remote sensing

MDPI

Article

Double-Branch Network with Pyramidal Convolution and Iterative Attention for Hyperspectral Image Classification

Hao Shi [1], Guo Cao [1,*], Zixian Ge [1], Youqiang Zhang [2] and Peng Fu [1]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; hao1227@njust.edu.cn (H.S.); gezixian727@njust.edu.cn (Z.G.); fupeng@njust.edu.cn (P.F.)
[2] School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; zhangyq@njupt.edu.cn
* Correspondence: caoguo@njust.edu.cn

Abstract: Deep-learning methods, especially convolutional neural networks (CNN), have become the first choice for hyperspectral image (HSI) classification to date. It is a common procedure that small cubes are cropped from hyperspectral images and then fed into CNNs. However, standard CNNs find it difficult to extract discriminative spectral–spatial features. How to obtain finer spectral–spatial features to improve the classification performance is now a hot topic of research. In this regard, the attention mechanism, which has achieved excellent performance in other computer vision, holds the exciting prospect. In this paper, we propose a double-branch network consisting of a novel convolution named pyramidal convolution (PyConv) and an iterative attention mechanism. Each branch concentrates on exploiting spectral or spatial features with different PyConvs, supplemented by the attention module for refining the feature map. Experimental results demonstrate that our model can yield competitive performance compared to other state-of-the-art models.

Keywords: hyperspectral; deep learning; convolutional neural network; attention mechanism

check for updates

Citation: Shi, H.; Cao, G.; Ge, Z.; Zhang, Y.; Fu, P. Double-Branch Network with Pyramidal Convolution and Iterative Attention for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 1403. https://doi.org/10.3390/rs13071403

Academic Editor: Pedro Melo-Pinto

Received: 19 March 2021
Accepted: 2 April 2021
Published: 6 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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 remote sensing, containing a rich triad of spatial, radiometric and spectral information, is a frontier area of remote-sensing technology. The hyperspectral remote sensor with remarkable features of high spectral resolution (5~10 nm) and wide spectral range (0.4 μm~2.5 μm) can use dozens or even hundreds of narrow spectral bands to collect information. All the bands can be arranged together to form a continuous and complete spectral curve, which covers the full range of electromagnetic radiation from the visible to the near-infrared wavelength. Hyperspectral image (HSI) implements the effective integration of spatial and spectral information of remote-sensing data and thus addresses important remote-sensing applications, e.g., agriculture [1], environmental monitoring [2], and physics [3].

Traditional spectral-based methods such as k-nearest neighbors [4], multinomial logistic regression (MLR) [5], and support vector machines (SVM) [6], tend to treat the raw pixels directly as input. However, given the large number of spectral bands in HSI, the classifier must deal with these features in a high-dimensional space. Due to the numerous spectral bands in HSI, the classifier is confronted with high-dimensional features and the limited samples makes it difficult to train a classifier with high accuracy. This problem is known as the curse of dimensionality or the Hughes phenomenon. To tackle this problem, dimensionality reduction such as feature selection [7] or feature extraction [8] is a common tactic. Moreover, considering that neighboring pixels probably belong to the same class, another line of research aims at focusing on spatial information. Gu et al. [9] and Fang et al. [10] used SVM as a classifier with a multiple kernel learning strategy to process the HSI data and obtained the desired results. In [11], the original HSI data was fused with multi-scale superpixel segmentation maps and then fed into SVM for processing. Methods

of this sort essentially implement feature engineering with the help of spectral–spatial information on the HSI and then create a classification map.

However, the aforementioned approaches can be considered to be traditional feature engineering, which means that the performance depends on the handcrafted features. Furthermore, as these methods belong to shallow models, the generated features should also be regarded as shallow features, which are unable to capture the essential characteristics of the observed object and therefore tend to underperform in sophisticated scenarios [12].

Due to the impressive ability to automatically extract non-linear hierarchical features, deep learning (DL) has gradually supplanted numerous traditional algorithms in recent years, gaining an overwhelming advantage in many computer vision tasks including objection detection [13], semantic segmentation [14], and image generation [15]. Naturally, HSI classification, as a typical classification task, is constantly benefiting from the state-of-the-art deep-learning techniques. Several deep-learning-based methods have been proposed for HSI classification. In [16], Chen et al. introduced a stacked autoencoder (SAE) to extract abundant features for HSI classification. Zhao et al. [17] also leveraged a stacked sparse autoencoder to derive hierarchical more abstract and deeper features from spectral vectors, spatial vectors and spectral–spatial vectors. Li et al. [18] investigated deep belief networks (DBNs) for spectral–spatial features extraction, improving the accuracy of HSI classification. Zhong et al. [19] improved prior diversity during pre-training and fine-tuning of the DBN model, resulting in improved HSI classification performance.

Among the DL-based methods, the convolutional neural network (CNN) [20] is the predominant formulation for extracting spectral–spatial features by virtue of its local perception and parameter sharing characteristics. Mei et al. [21] proposed a CNN model incorporating spectral features with spatial context by computing the mean of the pixel neighborhood and the mean and standard deviation of each spectral band in that neighborhood. Similarly, Lee et al. [22] presented a contextual deepCNN (CDCNN) for feature extraction. Moreover, Zhao and Du [23] combined a spatial feature extraction process with a spectral feature extraction process based on the CNN model. Concretely, the local discriminative embedding is performed first, followed by stacked features and classification. Although these methods employ different techniques to extract spectral–spatial information separately apart from CNN, they do not fully leverage the joint spectral–spatial information. In view of the fact that hyperspectral data can be represented in a 3D cube format, 3D convolution in spectral and spatial dimensions can naturally be a 'silver bullet' in simultaneously extracting the spectral–spatial features of HSI [24,25]. Furthermore, inspired by the deeper network such as residual network (ResNet) [26] and the dense convolutional network (DenseNet) [27], Zhang et al. [28] proposed a spectral–spatial residual network (SSRN), which stacks the spectral and spatial residual blocks consecutively. Wang et al. [29] employed DenseNet in their fast dense spectral–spatial convolution (FDSSC) algorithm.

On the other hand, it is worth noting that different spectral bands and different spatial patches in the HSI cube may make different contributions to feature extraction. Accordingly, there has been a surge of interest in the attention mechanism [30–32]. By focusing on important features and suppressing unnecessary features, attention mechanisms can augment model sensitivity to informative spectral bands and spatial positions. Thus, Ma et al. [33] designed a double-branch multi-attention mechanism network (DBMA), obtaining desirable results. Furthermore, based on DBMA and dual-attention network (DANet) [34], Li et al. [35] proposed the double-branch dual-attention mechanism network (DBDA) for HSI classification.

In this paper, inspired by these advanced techniques, we propose an attention-aided spectral–spatial CNN model for hyperspectral image classification. Instead of following the traditional approach of using standard 3D convolution to extract features from HSI, we apply the pyramidal convolution which can extract hierarchical features. Furthermore, a latest attention mechanism is adopted to refine the features for better classification. Our new deep model is composed of two branches, which extract spectral and spatial features,

respectively. In each branch, pyramidal convolution is introduced to exploit abundant features at different scales. Then, a novel iterative attention mechanism is applied to refine the feature maps. By concatenating or using weighted addition, we fuse the double-branch features. Finally, the fused spectral–spatial features are fed into the fully connected layer to obtain classification results with the SoftMax function. The main contributions of this article are as follows:

(1) A new double-branch model based on pyramidal 3D convolution is proposed for HSI classification. Two branches can separately extract spatial features and spectral features efficiently.
(2) A new iterative attention mechanism, expectation-maximization attention (EMA), is introduced to HSI classification. It can refine the feature map by highlighting relevant bands or pixels and suppressing the interference of irrelevant bands or pixels.
(3) Some effective techniques, such as the new activation function Mish, dynamically varying learning rates and early stopping, are applied in the proposed model and satisfactory results are obtained.

The rest of this paper is organized as follows: In Section 2, we briefly describe the related work. Our proposed architecture is described in detail in Section 3. In Sections 4 and 5, we conduct several experiments and analyze the experimental results. Finally, conclusions and future work are presented in Section 6.

2. Related Work

In this section, we briefly review several highly correlated techniques before introducing the proposed HSI classification framework, which is pyramidal convolution (PyConv), ResNet and DenseNet, and attention mechanism.

2.1. A Multi-Scale 3D Convolution—PyConv

As mentioned in the preceding section, the 3D-CNN-based approach has carved out a niche for itself in HSI classification. Considering that the spectral dimension of HSI is abundant with detailed information of land covers, 3D convolution is an appealing operation in exploiting the spatial and spectral information in HSI for classification.

Based on the standard 3D convolution [36], several offshoots have evolved [37–39]. Among them, the multi-scale 3D convolution is of interest in this paper. In [40], a multi-scale 3D convolution named pyramidal convolution (PyConv) was proposed, illustrated in Figure 1. Using a pyramid with different types of kernels, PyConv can process the input feature maps FM_i at different scales, resulting in a series of output feature maps FM_o with complementary information. Generally, PyConv is a hierarchical structure that stacks 3D convolution kernels with different sizes. At each level of PyConv, the spatial size of the kernels varies, increasing from the bottom of the pyramid to the top. As the spatial size increases, the depth of the kernel simultaneously decreases. Consequently, as shown in Figure 1, this leads to two pyramids, facing opposite directions. One pyramid is wide at the bottom and narrow at the top in terms of the depth of the kernel, and the other inverted pyramid is narrow at the bottom and wide at the top in terms of the spatial size of the kernel. This pyramidal structure provides a pool of combinations in which there can be different types and sizes of kernels. Thanks to this, the network can possess the ability to acquire complementary information since kernels with smaller receptive fields focus on small objects and details while kernels with larger receptive fields can concentrate on larger objects and contextual information.

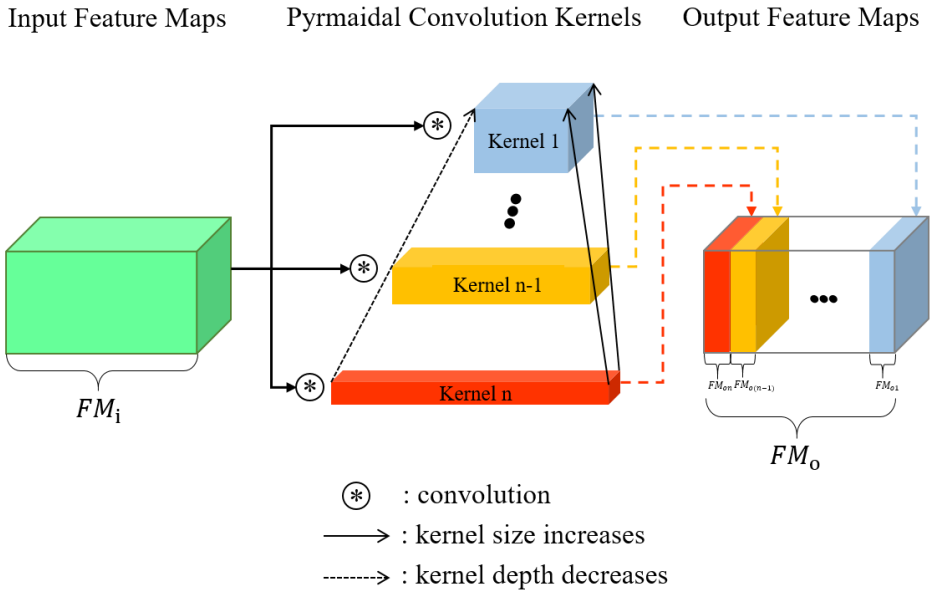


Figure 1. Pyramidal convolution (PyConv).

2.2. ResNet and DenseNet

Deep networks can lead to better performance, but optimizing deep networks is very difficult. To combat this dilemma, ResNet and DenseNet are powerful tools.

Inspired by residual representations in image recognition, ResNet introduces shortcut connections to the network. As shown in Figure 2a H, denotes hidden layers, including convolution layers, activation function layers, and batch normalization (BN) layers. In the original text of ResNet, shortcut connections simply perform identity mapping, enabling information or gradient to pass directly without travelling through intermediate layers. To mathematically formalize residual learning, identity mapping by shortcuts is integrated into a basic block in ResNet, which can be defined as:

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{1}$$

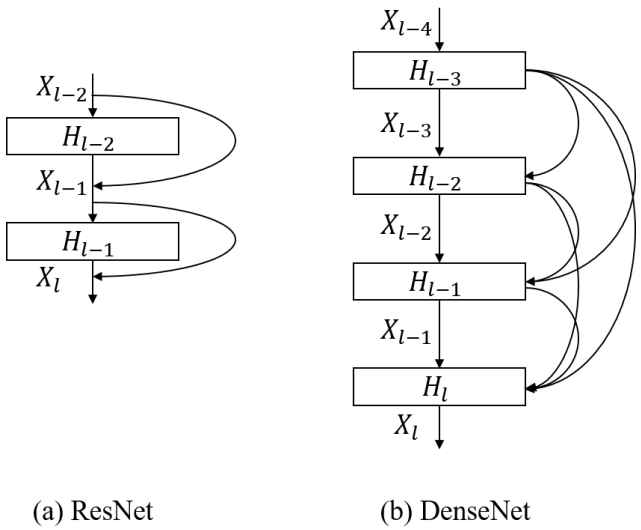


Figure 2. (**a**)The core block of residual network (ResNet) and (**b**) dense convolutional network (DenseNet).

Based on ResNet, DenseNet connect all layers directly with each other to ensure maximum information flow through the network all the time. To maintain the feed-forward nature, each layer concatenates the outputs of all previous layers as inputs in the channel dimension and transmits its own feature maps to all subsequent layers. Figure 2b illustrates this layout. Accordingly, the input x_l of l^{th} layer can be formulated as:

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]) \tag{2}$$

where H_l refers to a module consists of convolution layers, activation layers, and BN layers, and $[x_0, x_1, \cdots, x_{l-1}]$ denotes the concatenation of the feature maps generated by all preceding layers.

2.3. Attention Mechanism

Given that the recognition ability of the different bands varies, the same object tends to show different spectral responses to different bands. Plus, different areas of the data cube contain different semantic information. Such prior information can facilitate the competence of the model once when it is fully exploited. The attention mechanism is exactly the powerful technique that meets the demands. The essence of the attention mechanism is to obtain a new representation with linear weighting based on the correlations between objects, which can be interpreted as a method of feature transformation. To date, the attention mechanism has been successfully applied to various tasks, such as video classification [41], machine translation [42] and scene segmentation [43].

Among the diverse attention models, the self-attention [42] is popular, which computes a weighted summation of location contexts. Non-local [40] first introduced the self-attention mechanism to computer vision tasks. DANet [34] treated the Non-local operation as the spatial attention, and further proposed the channel attention, integrating two branches as an overall framework. A^2 net [44] used a dual-attention block to gather crucial features from entire spatio-temporal spaces into a compact set and then adaptively distribute them to each position.

However, these methods tend to drive each pixel to capture global information, resulting in attention maps with high time and space complexity. Motivated by the success of attention in the above works, EMANet [45] rethought the attention mechanism from the perspective of the expectation-maximization (EM) algorithm and computed attention maps in an iterative manner, significantly alleviating the burden of computation. As shown in Figure 3, a set of bases representing the input feature is initialized first, then with the EM algorithm, the update of the attention maps is executed in E step and the update of bases is executed in M step. Two steps are conducted alternately until convergence. Such mechanism can be integrated into a unit called Expectation-Maximization Attention Unit (EMAU), which can be conveniently inserted to CNNs.

Suppose an input feature map is $X \in R^{N \times C}$ and the bases are initialized as $\mathcal{B} \in R^{K \times C}$. In E step, we use bases to generate the attention maps $Y \in R^{N \times K}$ according to the following formulations:

$$y_{nk} = \frac{\mathrm{K}(x_n, \beta_k)}{\sum_{i=1}^{K} \mathrm{K}(x_n, \beta_i)} \tag{3}$$

$$\mathrm{Z} = \mathrm{softmax}\left(\mathrm{X}\mathcal{B}^T\right) \tag{4}$$

where y_{nk} represents the weight of the contribution of the k-th base β_k to the n-th pixel x_n. Equation (4) is the matrix calculation version of Equation (3), which is the actually application in the experiment.

In M step, the attention maps are used to update the bases:

$$\beta_k = \frac{\sum_{n=1}^{N} y_{nk} x_n}{\sum_{n=1}^{N} y_{nk}} \tag{5}$$

where the bases $\mathcal{B}$ is the weighted sum of X to keep both in the same representation space, aiming to guarantee the robustness of iterations.

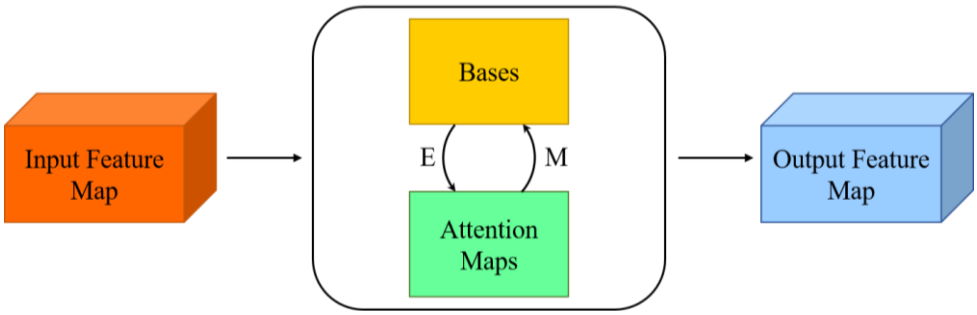


Figure 3. Expectation-maximization (EM) attention operation.

After two steps are executed alternately for T times, $\mathcal{B}$ and Y could converge approximately, which is guaranteed by the property of the EM algorithm. Experimental results also demonstrate that the number of iterations T is a small constant, i.e., expectation-maximization attention can converge quickly. Then, the final $\mathcal{B}$ and Y are used to reconstruct X. The new X, notated as $\widetilde{X}$, can be formulated as:

$$\widetilde{X} = Y\mathcal{B} \tag{6}$$

here $\widetilde{X}$ can be deemed as a low-rank version of X.

3. Methodology

This section is structured as follows. First, we introduce the framework of the proposed method. Second, two branches respectively focusing on spectral information and spatial information are described in detail. Third, fusion operations of spectral and spatial branches are discussed. Finally, several techniques aimed at boosting the network performance are covered.

3.1. Framework of the Proposed Model

The flowchart in Figure 4 depicts the proposed model for HSI classification. Generally, it consists of two branches: the spectral branch and the spatial branch. Moreover, Expectation-Maximization attention modules are incorporated into both branches to apply attention-based feature refinement. Concatenation or weighted sum are implemented subsequently to fuse bipartite features. Finally, classification is performed with the SoftMax function.

Concretely, let the HSI data set be $\mathcal{H} \in R^{h\times w\times d}$, where h, w and d denote the height and width of the spatial dimensions and the spectral bands. Assume that $\mathcal{H}$ is composed of N labeled pixels $X = \{x_1, x_2, \cdots, x_N\} \in R^{1\times 1\times d}$ and the corresponding category label set is $Y = \{y_1, y_2, \cdots, x_N\} \in R^{1\times 1\times C}$, where C represents the numbers of land cover classes. To effectively exploit the inherent information in HSI, a common practice is to form a 3D patch cube with several pixels surrounding the given pixel. In this manner, X can be decomposed into a new data set $Z = \{z_1, z_2, \cdots, z_N\} \in R^{w\times w\times d}$, where w is the width of cubes. If the target pixel is on the edge of the image, the values of adjacent missing pixels are set to zero. Then, Z is randomly divided into training, validation and testing sets denoted by Z_{train}, Z_{val} and Z_{test}. Accordingly, their corresponding label sets are Y_{train}, Y_{val} and Y_{test}. For each configuration of the model, the training set is used to optimize the parameters while the validation set is used to supervise the training process and select the best-trained model. Finally, the test set is used to verify the performance of the best-trained model.

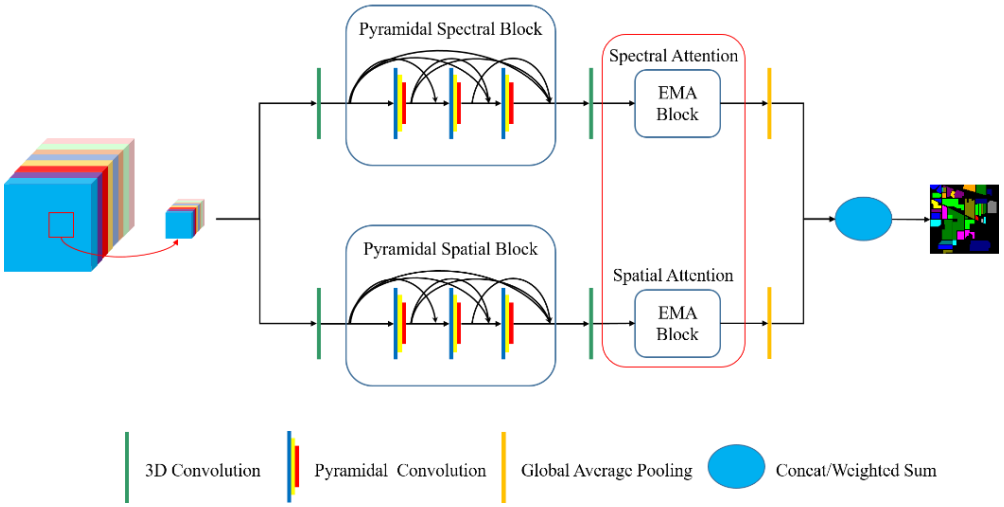


Figure 4. Flowchart of the proposed model.

3.2. Pyramidal Spectral Branch and Pyramidal Spatial Branch

As shown in Figure 4, the spectral branch and the spatial branch consist of PyConv and EMA. First, the pyramidal blocks used in two branches will be described in detail.

Generally, a 3D convolutional layer is first applied to perform a feature transformation on the HSI cube in the spectral dimension, reducing the computational overhead. Then, a pyramidal spectral block is attached. As shown in Figure 5, each layer in the pyramidal convolution consists of three 3D convolution operations with decreasing levels in the spectral dimension, discriminated by blue, yellow and red, respectively. The kernel sizes of the 3D convolution operations in each layer are set to $1 \times 1 \times 7$, $1 \times 1 \times 5$, $1 \times 1 \times 3$, respectively. Furthermore, to make the network powerful and converge rapidly, each convolution is subsequently followed by a batch normalization (BN) layer to regularize and an activation function Mish [46] to learn a non-linear representation. The number of output channels in each layer is consistent and can be set to k', then the number of the final output of the block can be formulated as:

$$k = n + 3 \times k' \tag{7}$$

where n is the number of the output channel of the preceding 3D convolution layer and k actually is the number of 3D convolution kernels. However, since only the spectral dimension of these convolution kernels varies and is never equal to 1, it can be assumed that mainly the spectral information is explored.

Similar to the pyramidal spectral block, the pyramidal spatial block is built by leveraging the interspatial relationships of feature maps. As illustrated in Figure 6, in contrast to the pyramidal spectral block, the kernel size of the pyramidal spatial block changes in the spatial dimension while keeping fixed in the spectral dimension. Moreover, a 3D convolution layer is also applied before to compact the spectral dimension of the HSI cube, which is exhibited in Figure 4. Again, each layer in the block not only includes a 3d convolutional layer, but also is combined with a batch normalization layer and a Mish activation function layer. The relationship between the input and output of the pyramidal spatial block is aligned with Equation (7).

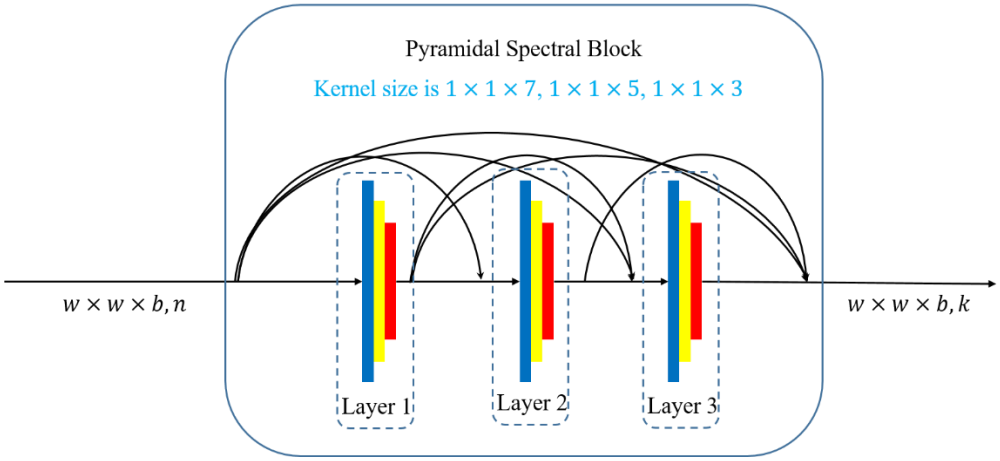


Figure 5. Pyramidal spectral block.

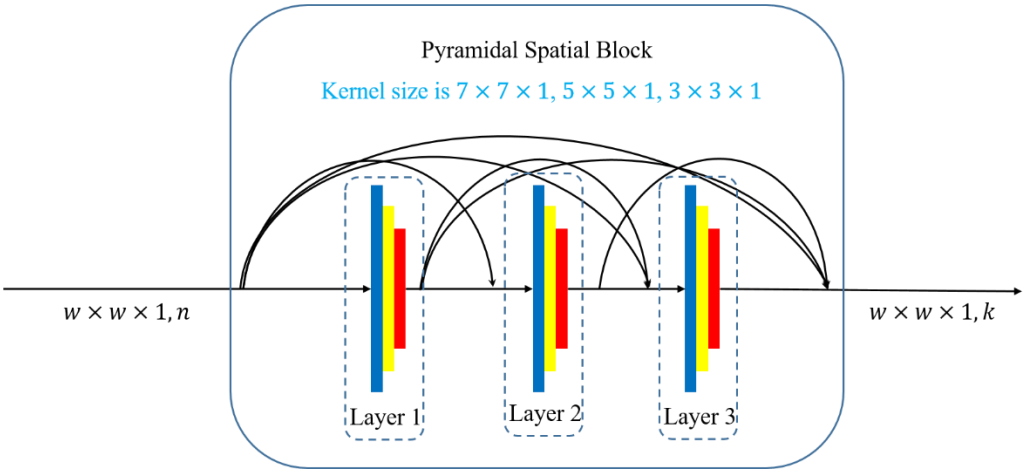


Figure 6. Pyramidal spatial block.

3.3. Expectation-Maximization Attention Block

After attaching the pyramidal spectral or spatial block, a 3D convolutional layer is needed to 'resize' intermediate feature maps for subsequent input to the EMA block. Then, the EMA block follows to refine feature maps. In view of the fact that for the same object, the spectral response may vary dramatically on different bands. In addition, different positions of the extracted feature maps can provide different semantic information for HSI classification. The performance for HSI classification can be improved if such prior information can be properly taken into account. Therefore, the EMA block is introduced. Two EMA blocks located in the spectral and spatial branches are designed with a similar structure. The EMA block located in the spectral branch iterates the attention map along the spectral dimension (denoted as spectral attention), while the EMA block located in the spatial branch iterates the attention map along the spatial dimension (denoted as spatial attention).

As shown in Figure 7, given an intermediate feature map X as input, a compact base set is initialized with Kaiming's initialization [47]. Then, attention maps can be generated in E step and the base set can be updated in M step, as described in Section 2.3. After a few iterations, with the converged bases and attention maps, a new refined feature map $\hat{X}$ can be obtained. Instead of outputting $\hat{X}$ directly, a small factor α is adopted to equilibrate X with $\hat{X}$. Multiplying $\hat{X}$ by α and then adding it to X, the final output $\overline{X}$ is generated. This operation facilitates the stability of the training and empirical performance validates the potency.

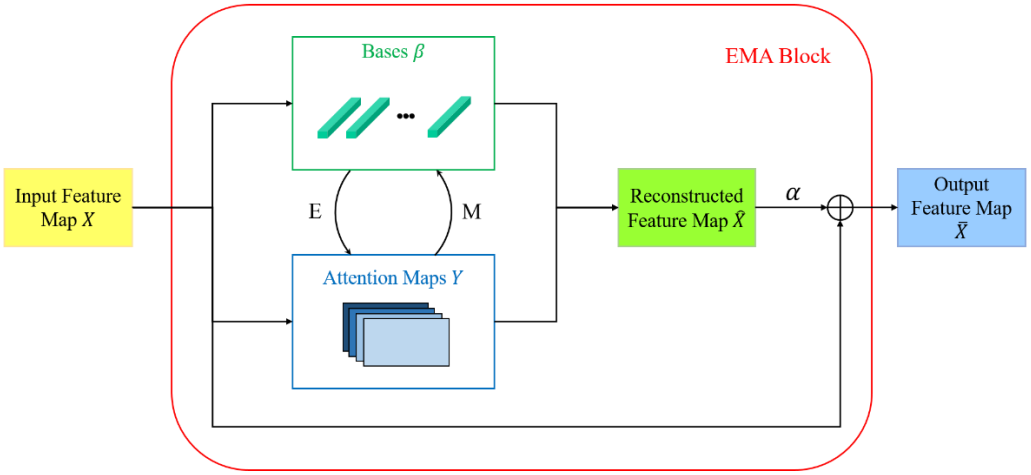


Figure 7. Overall structure of the expectation-maximization attention (EMA) block.

Back to the initialization of bases, this is actually a key point. The procedure described above only portrays the steps to implement EMA on a single image. However, thousands of images must be processed in the HSI classification task. The spectral feature distribution and spatial feature distribution are distinct for each image, so the bases β computed upon an image should not be the paradigm for all images. In this paper, we choose to run EMA on each image and consistently update the initial values of the bases β_0 during the training process with the following strategy:

$$\beta_0 = \gamma\beta_0 + (1-\gamma)\beta_T \tag{8}$$

where β_0 represents the initial values of bases, β_T is generated after iterating over an image and $\gamma \in [0, 1]$.

3.4. Fusion of Spectral and Spatial Branches

With the aid of the spectral branch and spatial branch, multiple feature maps are generated. Then, how to fuse them to obtain a desirable classification result is a problem. Generally, there are two options, add or concatenation. Here, spatial features and spectral features are added with a certain weight, which is constantly adjusted by back-propagation during the training process. Both fusion operations are experimented and the results are detailed in Section 5.5. Once the fusion is finished, the feature maps subsequently flow through the fully connected layer and the SoftMax activation function and finally the classification result is obtained.

3.5. Network Training

3.5.1. A New Activation Function

The activation function is an important element in a deep neural network and the rectified linear unit (ReLU) is often favored. Recently, Mish [46], a self-regularized non-monotone activation function, has received increasing attention. The formula for Mish is as follows:

$$mish(x) = x * tan\ h(\ln(1 + e^x)) \tag{9}$$

where x is the input of the activation function.

The graph of Mish and ReLU can be seen in Figure 8. Unlike ReLU, Mish allows small negative inputs inflow to improve the model performance and keep the network sparsity instead of pruning all the negative inputs. Moreover, Mish is a smooth function and continuously differentiable, which is beneficial to optimization and generalization.

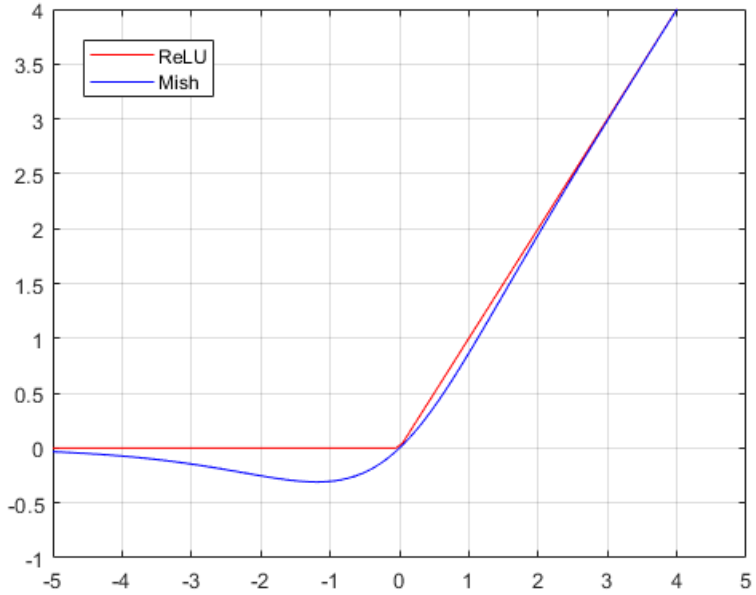


Figure 8. The graph of Mish and ReLU.

3.5.2. Other Training Tricks

To mitigate the overfitting problem, dropout [48] is a typical strategy. Given a percentage p, which is selected as 0.5 in the proposed model, the network would drop out hidden or visible units temporarily. In the case of stochastic gradient descent, a new network is trained in each mini-batch due to the property of random dropping. Moreover, dropout can make only a few units in the network possess high activation ability, which is conducive to the sparsity of the network. In our framework, a dropout layer is applied after the EMA block.

In addition, the early stopping strategy, and the dynamic learning rate adjustment method are also adopted to accelerate the network training. Specifically, early stopping means stopping the training if the loss function no longer decreases in a couple of training epochs (which is 20 in our method). Dynamic learning rate means that we adjust the learning rate during the training process to avoid the model trapped in a local optimum. Herein, we use the cosine annealing [49] strategy, which is formulated as follows:

$$\eta_t = \eta_{min}^i + \frac{1}{2}\left(\eta_{max}^i - \eta_{min}^i\right)\left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right) \tag{10}$$

where η_t is the learning rate for the i-th run while η_{min}^i and η_{max}^i are ranges for the learning rate. T_{cur} denotes how many epochs have been executed since the last restart and T_i represents the number of epochs in one restart cycle.

4. Experiment

4.1. Datasets Description

In the experiments, four publicly available datasets, the Pavia University (UP)dataset, the Indian Pines (IP) dataset, the Salinas Valley (SV) dataset, and the Botswana dataset (BS), are applied to conduct a series of experiments.

Pavia University (UP): captured by the reflective optics imaging spectrometer (ROSIS-3) sensor at the University of Pavia, northern Italy, the Pavia University dataset is comprised of 103 bands with spatial resolution of 1.3 mpp in the wavelength ranging from 0.43 µm to 0.86 µm. The spatial size is 610 × 340 pixels and 9 land cover classes are involved.

Indian Pines (IP): captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in the north-western Indiana, the Indian Pines dataset is comprised of

200 bands with spatial resolution of 20 mpp in the wavelength ranging from 0.4 μm to 2.5 μm. The spatial size is 145 × 145 pixels and 16 land cover classes are involved.

Salinas Valley (SV): captured by the AVIRIS sensor the AVIRIS sensor over the agricultural area described as SV in California, CA, USA, the Salinas Valley dataset is comprised of 204 bands with spatial resolution of 3.7 mpp in the wavelength ranging from 0.4 μm to 2.5 μm. The spatial size is 512 × 217 pixels and 16 land cover classes are involved.

Botswana (BS): captured by the NASA EO-1 satellite over the Okavango Delta, Botswana, the Botswana dataset is comprised of 145 bands with spatial resolution of 20 mpp in the wavelength ranging from 0.4 μm to 2.5 μm. The spatial size is 1476 × 256 pixels and 14 land cover classes are involved.

The performance of deep-learning-based models strongly depends on the data. Generally, the more labeled data used for training, the better the model performs. Currently, many HSI classification methods can achieve almost 100% accuracy with sufficient training samples. Model performance given the lack of training samples is noteworthy. Therefore, the size of the training samples and validation samples in the experiments are set relatively small to challenge the proposed model. In addition, to conveniently compare with the previous methods, we follow the settings in [35], i.e., the proportion of samples for training and validation is both set to 3% for IP, 0.5% for UP and SV and 1.2% for BS.

4.2. Experimental Configuration

All experiments were executed on the same platform configured with Intel Core i7-8700K processor at 3.70 GHz, 32 GB of memory and an NVIDIA GeForce GTX 1080Ti GPU. The software environment is the system of window 10 (64 bit) home and deep-learning frameworks of PyTorch.

Optimization is performed by Adam optimizer with the batch size of 16 and learning rate of 0.0005. To assess the results quantitatively, three metrics are adopted: overall accuracy (OA), average accuracy (AA), and Kappa coefficient.

To assess the effectiveness of our approach, several methods are adopted for comparison. The SVM with a radial basis function (RBF) kernel [6] is selected as a representative of the traditional methods. CDCNN [22], SSRN [28] and FDSSC [29] are chosen on behalf of the deep-learning-based approaches. DBMA [33] and DBDA [35], similar to our model with a two-branch structure, are selected as the state-of-the-art double-branch models. The parameters of each model are set according to the original paper. Given that the codes are available, the results of the classification with these methods on the four datasets are in accordance with our own replication. For a fair comparison, all algorithms are executed ten times and the best results are retained.

4.3. Classification Results

4.3.1. Classification Results for the IP Dataset

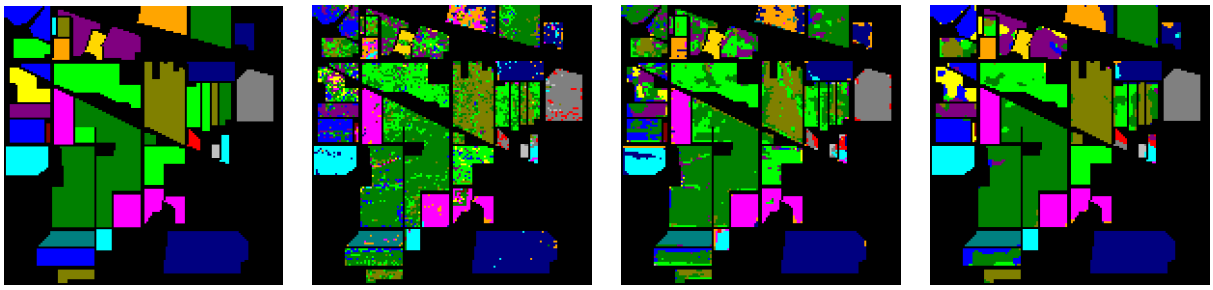
The accuracy for the IP dataset obtained by different methods is shown in Table 1, where the best accuracy is in bold for each category and for the three metrics. The corresponding classification maps are also illustrated in Figure 9.

The proposed model yields the best results, i.e., 95.90% in OA, 96.19% in AA and 0.9532 in Kappa, as shown in Table 1. CDCNN obtains the lowest accuracy since the training samples are too limited for the 2DCNN-based model. Compared with CDCNN, SVM performs a little better; however, the pepper noise is quite severe, which is shown in Figure 9b. Owing to the integration of spatial and spectral information by 3DCNN, both SSRN and FDSSC are far superior to SVM and CDCNN, exceeding them by almost 20% in OA. Furthermore, FDSSC draws on the dense connection, resulting in better performance. DBMA and DBDA follow basically the same idea i.e., two branches are used to extract spectral and spatial features and the attention mechanism are introduced. However, they are prone to overfitting when the training samples are limited. Moreover, the attention mechanisms they use are simple and cannot distinguish different classes well. In contrast, our proposed model not only uses two branches to extract features, but also introduces an

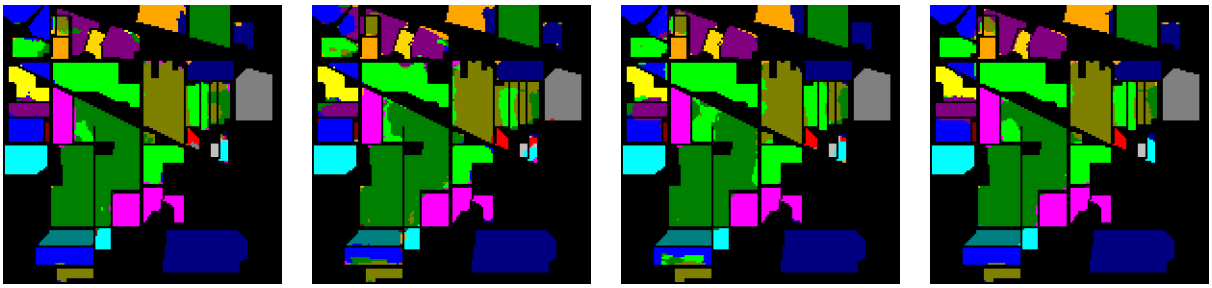
attention mechanism based on the EM algorithm, which can iteratively update the attention map and reduce the intra-class feature variance, thus making it easier to distinguish different class targets. As can be seen in Table 1, our model performs well balanced and excellent on each category, without extremely low scores. This demonstrates the superior discriminative capability of our model for each category.

Table 1. The classification accuracy for the IP dataset based on 3% training samples.

Class	SVM	CDCNN	SSRN	FDSSC	DBMA	DBDA	Proposed
1	24.19	30.00	66.67	90.91	65.15	85.42	**100**
2	56.71	53.90	91.71	**97.82**	93.09	83.64	92.88
3	65.09	60.88	86.09	95.64	95.27	**99.53**	96.01
4	39.64	28.06	63.56	97.14	99.53	**100**	87.34
5	87.33	81.04	95.60	**99.76**	97.80	99.05	98.17
6	83.88	88.93	**99.56**	94.38	95.87	99.27	96.17
7	57.50	63.16	**100**	86.21	75.00	87.50	95.45
8	89.29	90.72	95.51	97.60	**100**	99.55	**100**
9	22.58	53.57	100	69.57	45.16	84.21	**100**
10	66.70	33.95	82.04	90.53	77.67	86.99	**94.78**
11	62.50	68.32	84.13	95.42	93.44	96.80	**96.82**
12	51.87	44.22	87.88	**96.69**	83.30	91.56	91.65
13	94.79	67.55	97.95	**100**	99.44	**100**	98.41
14	90.43	93.36	92.72	95.52	93.39	92.07	**98.99**
15	62.82	77.11	90.15	92.86	83.19	91.91	**95.70**
16	98.46	80.77	97.65	**98.82**	95.45	95.45	96.71
OA	69.35	61.89	88.43	95.51	91.27	93.14	**95.90**
AA	65.86	63.47	89.45	93.67	87.05	93.31	**96.19**
Kappa	64.66	57.64	86.75	94.88	90.05	92.18	**95.32**



(**a**) Ground Truth (**b**) SVM (**c**) CDCNN (**d**) SSRN



(**e**) FDSSC (**f**) DBMA (**g**) DBDA (**h**) Proposed

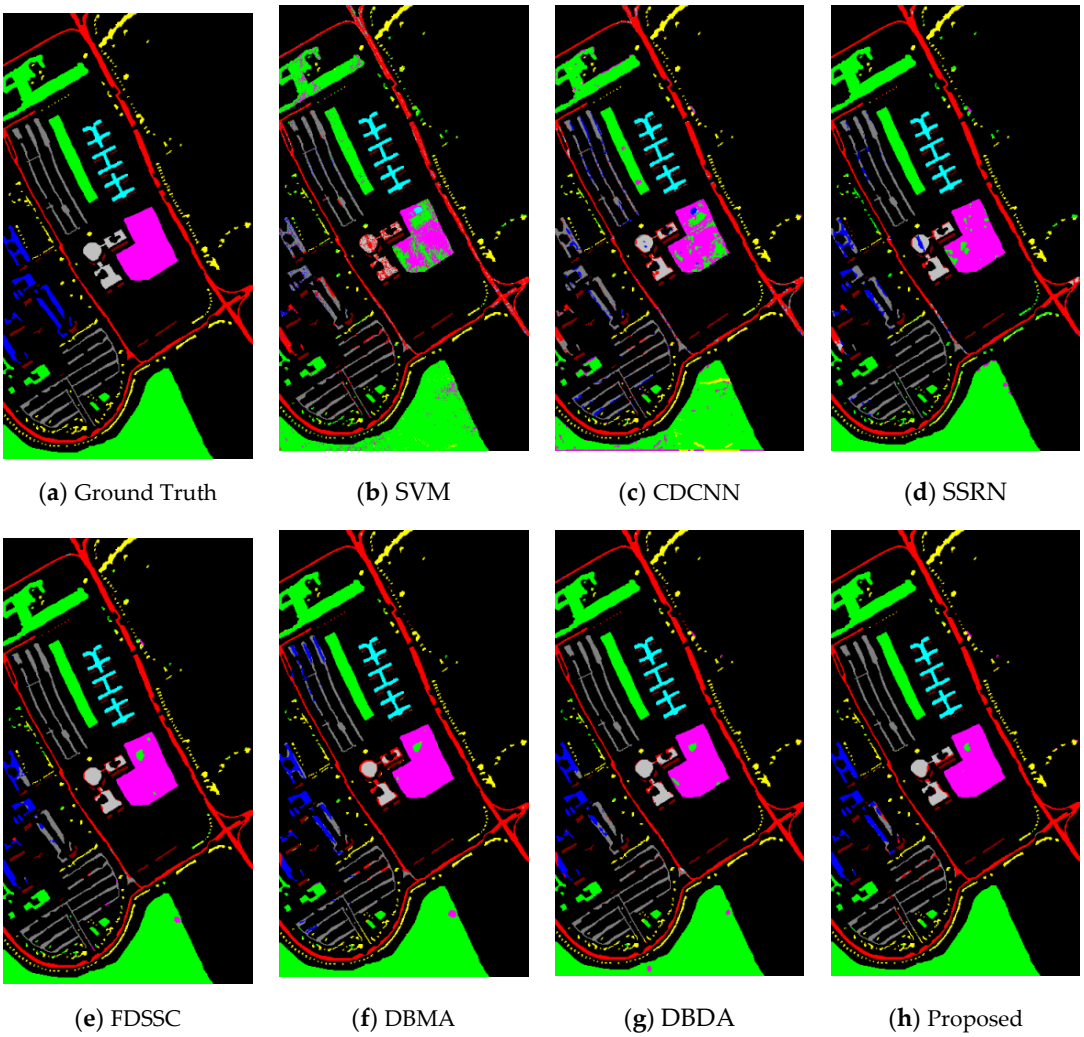
Figure 9. Classification maps achieved by seven different methods for the IP dataset. (**a**) Ground-truth. (**b**–**h**) The classification maps of the corresponding methods.

4.3.2. Classification Results for the UP Dataset

The accuracy for the UP dataset obtained by different methods is shown in Table 2, where the best accuracy is in bold for each category and for the three metrics. The corresponding classification maps are also illustrated in Figure 10.

Table 2. The classification accuracy for the UP dataset based on 0.5% training samples.

Class	SVM	CDCNN	SSRN	FDSSC	DBMA	DBDA	Proposed
1	80.27	87.17	**99.35**	95.79	90.10	94.43	93.78
2	86.95	93.59	96.14	97.46	98.52	98.52	**99.21**
3	71.74	43.80	95.99	99.67	74.93	98.86	**99.72**
4	96.45	86.71	99.56	**99.81**	95.10	98.46	97.81
5	90.85	98.67	**100**	99.63	99.70	99.55	99.92
6	77.03	83.76	96.08	97.57	97.93	97.90	**99.15**
7	69.71	90.17	73.41	**100**	98.35	97.61	**100**
8	67.31	67.51	79.25	79.27	84.75	83.57	**91.52**
9	99.89	97.18	**100**	99.25	98.82	99.45	99.57
OA	83.08	87.00	94.23	95.65	94.52	96.31	**97.60**
AA	82.24	83.17	93.31	96.49	93.13	96.48	**97.85**
Kappa	77.07	82.71	92.31	94.19	92.72	95.08	**96.82**



(**a**) Ground Truth (**b**) SVM (**c**) CDCNN (**d**) SSRN

(**e**) FDSSC (**f**) DBMA (**g**) DBDA (**h**) Proposed

Figure 10. Classification maps achieved by seven different methods for the UP dataset. (**a**) Ground-truth. (**b**–**h**) The classification maps of the corresponding methods.

As shown in Table 2, our method achieves the best results on the three metrics. In particular, the average improvement over the second-best model, DBDA, is +1.29%, +1.37%, 1.74% for OA, AA, and Kappa metrics, respectively. Specifically, for each class, the best results are obtained by our method in 5 out of 9 classes. In addition, it is worth noting that in class 8, which is the most difficult to classify, only our model exceeds 90% in classification accuracy. Class 8 is represented by the dark gray line in Figure 10a, which is too slender for models to capture. Please note that only DBMA, DBDA and our method achieve the accuracy over 80 % on category 8. This illustrates the advantage of the attention mechanism in capturing fine features. Moreover, the accuracy of our method exceeds 90%, indicating that the attention mechanism adopted by our model stands out.

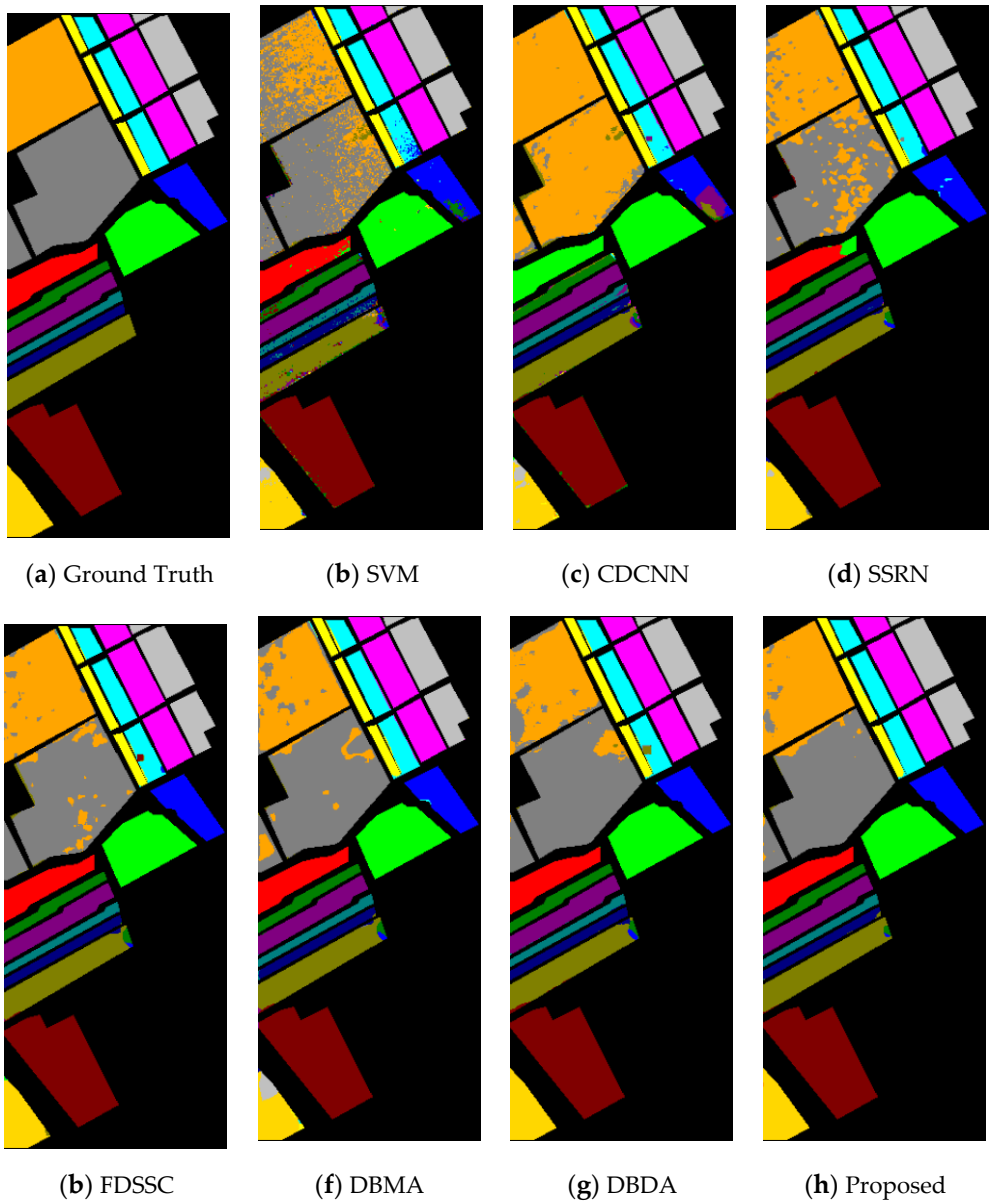
4.3.3. Classification Results for the SV Dataset

The accuracy for the SV dataset obtained by different methods is shown in Table 3, where the best accuracy is in bold for each category and for the three metrics. The corresponding classification maps are also illustrated in Figure 11.

Table 3. The classification accuracy for the SV dataset based on 0.5% training samples.

Class	SVM	CDCNN	SSRN	FDSSC	DBMA	DBDA	Proposed
1	99.85	0	**100**	**100**	**100**	**100**	**100**
2	98.95	64.86	94.53	99.70	**100**	**100**	**100**
3	89.88	97.52	95.92	96.21	97.93	98.88	**99.29**
4	97.30	92.61	96.44	96.30	90.40	93.58	**100**
5	93.56	98.51	98.53	99.59	97.91	**99.60**	98.19
6	99.79	97.01	99.97	99.59	98.15	**100**	**100**
7	91.33	94.43	98.94	100	92.42	98.22	**100**
8	74.73	93.88	92.93	90.06	91.85	**99.65**	97.16
9	97.69	99.13	98.60	98.55	99.56	97.24	**99.79**
10	90.01	82.96	98.26	**98.69**	98.27	97.85	98.05
11	75.92	85.48	93.72	93.72	93.14	90.75	**96.00**
12	95.19	75.87	99.84	**100**	99.12	100	99.90
13	94.87	98.49	99.56	**100**	98.70	100	99.78
14	89.26	96.17	96.69	93.62	97.93	96.18	**99.81**
15	75.86	41.55	73.09	96.94	88.61	81.42	**94.61**
16	99.03	99.55	**100**	**100**	99.93	**100**	**100**
OA	88.09	73.72	93.00	96.57	95.27	95.81	**98.33**
AA	91.45	82.38	96.06	97.69	96.49	97.08	**98.91**
Kappa	86.71	71.18	92.23	96.18	94.73	95.35	**98.14**

Again, the proposed model obtains the best results with 98.33% OA, 98.91% AA, and 0.9814 Kappa. On the class 15, none of the methods achieves over 90% accuracy except ours. This can be observed in Figure 11. If we concentrate on the yellow area and the gray area in the upper left corner of classification maps, it can be found that these two areas interfere with each other terribly in all the models except ours.



(**a**) Ground Truth (**b**) SVM (**c**) CDCNN (**d**) SSRN

(**b**) FDSSC (**f**) DBMA (**g**) DBDA (**h**) Proposed

Figure 11. Classification maps achieved by seven different methods for the SV dataset. (**a**) Ground-truth. (**b**–**h**) The classification maps of the corresponding methods.

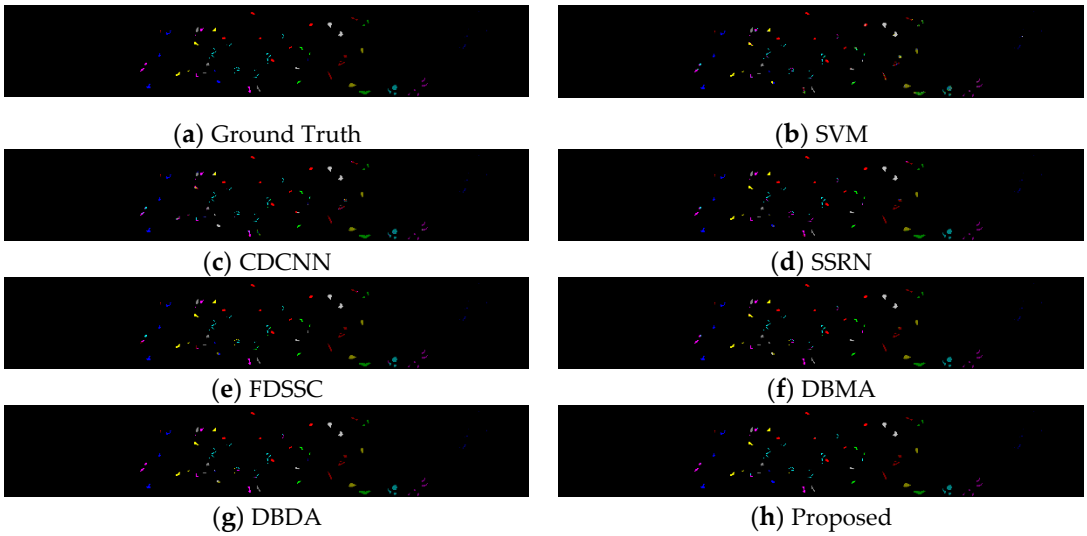
4.3.4. Classification Results for the BS Dataset

The accuracy for the BS dataset obtained by different methods is shown in Table 4, where the best accuracy is in bold for each category and for the three metrics. The corresponding classification maps are also illustrated in Figure 12.

Since the BS dataset is small and only with 3248 labeled samples, training samples may be scarce for the model. Nevertheless, the proposed method yields the best results, which demonstrates the competency of our method in exploiting spectral information and spatial information.

Table 4. The classification accuracy for the BS dataset based on 1.2% training samples.

Class	SVM	CDCNN	SSRN	FDSSC	DBMA	DBDA	Proposed
1	**100**	91.96	99.62	97.41	98.13	97.72	99.62
2	70.71	80.20	**100**	74.24	89.91	98.99	**100**
3	84.11	90.50	98.68	**100**	**100**	**100**	**100**
4	69.96	**96.97**	95.87	86.83	91.63	92.95	91.67
5	82.63	72.40	83.92	85.92	94.59	**96.97**	87.46
6	65.71	65.51	69.25	92.78	77.92	85.62	**100**
7	78.78	71.65	**100**	**100**	86.64	87.54	**100**
8	65.88	90.00	97.51	92.45	**100**	**100**	**100**
9	75.19	77.74	92.11	84.05	95.24	**100**	99.34
10	69.82	90.95	83.99	86.83	83.56	86.52	**98.78**
11	95.50	83.90	97.34	100	99.32	**100**	**100**
12	93.10	88.24	**100**	86.70	99.44	**100**	**100**
13	76.25	71.51	**100**	**100**	**100**	**100**	**100**
14	90.41	68.86	**100**	**100**	**100**	**100**	**100**
OA	78.63	79.84	92.86	92.00	93.11	95.35	**98.10**
AA	79.57	81.46	94.16	91.94	94.03	96.17	**98.35**
Kappa	76.88	78.13	92.26	91.34	92.53	94.97	**97.94**



(**a**) Ground Truth (**b**) SVM (**c**) CDCNN (**d**) SSRN (**e**) FDSSC (**f**) DBMA (**g**) DBDA (**h**) Proposed

Figure 12. Classification maps achieved by seven different methods for the BS dataset. (**a**) Ground-truth. (**b**–**h**) The classification maps of the corresponding methods.

5. Discussion

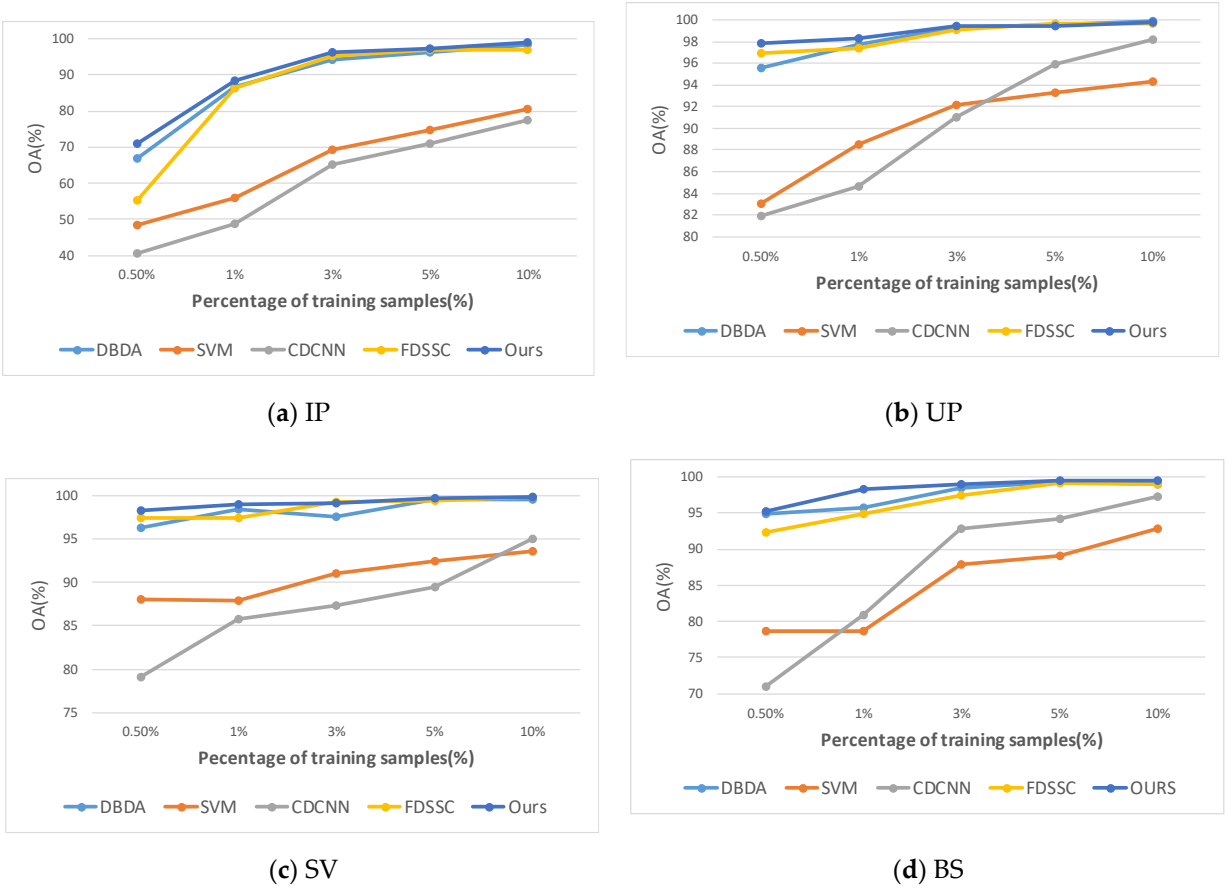
In this part, more experiments are carried out to comprehensively discuss the impacts and capabilities of the relevant components in the proposed model.

5.1. Investigation of the Proportion of Training Samples

It is well known that the amount of training samples significantly affects the performance of deep-learning models. In this section, we randomly select 0.5%, 1%, 3%, 5% and 10% of samples as training sets to investigate the performance of different models with different proportion of training data. The experimental results are illustrated as Figure 13.

It is expected that as the percentage of training data increases, the OA of all methods improves. Moreover, all three approaches using 3DCNN consistently outperform CDCNN with only 2DCNN and the traditional model SVM. In addition, all three methods using 3DCNN consistently outperform the CDCNN with only 2DCNN and the traditional model SVM. Also, the discrepancy between these methods is narrowing as the amount of training

samples increases. Please note that our proposed method obtains the best results regardless of the proportion, especially when the samples are not sufficient.



(**a**) IP (**b**) UP

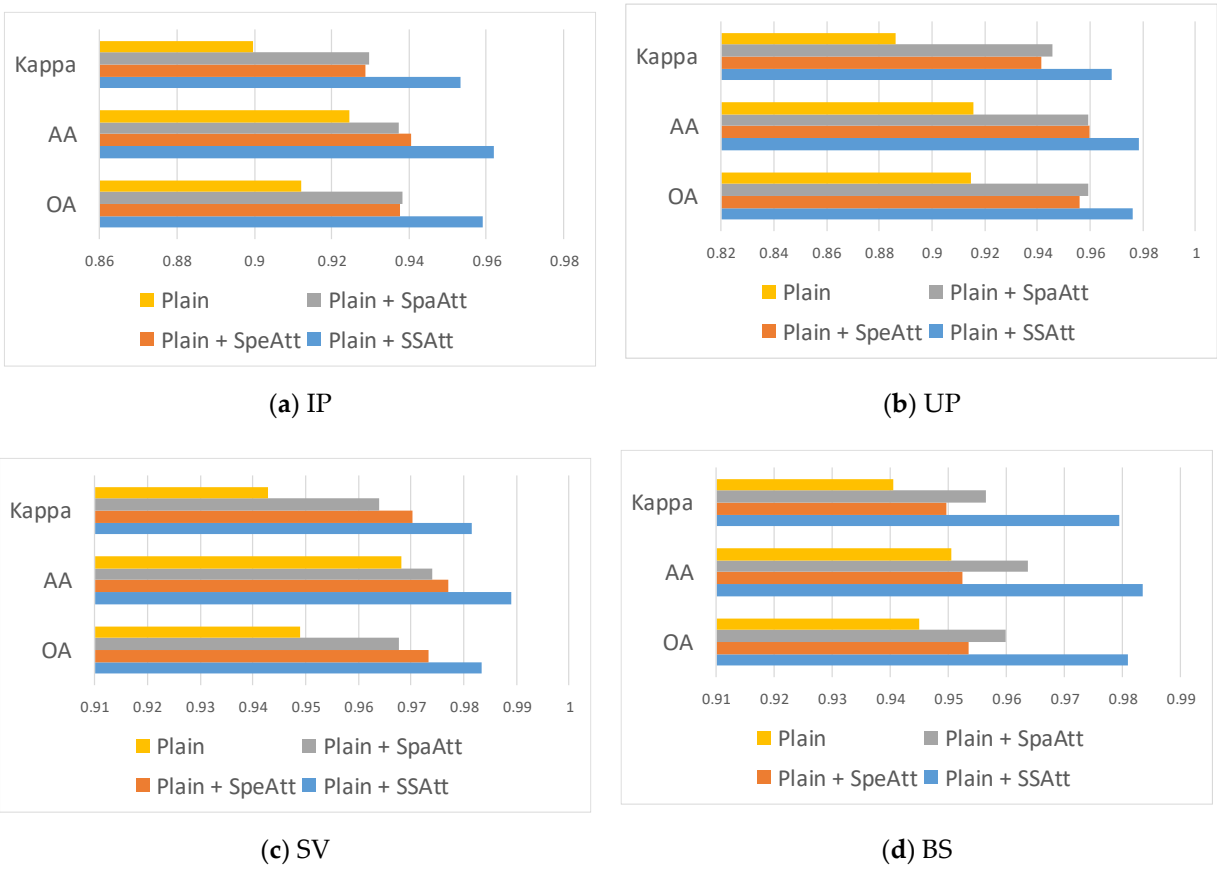
(**c**) SV (**d**) BS

Figure 13. The OA of SVM, CDCNN, FDSSC, DBDA and our method with different proportions of training samples on the (**a**) IP, (**b**) UP, (**c**) SV and (**d**) BS.

5.2. Investigation of the Attention Mechanism

Our model integrates spectral attention and spatial attention. In this section, we will test the effectiveness of these attention modules. Specifically, we consider a PyConv-only network without any attention module as a baseline (denoted as Plain). It is a simple double-branch model that extracts spatial and spectral features separately. Moreover, we denote the three derivatives: the subnetwork integrated with spectral attention, the subnetwork integrated with spatial attention and the subnetwork integrated with both as Plain + SpeAtt, Plain + SpaAtt and Plain + SSAtt, respectively.

Figure 14 shows the comparison of the classification results of different networks in terms of OA, AA and Kappa. Different colors indicate different subnetworks. From the figure, we can see that either spectral attention or spatial attention, once integrated into the network, can contribute to the performance of the original network. This confirms the effectiveness of the proposed attention module. In addition, we can observe that the Plain + SSAt outperforms all the other subnetworks. This implies that spectral attention and spatial attention can complement each other to contribute more to the final classification decision.



(**a**) IP (**b**) UP

(**c**) SV (**d**) BS

Figure 14. Classification performance of different components on the (**a**) IP, (**b**) UP, (**c**) SV and (**d**) BS.

5.3. Ablation Study for Iteration Number of the Attention Map

The number of iterations in the EMA block actually affects the performance of the model. We plot the trend of the three metrics OA, AA and Kappa against iteration number as Figure 15. We fix the base at 16 and vary the iteration number to investigate its effect on the model performance. From Figure 15, it can be seen that the model performance basically converges at first, and then the metrics start to oscillate. Specifically, three metrics on the IP dataset perform very unstably due to the unique nature of IP, i.e., the spatial size of IP is relatively small but with 16 categories, which makes the classification more arduous. Accordingly, the iteration number on the IP is set to 2, and that on the UP, SV and BS is set to 3.

5.4. Comparison of the Activation Function

In this article, a new activation function Mish is introduced to enhance the performance of the model. Here, the comparison of performance between Mish and ReLU is illustrated as Figure 16. It is obvious that if the proposed model adopts Mish as the activation function instead of ReLU, the OA could be improved to a certain extent.

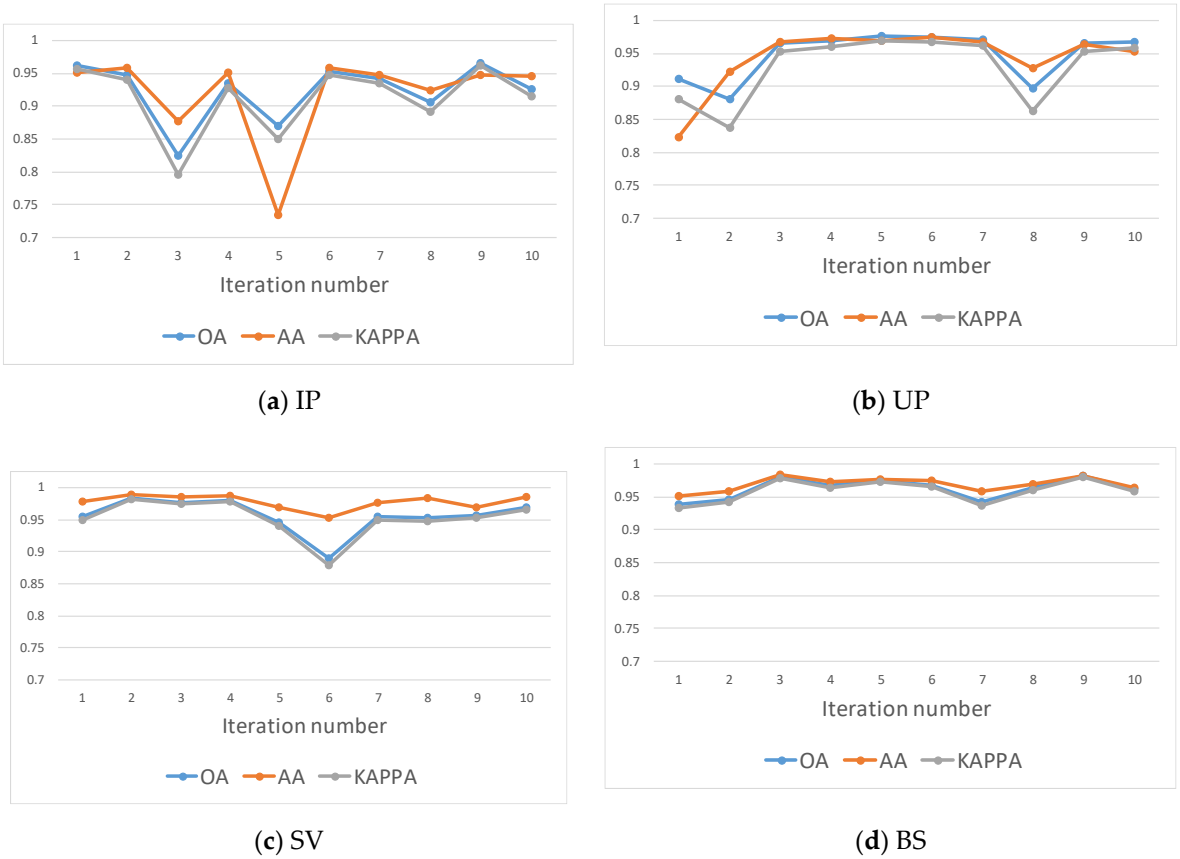


Figure 15. OA, AA and Kappa of the proposed model with different iteration number on the (**a**) IP, (**b**) UP, (**c**) SV and (**d**) BS.

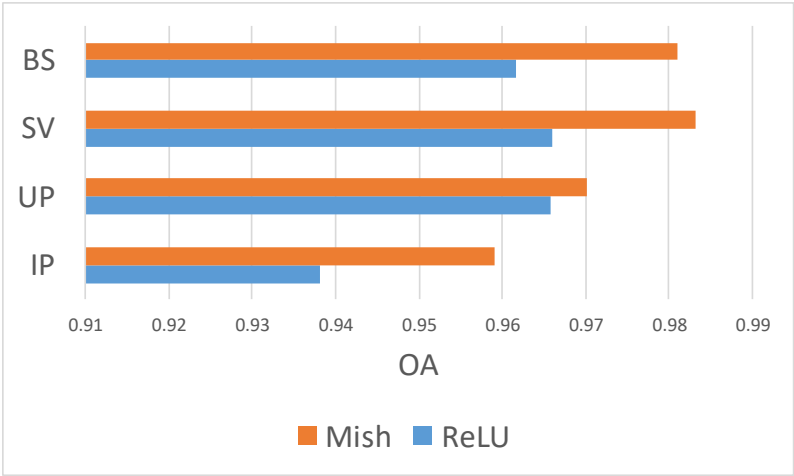


Figure 16. Comparison of OA with activation functions Mish and ReLU on different datasets.

5.5. Comparison of Different Feature Fusion

As illustrated in Figure 17, concatenation outperforms overall. This is as expected, as the spectral and spatial features are in the independent feature space. If addition operation is adopted instead of concatenation, information from different domains tends to be intermixed or even interfere with each other. However, we can see that the addition operation outperforms the concatenation operation on the IP dataset. At first it is attributed to a coincidence, nonetheless, the results remain the same after repeated experiments. Such experimental results may be caused by the specificity of the IP dataset, which has been mentioned in Section 5.3. In contrast to the concatenation operation, weighted addition can

use weights to adjust the influence of spectral and spatial information on the classification results. This property is probably more useful for the IP dataset, since its spatial size is smallest while the numbers of spectral bands categories are large.

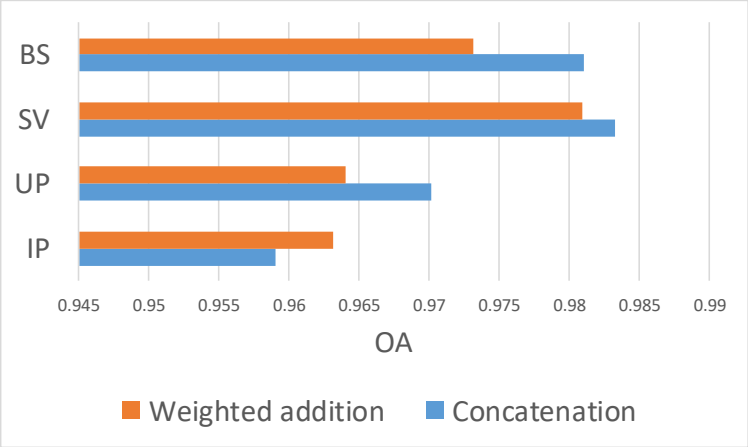


Figure 17. Comparison of OA with different feature fusion operations.

5.6. Investigation of Running Time

A decent method should achieve a favorable accuracy-efficiency trade-off. In this section, we will compare the time costs of the seven algorithms on the IP, UP, SV, and BS datasets. Tables 5–8 show the training time and test time of seven methods on the four datasets.

Table 5. The running time of seven methods on the IP dataset.

Dataset	Algorithm	Training Time (s)	Testing Time (s)
IP	SVM	21.50	1.15
	CDCNN	11.25	1.69
	SSRN	82.98	3.45
	FDSSC	122.94	4.51
	DBMA	47.36	8.56
	DBDA	64.64	7.98
	Proposed	61.34	4.51

Table 6. The running time of seven methods on the UP dataset.

Dataset	Algorithm	Training Time (s)	Testing Time (s)
UP	SVM	3.69	2.94
	CDCNN	8.39	6.28
	SSRN	20.45	10.18
	FDSSC	53.84	12.78
	DBMA	32.06	22.79
	DBDA	16.03	21.09
	Proposed	38.72	12.49

From Tables 5–8, it can be observed that SVM spends less training time and testing time than DL-based methods in most cases. Furthermore, since 2D convolution is more parameter- and computation-conserving than 3D convolution, CDCNN, as a representative of 2D-CNN-based methods, is more time-conserving than 3D-CNN-based methods. Among these five 3D-CNN-based methods, the training time and testing time of our method are moderate. Considering the accuracy of our method is promising, it can be concluded that the proposed method strikes a better balance between accuracy and efficiency.

Table 7. The running time of seven methods on the SV dataset.

Dataset	Algorithm	Training Time (s)	Testing Time (s)
SV	SVM	10.39	4.77
	CDCNN	8.27	9.77
	SSRN	48.29	19.63
	FDSSC	151.29	24.17
	DBMA	110.53	49.92
	DBDA	67.08	46.05
	Proposed	94.38	25.41

Table 8. The running time of seven methods on the BS dataset.

Dataset	Algorithm	Training Time (s)	Testing Time (s)
BS	SVM	1.09	0.21
	CDCNN	5.37	0.50
	SSRN	10.32	0.91
	FDSSC	12.22	1.17
	DBMA	8.49	2.21
	DBDA	9.55	2.08
	Proposed	11.49	1.23

6. Conclusions

In this paper, we propose a novel HSI classification method that consists of a double branch with the pyramidal convolution and an iterative attention. First, the input of the whole framework is not subjected to dimensionality reduction such as PCA. The original 3D data is cropped into 3D cubes as input. Then, two branches are constructed with two novel techniques, namely pyramidal convolution and an iterative attention mechanism, EM attention, to extract spectral features and spatial features, respectively. Meanwhile, a new activation function, Mish, is introduced to accelerate the network convergence and improve the network performance. Finally, with several experiments, we analyze our model from multiple perspectives and demonstrate that the proposed model yields the best or competitive results on four datasets on comparison to other algorithms.

A future direction of our work is to explore better attention mechanisms to obtain finer feature representations. Furthermore, it seems interesting to further reduce the data requirements with new techniques such as a few-shot learning or zero-shot learning.

Author Contributions: Conceptualization, H.S.; formal analysis, H.S.; funding acquisition, G.C., P.F. and Y.Z.; methodology, H.S; validation, H.S; writing—original draft, H.S; writing—review and editing, G.C., Z.G. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Natural Science Foundation of Jiangsu Province under Grant (BK20191284), the NUPTSF under Grant (NY220157), and the National Natural Science Foundation of China under Grant (61801222).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Four publicly available datasets are analyzed in this study, which can be found here: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 15 March 2021).

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

References

1. Zhang, X.; Sun, Y.; Shang, K.; Zhang, L.; Wang, S. Crop classification based on feature band set construction and object-oriented approach using hyperspectral images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4117–4128. [CrossRef]

2. Stuart, M.B.; McGonigle, A.J.S.; Willmott, J.R. Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems. *Sensors* **2019**, *19*, 3071. [CrossRef] [PubMed]
3. Carter, G.A.; Lucas, K.L.; Blossom, G.A.; Holiday, C.L.L.; Mooneyhan, D.S.; Fastring, D.R.; Holcombe, T.R.; Griffith, J.A. Remote sensing and mapping of tamarisk along the colorado river, USA: A comparative use of summer-acquired hyperion, thematic mapper and quickbird data. *Remote Sens.* **2009**, *1*, 318–329. [CrossRef]
4. Tu, B.; Wang, J.; Kang, X.; Zhang, G.; Ou, X.; Guo, L. KNN-based representation of superpixels for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 4032–4047. [CrossRef]
5. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 809–823. [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. Fauvel, M.; Zullo, A.; Ferraty, F. Nonlinear parsimonious feature selection for the classification of hyperspectral images. In Proceedings of the 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lausanne, Switzerland, 24–27 June 2014; pp. 1–4. [CrossRef]
8. Khan, Z.; Shafait, F.; Mian, A. Joint group sparse PCA for compressed hyperspectral imaging. *IEEE Trans. Image Process.* **2015**, *24*, 4934–4942. [CrossRef]
9. Gu, Y.; Wang, C.; You, D.; Zhang, Y.; Wang, S.; Zhang, Y. Representative multiple kernel learning for classification in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 2852–2865. [CrossRef]
10. Fang, L.; Li, S.; Duan, W.; Ren, J.; Benediktsson, J.A. Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6663–6674. [CrossRef]
11. Yu, H.; Gao, L.; Liao, W.; Zhang, B.; Pizurica, A.; Philips, W. Multiscale superpixel-level subspace-based support vector machines for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2142–2146. [CrossRef]
12. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
13. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 14–19 June 2020. [CrossRef]
14. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
15. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019. [CrossRef]
16. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
17. Zhao, C.; Wan, X.; Zhao, G.; Cui, B.; Liu, W.; Qi, B. Spectral-spatial classification of hyperspectral imagery based on stacked sparse autoencoder and random forest. *Eur. J. Remote Sens.* **2017**, *50*, 47–63. [CrossRef]
18. Li, T.; Zhang, J.; Zhang, Y. Classification of hyperspectral image based on deep belief networks. In Proceedings of the 2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, 27–30 October 2014; pp. 5132–5136. [CrossRef]
19. Zhong, P.; Gong, Z.; Li, S.; Schonlieb, C.B. Learning to diversify deep belief networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3516–3530. [CrossRef]
20. Yu, C.; Zhao, M.; Song, M.; Wang, Y.; Li, F.; Han, R.; Chang, C.I. Hyperspectral image classification method based on cnn architecture embedding with hashing semantic feature. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1866–1881. [CrossRef]
21. Mei, S.; Ji, J.; Bi, Q.; Hou, J.; Du, Q.; Li, W. Integrating spectral and spatial information into deep convolutional neural networks for hyperspectral classification. In Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 5067–5070. [CrossRef]
22. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef]
23. Zhao, W.; Du, S. Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
24. Ben Hamida, A.; Benoit, A.; Lambert, P.; Ben Amar, C. 3-D deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434. [CrossRef]
25. Zhong, S.; Chen, S.; Chang, C.-I.; Zhang, Y. Fusion of spectral-spatial classifiers for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–20. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
27. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708. [CrossRef]
28. 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]

29. Fang, B.; Li, Y.; Zhang, H.; Chan, J.C.W. hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism. *Remote Sens.* **2019**, *11*, 159. [CrossRef]
30. Zheng, Y.; Li, J.; Li, Y.; Guo, J.; Wu, X.; Chanussot, J. Hyperspectral pansharpening using deep prior and dual attention residual network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8059–8076. [CrossRef]
31. 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]
32. Sun, H.; Zheng, X.; Lu, X.; Wu, S. Spectral-spatial attention network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3232–3245. [CrossRef]
33. 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]
34. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3141–3149. [CrossRef]
35. 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]
36. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 4489–4497. [CrossRef]
37. Ying, X.; Wang, L.; Wang, Y.; Sheng, W.; An, W.; Guo, Y. Deformable 3D convolution for video super-resolution. *IEEE Signal. Process. Lett.* **2020**, *27*, 1500–1504. [CrossRef]
38. Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; Kautz, J. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
39. Ge, C.; Qu, Q.; Gu, I.Y.H.; Store Jakola, A. 3D multi-scale convolutional networks for glioma grading using MR images. In Proceedings of the International Conference on Image Processing(ICIP), Athens, Greece, 7–10 October 2018. [CrossRef]
40. Duta, I.C.; Liu, L.; Zhu, F.; Shao, L. Pyramidal convolution: Rethinking convolutional neural networks for visual recognition. *arXiv* **2020**, arXiv:2006.11538, 1–16.
41. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 7794–7803. [CrossRef]
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762, 5999–6009.
43. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J.; Zhao, H.; Zhang, Y.; Liu, S.; et al. PSANet: Point-wise spatial attention network for scene parsing—Supplementary material. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 267–283.
44. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. A2-Nets: Double attention networks. *arXiv* **2018**, arXiv:1810.11579, 352.
45. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-maximization attention networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 9166–9175. [CrossRef]
46. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv* **2019**, arXiv:1908.08681.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015. [CrossRef]
48. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
49. Loshchilov, I.; Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017—Conference Track Proceedings, Toulon, France, 24–26 April 2017.