ± 0.33	94.27 ± 0.42	98.92 ± 1.13	96.69 ± 0.63	99.71 ± 0.21
6	Turbid water	97.12 ± 0.64	**100.00 ± 0.0**	99.38 ± 0.57	**100.00 ± 0.0**	99.99 ± 0.02	99.71 ± 0.22	**100.00 ± 0.0**	98.23 ± 0.53	98.37 ± 0.72
7	Tidal flat reed	99.95 ± 0.12	99.90 ± 0.17	96.40 ± 1.22	**100.00 ± 0.0**	94.98 ± 1.13	86.39 ± 5.01	99.51 ± 0.42	99.61 ± 0.23	96.20 ± 2.03
8	Clear water	77.68 ± 1.03	98.80 ± 0.13	81.15 ± 0.32	99.66 ± 0.41	**100.00 ± 0.0**	94.63 ± 2.32	95.66 ± 0.59	99.18 ± 0.27	94.57 ± 0.43
9	Bare lake beach	**100.00 ± 0.0**	99.80 ± 0.22	51.38 ± 4.21	**100.00 ± 0.0**	99.08 ± 0.55	85.99 ± 3.25	76.36 ± 2.32	**100.00 ± 0.0**	99.51 ± 0.37
	OA (%)	92.95 ± 0.64	97.19 ± 0.45	85.98 ± 0.87	97.30 ± 0.32	95.61 ± 0.47	93.74 ± 0.73	96.04 ± 0.63	97.57 ± 0.42	**98.08 ± 0.28**
	AA (%)	83.29 ± 0.92	93.95 ± 0.67	80.42 ± 0.98	94.27 ± 0.65	93.65 ± 0.76	91.11 ± 0.93	96.06 ± 0.89	96.86 ± 0.72	**98.06 ± 0.54**
	Kappa × 100	91.34 ± 0.33	96.57 ± 0.29	83.04 ± 0.43	96.70 ± 0.31	94.69 ± 0.32	92.39 ± 0.43	95.20 ± 0.41	97.05 ± 0.35	**97.67 ± 0.26**

In the experiments on the IP dataset, our proposed DMAN achieved the best classification accuracy, with OA, AA, and Kappa Coefficients of 96.74%, 95.88%, and 96.28%, respectively. The classification map achieved a near-truth map in general and outperformed other classification maps in most aspects. It is worth mentioning that none of the compared methods exceeded 90.6% AA in a very unbalanced sample environment with IP, and even if the OA was higher, this only means that the prediction accuracy of the class with a large proportion of samples is not bad. This suggests that the compared methods triggered an overfitting phenomenon in the face of feature extraction, learning some of the noise points of the classes into the features as well. The network structure with the worst classification results is RSSAN, which preemptively localizes the extracted features with an attention mechanism, but subsequently embeds an attention module for each layer that may disrupt the null spectrum feature semantic information with few samples. Also, with the introduction of the attention mechanism in the SSFTT, the OA decreased by only 1.05% compared to the proposed method because it more rationally arranges the information interaction and feature delivery. However, adding attention only to the shallow layer will blur the key features that had been highlighted, which are not easily learned by the linear layer, and will thus make the effect slightly lower. This is also reflected in the 5.69% decrease in AA. In addition, the FDSSC and SSRN also use full-band data classification, while the former uses dense connectivity compared to the latter's residual connectivity, and it was observed that the results of this study present a slight improvement, while the computational effort is substantially higher. This is the reason why we chose the residual structure in order to avoid gradient vanishing, which is simple in design and can effectively improve the classification accuracy and pass the information layer by layer towards deeper and more abstract features. As for the sample distribution and number of IPs, the DFFN and HybridSN control the complexity of the model very well and therefore perform well on this dataset. As shown in Figure 11, it is consistent with the accuracy. However, it is noteworthy that the seventh (grass-pasture-mowed) and ninth (oats) feature classes are the two classes that contain only one training sample and are the two classes that mainly contribute to the unbalanced classification environment. As can be seen, the misclassification area is significantly reduced compared to the other result maps while presenting purer color patches. Overall, our proposed method significantly reduces the presence of noise points when generating the classification maps and exhibits more accurate classification results with a relatively small number of misclassified pixels on the inter-class boundaries.

From the experiments on the UP dataset, according to Table 7, it can be seen that the DMAN method performs optimally. The OA value is 98.10%, which is 3.39%, 1.38%, 13.74%, 3.64%, 5.97%, 13.38%, 1.65%, and 0.42% higher than the other eight methods, respectively. While most of the methods have less difference between AA and OA, this is because the sample imbalance is lessened to a greater extent than in the IP dataset, with a Kappa Coefficient of 96.54%. The classification accuracy of all the categories of the proposed method reaches more than 96%, except for the eighth category. This is due to the lack of distinctiveness of the category features of self-blocking bricks, reflecting the fact that the DMAN is more difficult for extracting features with high discriminative degrees. Among the compared methods, AMS-M2ESL has the best classification effect, with an OA of 96.98%. This is due to the fact that the UP dataset samples are meticulously etched and have a large number of edge pixels, while the original article uses a smaller patch size that does not over-smooth on this data and the distance covariance-based descriptor in the model effectively explores the linear and nonlinear interdependencies in the spectral domain. On the contrary, in the HybridSN and DFFN, which perform well in the IP dataset, the accuracy of the HybridSN decreases significantly. Since the patches of both methods are relatively large, it can be inferred that the multi-scale fusion strategy that we adopted is effective. In addition, the multiple scales in the DFFN include the merging of features at different levels. Based on Figure 12, it can be visualized that the RSSAN results show many misassigned pixels and a low classification accuracy. The DFFN, HybridSN, and SSFTT have intra-class

pixel patch prediction errors, such as class VII (bare soil). Overall, the classification map of our proposed DMAN method is closest to the truth map.

As shown in Table 3, the YRE dataset has only 79 training samples at a 0.1% data partitioning ratio; yet, the majority of the methods achieved better results than before. This is due to the balanced distribution of the number of categories while artificially labeling the purer image elements. Our method still achieved the best results, with OA, AA, and Kappa Coefficients of 98.08%, 98.06%, and 97.67%, respectively, and a classification accuracy of 94% or more for all categories. It can be seen that GSC-ViT achieved the highest accuracy for most of the categories, but for category 4 (Suaeda salsa), there are a large number of pixels that are misclassified as this, which indicates that the model is lightweight but lacks sensitivity to feature similarity class distinction. For AMS-M2ESL, which is also a transformer backbone network, this reflects the importance of considering relationships between pixels within a block. Based on Figure 13, the misclassification and omission of the fifth class (*Spartina alterniflora*) is more obvious, which is because it possesses two major classes of features, and random samples may not portray them in a balanced and adequate way. Among all the generated classification maps, the DMAN demonstrated superior results, with significant improvements in the control of noise points. This is reflected not only in the overall accuracy of the results but also in the subtleties that show a more precise and robust classification.

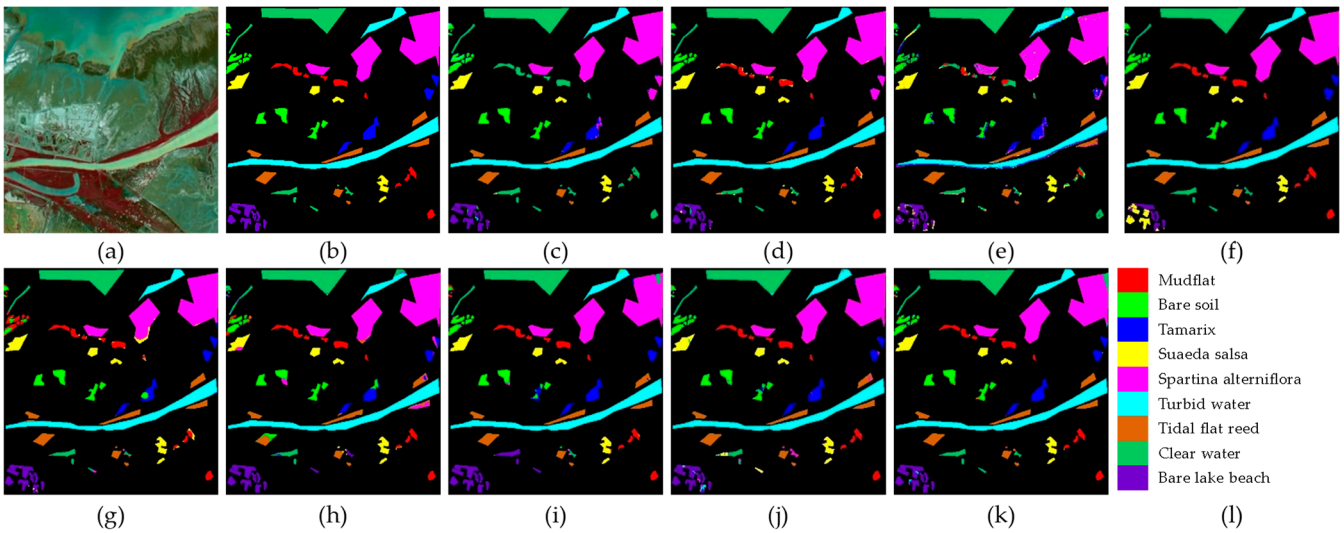


Figure 13. Classification maps of different methods on the YRE dataset. (**a**) False-color; (**b**) ground truth map; (**c**) FDSSC; (**d**) SSRN; (**e**) RSSAN; (**f**) GSC-ViT; (**g**) DFFN; (**h**) HybridSN; (**i**) SSFTT; (**j**) AMS-M2ESL; (**k**) DMAN; and (**l**) color bar.

Comparing the results of the three datasets, it can be inferred that the design of the RSSAN is not suitable for small-sample problems. Frequent per-layer attention block embedding may be the main reason for the poor classification results. In contrast, the performance of the SSFTT was relatively stable, and satisfactory results were obtained. So, a reasonable attention setting can ensure the generalization of the model.

3.4. Ablation Study

To further investigate the real contribution of each part of the proposed model, we conducted ablation experiments using three datasets, where the dataset division ratios were consistent with those described in the previous section, and the parameters were kept at their original settings. We investigated the effectiveness of the proposed three attentions and proposed a total of seven combinations of S0 to S6, as shown in Table 9, where the DMANs containing all the attentions are not described here without repeating the experiments.

Table 9. The way the attention modules are combined in the model.

Name	NAM	ESA	PPA
S0	without attention		
S1	√		
S2		√	
S3			√
S4	√	√	
S5	√		√
S6		√	√

From the results in Figure 14, it can be seen that most of the classification methods embedded in a single spectral, spatial, or hybrid attention module are inferior to the classification methods of any two of the attention combinations, and thus, all three attention modules positively contribute to each other. However, there are exceptions in the YRE dataset, where the PPA slightly outperforms the model structure of the combination of the NAM and ESA, with a 0.38% improvement in OA. First, because the YRE dataset has low spatial resolution and rich spectral information, the multi-scale pooling and post-splicing convolutional capture operation of the PPA can integrate the spatial context information and band dependence, which is more in line with the characteristics of the YRE dataset. Second, the number of parameters contained in the PPA is more than the other two attentions, which can effectively improve the expression ability of the network to better recognize the features.

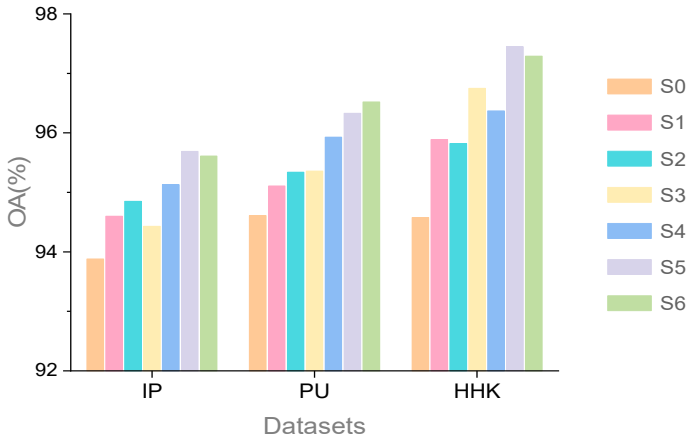


Figure 14. The OAs of various combinations in three datasets.

Each is viewed separately; first, for the comparison among the methods combining the two attentions, the inclusion of the PPA tends to have a greater boost, reflecting the importance of the integration process of the two branches, and the direct access to global pooling and full connectivity may disturb the null spectral features. The three attentions alone are difficult to compare, with different experimental effects in different datasets, but the improvement over the backbone model without attention is still significant. Of course, based on the aforementioned comparison experiments, the full DMAN will have better results.

3.5. Analysis of the Hybrid Strategy in the HRPM Structure

Random projection theory reveals the feasibility of dimensionality reduction by random matrices, on which the RPNet utilizes the potential value of the original data to construct convolution for dimensionality reduction. We propose the HRPM as an RPNet that includes a selectable convolution method, which was initially designed to retain categorical information more effectively under small-sample conditions. To demonstrate the effectiveness of this module as well as to study the impact of the structural composition

of the HRPM in the presence of fewer principal components, we chose to conduct this experiment on an IP dataset containing 5% of the training samples. The different structures of the HRPM are explained as follows. Fully considering the amount of computation and time cost, we set the depth at three layers, and a total of six different structures were involved in the experiment. The numbers 2 and 3 represent 2D convolution and 3D convolution, respectively, and the numbers are arranged to represent the order of each layer in the HRPM.

In addition, in order to ensure that the HRPM positively fuses the spectral and spatial information during dimensionality reduction, it is stipulated that the convolutional approach should not appear in turn because it would disrupt the feature information and make the original data generate a lot of noise, which would impede the process of model learning after our experiments.

According to the experimental results shown in Table 10, the 2D convolution alone, which is the convolution method of the RPNet, is the least effective, with the OA being 0.68% to 1.14% lower than the other methods, illustrating the importance of the additional consideration of spectral features during the dimensionality reduction process. The 2D convolution essentially combines the full spectral and spatial information directly into one channel after another, which would be redundant while also being sensitive to the target. The 2D convolution is essentially a direct integration of all the spectral and spatial information into one channel after another, which is redundant, and at the same time, insensitive to the neighboring bands in the vicinity of the target band, focusing only on the global features. The addition of 3D convolution allows for more targeted feature retention and enables multi-level local and global learning. In general, the type of convolution performed first does not have a significant impact on OA, but there is a significant difference between the two in terms of AA. The 3D convolution has the effect of refining the features in spectral dimensions; so, this approach naturally retains more categorical information than the other convolution.

Table 10. Impact of the HRPM structure on categorization effects on IP.

IP (5%)	**Structure of the HRPM**					
	222	**223**	**233**	**333**	**332**	**322**
OA (%)	94.42	95.17	95.52	95.34	95.56	95.10
AA (%)	87.22	89.65	87.03	92.78	91.39	90.92
Kappa × 100	93.62	94.48	94.89	94.68	94.93	94.40

3.6. Analysis of the Model Complexity

In this section, to evaluate the complexity of the proposed DMAN concerning other comparative methods, we counted the number of parameters as well as the average training and testing times for the three runs. To maintain fairness, we used the Adam optimizer and set the learning rate, batch size, and epoch to 1×10^{-3}, 32, and 100, respectively, with the same configuration for all models. The experimental results are shown in Table 11, where the optimal results have been bolded. Overall, it can be seen that the number of parameters of the DMAN is very small, mainly in two datasets, thanks to the light weight of the three attention modules and the settings of the convolutional parameters. In addition, since our model includes multi-scale as well as double-branching features, this directly affects the forward and backward propagation process, leading to longer training and testing times while still outperforming most methods. The increase in computational cost is acceptable considering the improved classification accuracy. It is worth mentioning that the SSFTT also has a very high training and prediction efficiency when the classification results of the three datasets perform well, which may be because its Feature Tokenizer plays a key role in capturing spatial–spectral features and abandons the traditional deep learning method of simply stacking convolutional layers, which is the area we want to improve in the future.

Table 11. Model complexity comparison.

Method	Datasets								
	IP			UP			YRE		
	Training Time (s)	Testing Time (s)	Params (M)	Training Time (s)	Testing Time (s)	Params (M)	Training Time (s)	Testing Time (s)	Params (M)
FDSSC	53.94	5.35	2.39	21.94	42.86	1.23	23.18	308.15	3.35
SSRN	30.08	3.10	0.36	12.07	19.13	0.22	13.23	150.27	0.49
RSSAN	27.40	3.88	0.17	11.89	26.21	0.09	8.21	102.32	0.17
GSC-ViT	42.86	5.82	0.56	18.55	44.99	**0.08**	10.89	163.06	0.12
DFFN	34.38	3.12	0.38	17.37	31.31	0.38	10.24	81.42	0.38
HybridSN	25.64	2.72	5.12	13.89	27.93	5.12	8.04	59.62	5.12
SSFTT	**13.84**	**1.22**	0.15	**9.17**	**11.93**	0.15	**5.31**	27.68	0.15
AMS-M2ESL	53.46	13.56	1.0	22.36	133.7	0.99	12.37	402.48	1.33
DMAN	18.42	2.21	**0.05**	10.83	20.78	0.10	6.64	**26.35**	**0.05**

3.7. Analysis of Model Noise Robustness

To further assess the stability of the model in different sample environments, we selected Gaussian noise and pretzel noise to verify the noise robustness of the model on three datasets. Each of the two categories in the pretzel noise accounted for half of the added proportion. In this experiment, the data were pre-normalized, and then, different levels of noise were added. We observed a change in classification accuracy; when the result showed a significant decrease, we terminated the experiment. The result is shown in Figure 15.

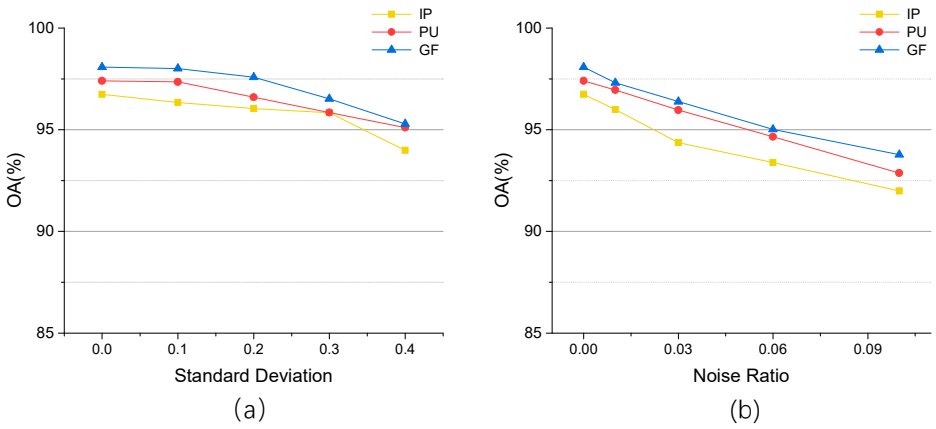


Figure 15. (**a**) Effect of Gaussian noise on the model. (**b**) Effect of pretzel noise on the model.

In (a), it can be seen that the classification effect enters a slower decreasing trend at 0.3 and beyond as the variance increases. In (b), a rapid decrease in accuracy can be observed as the noise scale increases. When the noise reaches a scale of 10%, the accuracy drops close to 5 percentage points, which is more variable than Gaussian noise, due to the fact that the placement of zeros and ones interferes with the model's ability to capture features to a greater extent. Overall, the magnitude of change in OA is insensitive to noise, proving the superior robustness of the proposed model.

4. Discussion

The proposed methodology was subjected to parametric analysis, comparative experiments, and ablation experiments, respectively, to determine the advantages and significant performance of the DMAN. The above results and phenomena are discussed next in this section.

It is well known that the choice of window size is undoubtedly one of the key factors affecting the performance of the model, and almost all analytical experiments on this

parameter are prevalent in the application of deep learning methods for HIS categorization. Of course, this is the case when using patch-based method inputs. It can be observed that the model's sensitivity to the window size is different under different datasets, but all of them are small windows. The SSRN and SSFTT, which achieved optimal results in the comparison experiments, also use smaller windows. This is because the small-sample condition focuses more on the category quality of the patch block, which determines the classification effect. The small window tends to contain more information about this category and introduces less noise.

For the model itself, this paper divides the compared methods into two categories: one with full-band input and one with input after applying PCA dimensionality reduction. Both methods have advantages and disadvantages in terms of accuracy. Since this is closely related to whether the model structure design is suitable for small-sample problems, it was not possible to assess the classification effect based on the way the data were utilized. However, dimensionality reduction tends to be more efficient in terms of running time; so, in the future, small-sample problems may move towards using dimensionality reduction methods for preprocessing. In addition, most deep learning methods do not utilize very few bands or a few principal components after PCA for classification. Our proposed DMAN, in combination with the HRPM, can compensate for this shortcoming. Since it additionally considers the randomness of the spectral dimension, the accuracy does not degrade too much in the case of a small number of samples.

From the classification results, it is clear that the DMAN achieves optimal results in both balanced and unbalanced sample environments. The small-sample problem does have limitations that reduce the accuracy, but the ablation experiments show that the attention mechanism is indeed an important way to make the model approach the upper limit. This is also because other fields have enhanced the maturity of the attention mechanism, leading to further breakthroughs in the small-sample problem. In addition, the results also show that the DMAN is expected to overcome the challenges associated with dispersed sample distribution and misclassification at class boundaries by reducing the noise level and rationalizing the use of pixels outside the central class within the window.

In summary, this study emphasizes the importance of continued in-depth research in this area, focusing not only on improving classification accuracy, but also on overcoming the serious challenges faced in small-sample classification, unbalanced sample distribution, and misclassification of category boundaries. Overcoming these problems is expected to provide useful guidance for the optimization and innovation of HSI classification methods in the future.

5. Conclusions

This paper presents a two-branch network model based on embedded multivariate attention and multi-scale fusion for hyperspectral image classification. The proposed model effectively addresses the challenge of fully utilizing the spatial and spectral information of hyperspectral images to extract key features in scenarios with limited sample sizes. The method applies to cases where few bands are retained after dimensionality reduction, where it will first be selected to enter the HRPM, which, after hybrid convolution and feature stacking, makes it possible to retain most of the feature information while increasing the inter-class separation. When the available bands are sufficient after dimensionality reduction, they are directly fed into the classification network. After two branches of spectral and spatial features are extracted separately with embedded attention, the attention is pooled through pyramid pooling, fusing abstract features and further emphasizing the key features, which enhances the network's characterization capability. The best classification results are demonstrated on two publicly available datasets and one of our labeled datasets, ultimately validating the significance of combining multi-scale feature extraction and attention in this model for small-sample classification.

In future work, more advanced and efficient methods need to be explored in the composition of deep learning models, aiming to enhance accuracy while maintaining

efficiency. For the small-sample problem, we plan to combine the sample expansion method, embedded in the network, to break through the limitations of small sample sizes for hyperspectral image classification.

Author Contributions: Conceptualization, Y.C. and X.W.; methodology, Y.C. and X.W.; software, Y.C. and S.Z.; validation, X.S. and Y.H.; formal analysis, X.W.; investigation, J.W.; resources, J.Z.; data curation, Y.C.; writing—original draft preparation, Y.C.; writing—review and editing, X.W.; visualization, Y.C. and J.W.; supervision, X.W.; project administration, X.S.; and funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by the Finance Science and Technology Project of Hainan Province (No. ZDYF2021SHFZ063), the Shandong Key Research and Development Project (No. 2018GNC110025, No. ZR2020QF067), the National Natural Science Foundation of China (No.42301380, No.42106179), the Qingdao Natural Science Foundation Grant (No.23-2-1-64-zyyd-jch), and the Science and Technology Support Plan for Youth Innovation of Colleges and Universities of Shandong Province of China (No.2023KJ232).

Data Availability Statement: In this paper, two public hyperspectral datasets were selected. The Indian Pines dataset and the Pavia University dataset are available at https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 15 September 2023).

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

References

1. Tan, K.; Wu, F.; Du, Q.; Du, P.; Chen, Y. A Parallel Gaussian–Bernoulli Restricted Boltzmann Machine for Mining Area Classification with Hyperspectral Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 627–636. [CrossRef]
2. Pascucci, S.; Pignatti, S.; Casa, R.; Darvishzadeh, R.; Huang, W. Special Issue "Hyperspectral Remote Sensing of Agriculture and Vegetation". *Remote Sens.* **2020**, *12*, 3665. [CrossRef]
3. Kuras, A.; Brell, M.; Rizzi, J.; Burud, I. Hyperspectral and Lidar Data Applied to the Urban Land Cover Machine Learning and Neural-Network-Based Classification: A Review. *Remote Sens.* **2021**, *13*, 3393. [CrossRef]
4. Audebert, N.; Le Saux, B.; Lefevre, S. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [CrossRef]
5. Li, W.; Chen, H.; Liu, Q.; Liu, H.; Wang, Y.; Gui, G. Attention Mechanism and Depthwise Separable Convolution Aided 3DCNN for Hyperspectral Remote Sensing Image Classification. *Remote Sens.* **2022**, *14*, 2215. [CrossRef]
6. Melgani, F.; Bruzzone, L. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
7. Ham, J.; Chen, Y.; Crawford, M.M.; Ghosh, J. Investigation of the Random Forest Framework for Classification of Hyperspectral Data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
8. Duan, Y.; Huang, H.; Tang, Y. Local Constraint-Based Sparse Manifold Hypergraph Learning for Dimensionality Reduction of Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 613–628. [CrossRef]
9. Luo, F.; Zhang, L.; Zhou, X.; Guo, T.; Cheng, Y.; Yin, T. Sparse-Adaptive Hypergraph Discriminant Analysis for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1082–1086. [CrossRef]
10. Duan, Y.; Huang, H.; Li, Z.; Tang, Y. Local Manifold-Based Sparse Discriminant Learning for Feature Extraction of Hyperspectral Image. *IEEE Trans. Cybern.* **2021**, *51*, 4021–4034. [CrossRef] [PubMed]
11. Jia, S.; Jiang, S.; Lin, Z.; Li, N.; Xu, M.; Yu, S. A Survey: Deep Learning for Hyperspectral Image Classification with Few Labeled Samples. *Neurocomputing* **2021**, *448*, 179–204. [CrossRef]
12. Dinç, S.; Aygün, R.S. Evaluation of Hyperspectral Image Classification Using Random Forest and Fukunaga-Koontz Transform. In *Machine Learning and Data Mining in Pattern Recognition*; Perner, P., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7988, pp. 234–245.
13. Kang, X.; Xiang, X.; Li, S.; Benediktsson, J.A. PCA-Based Edge-Preserving Features for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7140–7151. [CrossRef]
14. Mughees, A.; Tao, L. Multiple Deep-Belief-Network-Based Spectral-Spatial Classification of Hyperspectral Images. *Tsinghua Sci. Technol.* **2019**, *24*, 183–194. [CrossRef]
15. Chen, C.; Ma, Y.; Ren, G. Hyperspectral Classification Using Deep Belief Networks Based on Conjugate Gradient Update and Pixel-Centric Spectral Block Features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 4060–4069. [CrossRef]
16. Mei, X.; Pan, E.; Ma, Y.; Dai, X.; Huang, J.; Fan, F.; Du, Q.; Zheng, H.; Ma, J. Spectral-Spatial Attention Networks for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 963. [CrossRef]
17. Zhou, W.; Kamata, S.; Luo, Z.; Wang, H. Multiscanning Strategy-Based Recurrent Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5521018. [CrossRef]

18. Liang, L.; Zhang, S.; Li, J. Multiscale DenseNet Meets with Bi-RNN for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 5401–5415. [CrossRef]
19. 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]
20. 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]
21. Feng, J.; Chen, J.; Liu, L.; Cao, X.; Zhang, X.; Jiao, L.; Yu, T. CNN-Based Multilayer Spatial-Spectral Feature Fusion and Sample Augmentation with Local and Nonlocal Constraints for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1299–1313. [CrossRef]
22. Zhang, M.; Li, W.; Du, Q.; Gao, L.; Zhang, B. Feature Extraction for Classification of Hyperspectral and LiDAR Data Using Patch-to-Patch CNN. *IEEE Trans. Cybern.* **2020**, *50*, 100–111. [CrossRef] [PubMed]
23. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 277–281. [CrossRef]
24. Paoletti, M.E.; Mario Haut, J.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.J.; Pla, F. Deep Pyramidal Residual Networks for Spectral-Spatial Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 740–754. [CrossRef]
25. 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]
26. 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]
27. Gong, H.; Li, Q.; Li, C.; Dai, H.; He, Z.; Wang, W.; Li, H.; Han, F.; Tuniyazi, A.; Mu, T. Multiscale Information Fusion for Hyperspectral Image Classification Based on Hybrid 2D-3D CNN. *Remote Sens.* **2021**, *13*, 2268. [CrossRef]
28. Zhang, X.; Wang, Y.; Zhang, N.; Xu, D.; Luo, H.; Chen, B.; Ben, G. Spectral–Spatial Fractal Residual Convolutional Neural Network with Data Balance Augmentation for Hyperspectral Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10473–10487. [CrossRef]
29. Zahisham, Z.; Lim, K.M.; Koo, V.C.; Chan, Y.K.; Lee, C.P. 2SRS: Two-Stream Residual Separable Convolution Neural Network for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5501505. [CrossRef]
30. Wang, P.; Zheng, C.; Liu, S. Superpixel-Guided Multifeature Tensor for Hyperspectral Image Classification with Limited Training Samples. *Opt. Laser Technol.* **2023**, *159*, 109020. [CrossRef]
31. Zhang, S.; Zhang, J.; Wang, X.; Wang, J.; Wu, Z. ELS2T: Efficient Lightweight Spectral-Spatial Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5518416. [CrossRef]
32. Lin, C.; Wang, T.; Dong, S.; Zhang, Q.; Yang, Z.; Gao, F. Hybrid Convolutional Network Combining 3D Depthwise Separable Convolution and Receptive Field Control for Hyperspectral Image Classification. *Electronics* **2022**, *11*, 3992. [CrossRef]
33. Lv, Z.; Dong, X.-M.; Peng, J.; Sun, W. ESSINet: Efficient Spatial–Spectral Interaction Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5525715. [CrossRef]
34. Zhao, Z.; Xu, X.; Li, J.; Li, S.; Plaza, A. Gabor-Modulated Grouped Separable Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5518817. [CrossRef]
35. Liang, M.; Wang, H.; Yu, X.; Meng, Z.; Yi, J.; Jiao, L. Lightweight Multilevel Feature Fusion Network for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 79. [CrossRef]
36. Wang, J.; Huang, R.; Guo, S.; Li, L.; Pei, Z.; Liu, B. HyperLiteNet: Extremely Lightweight Non-Deep Parallel Network for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 866. [CrossRef]
37. Hu, W.-S.; Li, H.-C.; Deng, Y.-J.; Sun, X.; Du, Q.; Plaza, A. Lightweight Tensor Attention-Driven ConvLSTM Neural Network for Hyperspectral Image Classification. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 734–745. [CrossRef]
38. 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*, 5518615. [CrossRef]
39. Zhao, Z.; Xu, X.; Li, S.; Plaza, A. Hyperspectral Image Classification Using Groupwise Separable Convolutional Vision Transformer Network. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5511817. [CrossRef]
40. Li, M.; Li, W.; Liu, Y.; Huang, Y.; Yang, G. Adaptive Mask Sampling and Manifold to Euclidean Subspace Learning with Distance Covariance Representation for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5508518. [CrossRef]
41. Xiang, J.; Wei, C.; Wang, M.; Teng, L. End-to-End Multilevel Hybrid Attention Framework for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5511305. [CrossRef]
42. Huang, W.; Zhao, Z.; Sun, L.; Ju, M. Dual-Branch Attention-Assisted CNN for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 6158. [CrossRef]
43. 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]
44. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.-I. Feedback Attention-Based Dense CNN for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5501916. [CrossRef]
45. Yang, K.; Sun, H.; Zou, C.; Lu, X. Cross-Attention Spectral–Spatial Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5518714. [CrossRef]

46. Li, S.; Luo, X.; Wang, Q.; Li, L.; Yin, J. H2AN: Hierarchical Homogeneity-Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5509816. [CrossRef]
47. Bai, J.; Wen, Z.; Xiao, Z.; Ye, F.; Zhu, Y.; Alazab, M.; Jiao, L. Hyperspectral Image Classification Based on Multibranch Attention Transformer Networks. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5535317. [CrossRef]
48. Li, M.; Liu, Y.; Xue, G.; Huang, Y.; Yang, G. Exploring the Relationship Between Center and Neighborhoods: Central Vector Oriented Self-Similarity Network for Hyperspectral Image Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 1979–1993. [CrossRef]
49. Xu, Y.; Du, B.; Zhang, F.; Zhang, L. Hyperspectral Image Classification via a Random Patches Network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *142*, 344–357. [CrossRef]
50. Cheng, C.; Li, H.; Peng, J.; Cui, W.; Zhang, L. Hyperspectral Image Classification Via Spectral-Spatial Random Patches Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4753–4764. [CrossRef]
51. Ma, Y.; Liu, Z.; Chen, C.L.P. Multiscale Random Convolution Broad Learning System for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5503605. [CrossRef]
52. 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]
53. Alkhatib, M.Q.; Al-Saad, M.; Aburaed, N.; Almansoori, S.; Zabalza, J.; Marshall, S.; Al-Ahmad, H. Tri-CNN: A Three Branch Model for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 316. [CrossRef]
54. Zhang, E.; Zhang, J.; Bai, J.; Bian, J.; Fang, S.; Zhan, T.; Feng, M. Attention-Embedded Triple-Fusion Branch CNN for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 2150. [CrossRef]
55. Liu, Y.; Shao, Z.; Teng, Y.; Hoffmann, N. NAM: Normalization-based Attention Module. *arXiv* **2021**, arXiv:2111.12419.