electronics

MDPI

Article

Hyperspectral Image Classification Based on Dual-Scale Dense Network with Efficient Channel Attentional Feature Fusion

Zhongyang Shi, Ming Chen * and Zhigao Wu

Key Laboratory of Fisheries Information, Ministry of Agriculture and Rural Affairs, College of Information Science, Shanghai Ocean University, Shanghai 201306, China; m210901470@st.shou.edu.cn (Z.S.); m200711473@st.shou.edu.cn (Z.W.)

* Correspondence: mchen@shou.edu.cn

Abstract: Hyperspectral images (HSIs) have abundant spectral and spatial information, which shows bright prospects in the application industry of urban–rural. Thus, HSI classification has drawn much attention from researchers. However, the spectral and spatial information-extracting method is one of the research difficulties in HSI classification tasks. To meet this tough challenge, we propose an efficient channel attentional feature fusion dense network (CA-FFDN). Our network has two structures. In the feature extraction structure, we utilized a novel bottleneck based on separable convolution (SC-bottleneck) and efficient channel attention (ECA) to simultaneously fuse spatial–spectral features from different depths, which can make full use of the dual-scale shallow and deep spatial–spectral features of the HSI and also significantly reduce the parameters. In the feature enhancement structure, we used 3D convolution and average pooling to further integrate spatial–spectral features. Many experiments on Indian Pines (IP), University of Pavia (UP), and Kennedy Space Center (KSC) datasets demonstrated that our CA-FFDN outperformed the other five state-of-the-art networks, even with small training samples. Meanwhile, our CA-FFDN achieved classification accuracies of 99.51%, 99.91%, and 99.89%, respectively, in the case where the ratio of the IP, UP, and KSC datasets was 2:1:7, 1:1:8, and 2:1:7. It provided the best classification performance with the highest accuracy, fastest convergence, and slightest training and validation loss fluctuations.

Keywords: hyperspectral image classification; dense network; separable convolution; efficient channel attention; feature fusion

Citation: Shi, Z.; Chen, M.; Wu, Z. Hyperspectral Image Classification Based on Dual-Scale Dense Network with Efficient Channel Attentional Feature Fusion. *Electronics* **2023**, *12*, 2991. https://doi.org/10.3390/electronics12132991

Academic Editor: Byung Cheol Song

Received: 1 June 2023
Revised: 3 July 2023
Accepted: 5 July 2023
Published: 7 July 2023

1. Introduction

Hyperspectral images (HSIs) are a particular type of remote sensing image with abundant spectral and spatial information [1], which can be studied in many fields, including urban vegetation cover monitoring [2], water detection [3], agricultural resource detection [4], and environmental protection [5], etc. [6–8]. Recently, HSI classification tasks have been the focus of HSI research [9,10]. However, the over-redundancy of spectral band information makes it hard to extract fine features, which has been a challenge for feature extraction. Traditional feature extraction methods, such as support vector machine (SVM) [11,12] and multinomial logistic regression (MLR) [13,14], have been developed to build pixel-wise-based classifiers for analyzing the HSI. Although enough spectral features can be extracted using these methods, the acquired classification maps are still noisy or blurry. To this end, denoising, deblurring, super-resolution, and feature fusion strategies were proposed to solve these issues. Recently, deep learning algorithms based on convolutional neural networks (CNNs) [15] were found to have better performance in feature fusion [16]. CNN can extract spatial information without destroying the original spatial structure [17]. Specifically, deep learning algorithms have experienced a process from 1D-CNN [18] and 2D-CNN [19–21] to 3D-CNN [22]. In comparison, 3D-CNN can extract spatial–spectral features by using 3D convolution, which makes full use of the 3D data information of the HSI compared with the 1D-CNN and 2D-CNN.

However, 3D-CNN suffers from overfitting and degradation. The residual network (ResNet) [23] and the dense network (DenseNet) [24] were proposed to solve these problems. For example, Zhong et al. [25] proposed a spectral–spatial residual network (SSRN). They designed a consequent spatial–spectral residual block to sequentially learn the HSI's discriminative features, effectively improving the classification accuracy. The disadvantage, however, is an exorbitantly long training time. Inspired by SSRN, Wang et al. [26] designed a consequent spatial–spectral dense block based on DenseNet to improve feature reuse, which also helped to achieve better performance while reducing training time. To combine the advantages of the dense network and the residual network, Tu et al. [27] proposed a residual dense and dilated convolution network (RDDC-3DCNN) with both residual and dense blocks to fuse the features hierarchically, and the accuracy was further improved. When facing the small sample issue, unsupervised and semi-supervised networks were proposed, such as the Conv–Deconv network [28], generative adversarial networks (GANs) [29], graph convolutional networks (GCNs) [30], and robust self-ensembling network (RSEN) [31], etc. These networks effectively improve the accuracy of HSI classification tasks.

The above studies show that networks with residual or dense structures can realize feature fusion from layer to layer and, thus, obtain finer features. But, as convolutional layers increase, most fine features tend to be reduced or even lost. To solve these issues, a lot of advanced techniques, such as feature fusion and attention mechanisms, have been applied to HSI classification tasks. Zhang et al. [32] proposed a multi-scale dense network (MSDN) with a dense network as the backbone. The feature maps from low scale, medium scale, and high scale were used for feature fusion; this method ensured accuracy while improving the convergence speed, but the network has a vast number of parameters as well as a long training time. In addition, in the procedure of extracting spatial–spectral features, they did not find clear semantics for these feature maps. In [33], a novel multi-scale dense network (MSDN-SA) was proposed, and the spectral-wise attention mechanism was employed in the field of HSI classification for the first time.

Subsequently, attention mechanisms have been successfully practiced and developed in HSI classification over the years. Li et al. [34] proposed a 3D-SE-DenseNet based on a Squeeze-and-Excitation network (SENet) [35], which enhanced the ability to extract spectral features by automatically learning to construct an SENet after each dense block. In [17], a double-branch multi-attention network (DBMA) with the convolutional block attention module (CBAM) [36] was proposed, which will significantly reduce the interference between two different kinds of features. Based on DBMA and the adaptive self-attention mechanism [37], Li et al. [38] proposed a double-branch dual-attention mechanism network (DBDA). The DBDA framework uses less training time while obtaining higher accuracy than DBAM. Qing et al. [39] employed ECANet [40] in their multi-scale residual convolutional neural network (MRA-Net) to fully exploit the core components obtained from the principal component analysis (PCA) [41] technology, which successfully helped improve classification accuracy. Following that, Qing et al. [42] again proposed a 3D self-attention [43] multi-scale feature fusion network (3DSA-MFN), which can make full use of the contextual information of the HSI.

Very recently, the self-attention-based transformer [44] has been widely used in HSI classification, which can better process sequential data. For example, the spatial–spectral transformer (SST) [45] was proposed to solve the problem of gradient vanishing. In [46], the SpectralTransformer was proposed to effectively process the sequence attributes of spectral features. In [47], a spectral–spatial feature tokenization transformer (SSFTT) method was proposed to capture high-level semantic features. However, a transformer is relatively weak in discriminating local features [48]. In addition, few of the networks above can better balance the convergence performance and classification accuracy, and these networks tend to have large loss fluctuations in the training process.

Therefore, in this paper, inspired by the advanced MSDN network and [49], we propose an efficient channel attentional feature fusion dense network for HSI classification

to better fuse features of inconsistent semantics and scales. To conclude, three major contributions have been made to this study:

- We propose an efficient channel attentional feature fusion dense network (CA-FFDN) based on DenseNet and ECA. Our network has two main structures: feature extraction structure and feature enhancement structure. Principal component analysis (PCA) technology is applied to ensure the utilization of effective spectra and reduce noise interference.
- We employ the latest modified DenseNet as the backbone, which outstandingly reduces the parameter and training time in the network compared with MSDN. Meanwhile, an efficient attention mechanism is introduced to realize attentional feature fusion at two scales of the input and output layers, which suppresses the loss of spatial–spectral features and accelerates the convergence speed of the network.
- The proposed network has state-of-the-art classification results in comparison experiments with five advanced networks under the same experimental environment on three open-source datasets.

The rest of the paper is organized as follows: Section 2 introduces the proposed framework. Section 3 details the experimental results and analysis. Finally, Section 4 concludes the paper.

2. The Proposed Framework

The main framework of CA-FFDN is shown in Figure 1. First, the input HSI cube was operated with data normalization and PCA to depress the data variability and band noise. The HSI cube was then segmented into small cubes centered on labeled pixels and sent to the CA-FFDN. On the one hand, considering the problems that the increasing layers of the network will likely cause overfitting and gradient vanishing, we employed a densely connected network as the backbone of CA-FFDN. On the other hand, the CA-FFDN aims to extract more discriminative features while achieving the attentional feature of the HSI small cubes. Following that, CA-FFDN is divided into two structures: feature extraction structure and feature enhancement structure. The feature extraction structure consists of an input layer and output layer, which applies SC-bottleneck-based dense connections to extract more discriminate spatial and spectral features with the increasing depth of CA-FFDN while effectively reducing the network parameters. In addition, CA-FFDN concentrates channel information twice and performs attentional feature fusion with the ECA mechanism in the ECA-FF module. The number of channels of HSI also equals the number of the convolution kernel. As the depth of the feature extraction structure increases, the feature extraction structure will extract finer features owing to the skip connection and ECA mechanism. The feature enhancement structure is designed with two convolution operations and one averaging-pooling operation to further extract and integrate the spatial–spectral features. The kernel size is set at $3 \times 3 \times 3$, and the features after pooling are flattened. Finally, we obtain the classification maps through a fully connected layer that uses the softmax activation function.

2.1. Dense Network Based on SC-Bottleneck

In a recent paper, Wang et al. [50] applied separable convolution [51] to improve the bottleneck structure of dense networks, which is named SC-bottleneck in this paper. As shown in Figure 2, the number of channels of the input layer feature map is $l \times k$, and k is the growth rate. We use $1 \times 1 \times 1$ convolution to compress the channels to $4k$. After batch normalization (BN) and ReLU activation function, the traditional $3 \times 3 \times 3$ convolution operation was divided into two steps: $3 \times 3 \times 1$ and $1 \times 1 \times 3$ convolutions were used in depthwise convolution and pointwise convolution, respectively, to extract the spatial features and the spectral features of HSI. The number of the output feature map is k. The SC-bottleneck used separable convolution to further reduce the parameters and ensure the effective extraction of spatial and spectral features of HSI, leading to significant improvements in network performance.

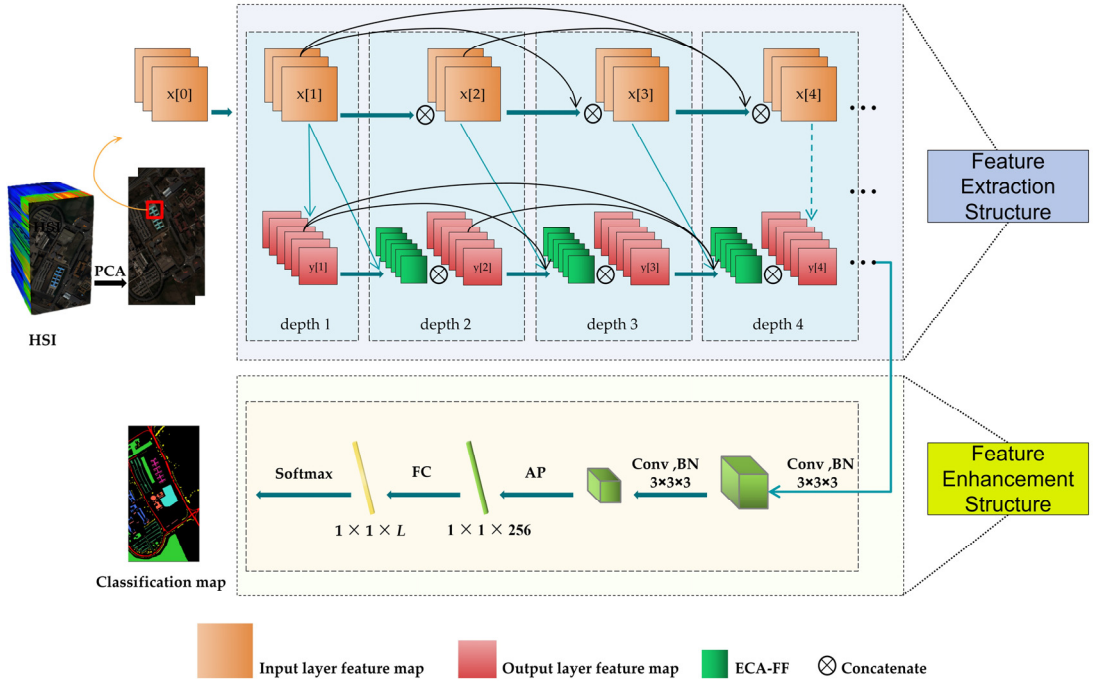


Figure 1. Framework of CA-FFDN.

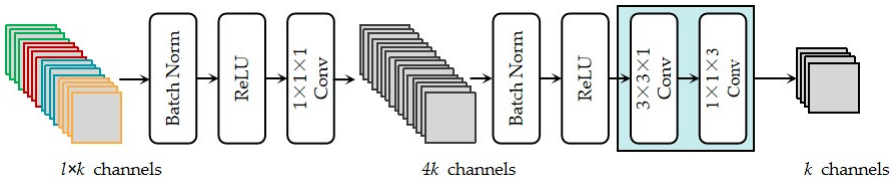


Figure 2. SC-bottleneck structure.

HSI maintains abundant information on the land cover. The general method of extracting finer features is to deepen the network's depth or increase its width, but it tends to show the phenomenon of overfitting. However, dense networks are an excellent solution to overfitting. The principle of the dense network is to reduce the loss in the process of input to output. Through layer-by-layer connection, the output features of all previous layers are superimposed before the information of each layer to realize feature reuse, which reduces the negative impact of gradient vanishing and overfitting and dramatically reduces the parameters. The process is expressed as Equation (1):

$$x_i = H_i(x_0 + x_1 + \cdots + x_{i-1}) \tag{1}$$

where x_i is the output result of the i-th layer, H_i represents the convolution operation of the i-th layer, including Convolution, BN, and Nonlinear activation, and we use ReLU as the activation function in this paper.

2.2. ECA Mechanism

Numerous experimental studies have shown that adding an attention mechanism into the networks will improve efficiency. An efficient channel attention network (ECANet) [40] is a kind of channel attention network, which is a further improvement in the Squeeze and Excitation network (SENet). As shown in Figure 3, the SENet contains two fully connected layers. The channels will have an operation of dimensionality reduction between the fully connected layers by setting the compression ratio, which affects the efficiency of the network, while the fully connected layers integrate the dependencies between all channels, which also affects the efficiency of the network.

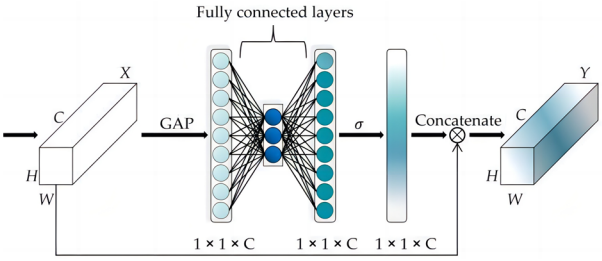


Figure 3. Structure of SENet.

Instead, ECANet considers the information of each channel within its k fields to capture the cross-channel interaction information. In Figure 4, it is assumed that the input of ECANet is $X \in R^{W \times H \times C}$. First, the Global Average Pooling (GAP) is applied to compress the global information into one channel $1 \times 1 \times C$, amplifying the receptive field. Equation (2) yields the results after GAP:

$$g(X) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} X_{ij} \tag{2}$$

where $g(X)$ represents the Global Average-Pooling operation.

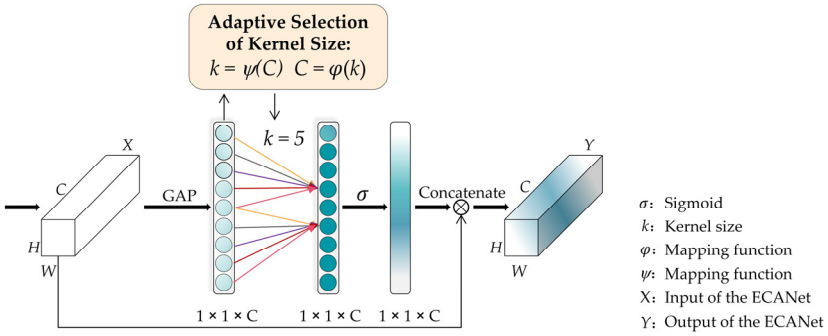


Figure 4. Structure of ECANet.

The cross-channel information interaction is performed using one-dimensional (1D) convolution with a convolution kernel size of k, where k represents the coverage of the channel information interaction. The larger the channel dimension C, the larger the coverage. Then, an adaptive function is introduced to avoid man-made errors by automatically selecting the size of the one-dimensional convolutional kernel to determine the value of k. There is a mapping relationship between k and the channel dimension C, as follows:

$$C = \varphi(k) \tag{3}$$

Then, approximate the mapping function in ECANet using the exponential function φ, as follows:

$$\varphi(k) = \exp(\gamma k - b) \tag{4}$$

Since the number of channels is usually set to an integer power of 2, the mapping relation can be further computed as:

$$\varphi(k) = 2^{(\gamma k - b)} \tag{5}$$

Given a channel number C, the size of the 1D convolution kernel can be determined using Equation (6):

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{6}$$

where γ and b are set to 2 and 1, respectively; odd indicates that the convolution kernel size only takes odd values.

Assuming that y_i is the compressed feature of c_i channels after GAP, y_i^j denotes the i-th output value of the channel adjacent to the j-th channel. Q_i^j denotes the set of k domain channels and P denotes y_i, the set of feature map channels for weighting. Then, the locally weighted weights w_i of y_i are computed as:

$$\begin{cases} y_i = g(c_i) & c_i \in P \\ w_i = \sigma\left(\sum_{j=1}^{k} \omega^j y_i^j\right) & y_i^j \in Q_i^j \end{cases} \tag{7}$$

where σ denotes the Sigmoid activation function.

Finally, the channel attention weight w is computed as:

$$w = \sigma(C1D_k(y)) \tag{8}$$

where $C1D$ denotes 1D convolution, and y is the compressed feature.

To summarize, ECANet adaptively selects 1D convolutional kernels via Equations (3)–(6), achieves weight sharing through Equation (7), and obtains the channel attention to compressed features via Equation (8).

2.3. Feature Extraction Structure

The feature extraction structure is divided into an input layer and an output layer, and the main part of the feature extraction structure is shown in Figure 5. $x[i]$ represents the feature map of the input layer with a growth rate of 6. $y[i]$ represents the feature map of the output layer with a growth rate of 12.

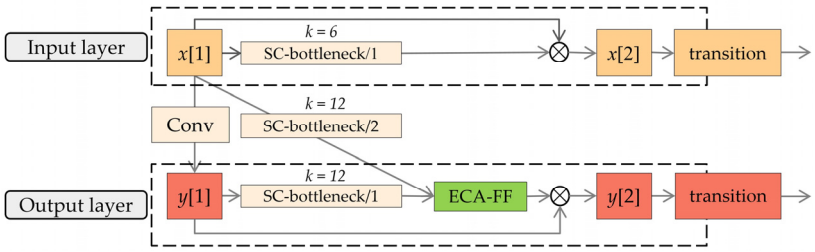


Figure 5. Feature extraction structure.

First, x[1] is convolved (Conv) by 64 convolution kernels of size $3 \times 3 \times 3$ to obtain $y[1]$. Then, two feature maps are obtained from $x[1]$ and $y[1]$. The second feature map is computed through the SC-bottleneck using $x[1]$ as the input feature map with a convolution stride size 2. Following that, they are sent to the efficient channel attentional feature fusion (ECA-FF) module shown in Figure 6. As the depth of the feature extraction structure increases, the number of ECA-FF modules increases, and the finer the extracted features of HSI, significantly reducing the influence of useless channel information. Equation (9) shows the general procedure:

$$y[i+1] = E[C_1(y[i])] + E[C_2(x[i])] \tag{9}$$

where $C_1(\cdot)$ and $C_2(\cdot)$ denote the SC-bottleneck convolution processes using convolution stride size 1 and 2, respectively. $E[\cdot]$ denotes the ECA operation, and $+$ denotes concatenating the channel dimension.

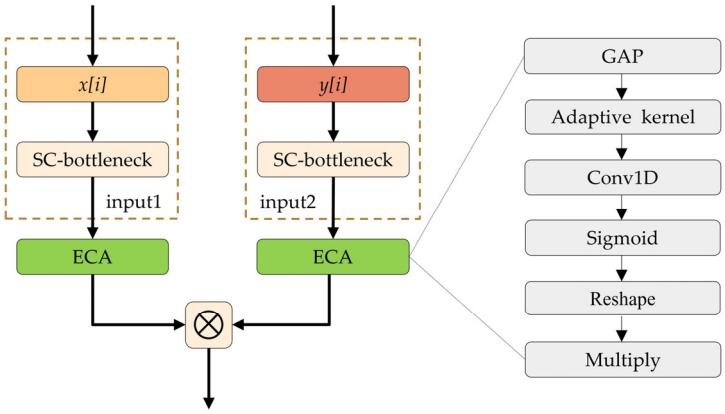


Figure 6. ECA-FF module.

Since the parameters inevitably increase gradually as the depth of CA-FFDN increases, a transition layer was added at the back of the feature maps of the input and output layers. The number of feature channels was reduced using a $1 \times 1 \times 1$ 3D convolutional kernel. Experiments proved that the transition layer greatly reduced the parameters and accelerated the convergence speed of the network. Moreover, we proposed concatenating the convolved feature map in the last layer of the input layer with the feature map in the output layer of the same layer for the course of improving the HSI channel information utilization.

2.4. Feature Enhancement Structure

As shown in Figure 7, the output of the feature extraction structure is the input of the feature enhancement structure. The input feature map size is $7 \times 7 \times 15$, and the number of channels C varies with the depth of the feature extraction structure. We employed two times $3 \times 3 \times 3$ convolution. However, the number of convolution kernels n significantly impacts the network performance of CA-FFDN. Too few convolution kernels will reduce the classification accuracy of the network, while too many will substantially increase the training time and, thus, reduce the network's performance. Therefore, we tested 80, 96, 100, and 128 convolution kernels. The results showed that 128 convolution kernels would lead to the highest accuracy. Moreover, BN and ReLU activation functions were added after each convolution operation to ensure the convergence speed of the CA-FFDN and prevent overfitting. Finally, the spatial–spectral features were integrated using the average-pooling layer, and the classification maps were obtained using the softmax-based fully connected layer after the Flatten operation.

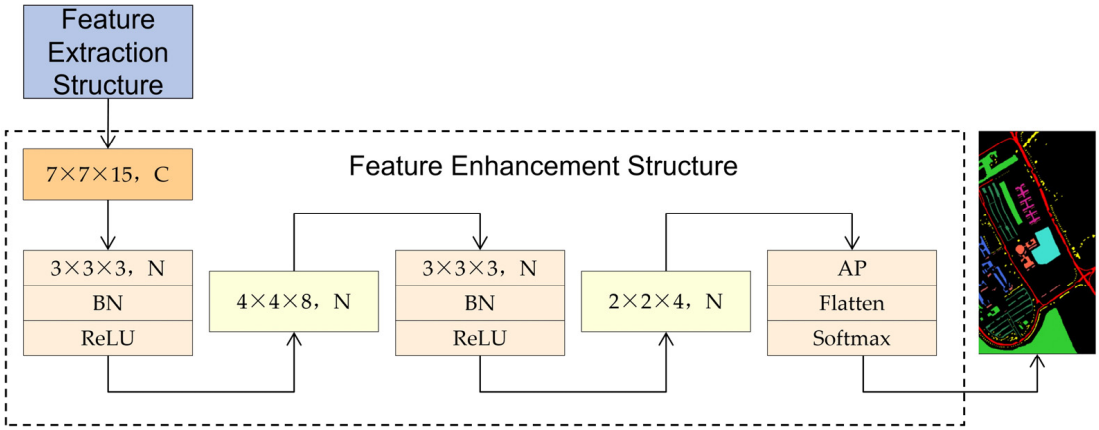


Figure 7. Feature enhancement structure.

3. Experimental Results and Analysis

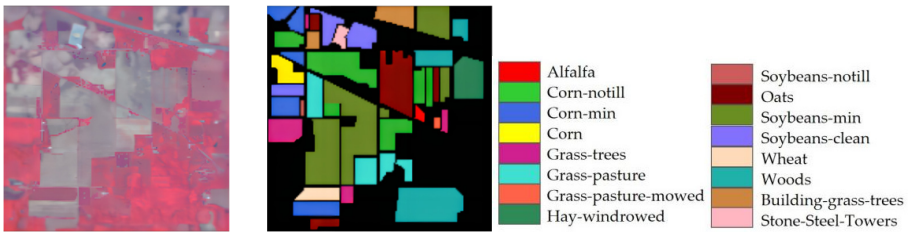
The hardware environment for all experiments utilizes an Intel(R) Core(TM) i7-9700K CPU @ 3.60 GHz processor with 64 GB of RAM and NVIDIA GeForce RTX 2080Ti GPU. The software environment is based on the deep learning framework of Tensorflow-gpu for Windows 10, utilizing the Pycharm2020 platform with Python 3.6 compiler.

3.1. Experimental Dataset

Three open-source hyperspectral datasets, Indian Pines (IP), Pavia of University (UP), and Kennedy Space Center (KSC) [52], were selected as experimental subjects. We divided the datasets into the training set, validation set, and testing set after random shuffling, and it is worth mentioning that we followed [34] to set the ratio of the samples. 20%:10%:70% for the IP and KSC datasets and 10%:10%:80% for the UP dataset. Last, but not least, overall accuracy (OA), average accuracy (AA), and kappa coefficient (K) are used for quantitative analysis of the experimental results. Higher metric values indicate that the network is more capable of classifying [53]. Table 1 shows the specific parameter settings of the datasets, and the false color maps of the IP, UP, and KSC datasets and their ground-truth maps are shown in Figures 8–10, respectively.

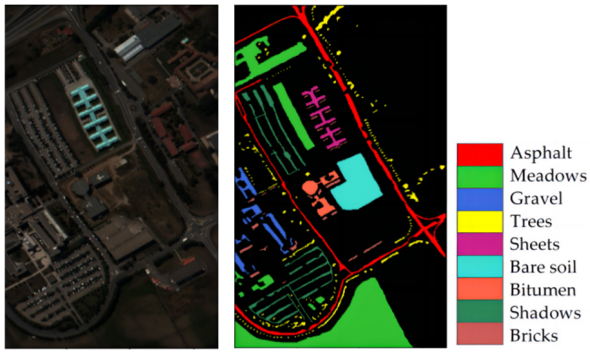
Table 1. Parameter setup of the datasets.

Parameter	Dataset		
	IP	**UP**	**KSC**
Sensor	AVIRIS	ROSIS	AVIRIS
Year of data acquisition	1992	2001	1996
Spectrum range/nm	400~2500	430~860	400~2500
Spatial resolution/m	20	1.3	18
Pixel resolution	145 × 145	610 × 340	512 × 614
Band	200	103	176
Land-cover	16	9	13
Total sample	10,249	42,776	5211



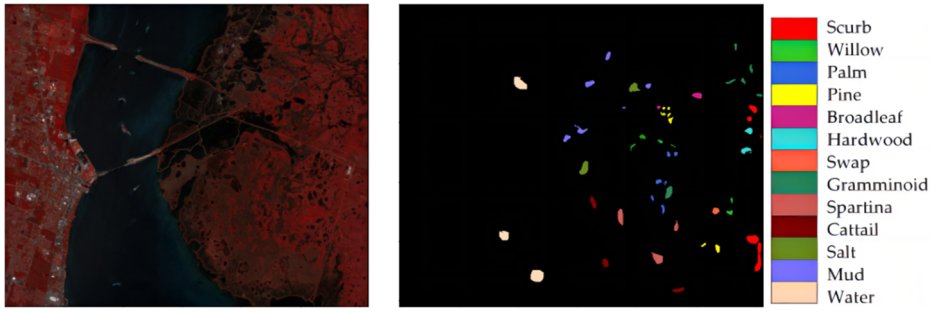
(**a**) (**b**)

Figure 8. Indian Pines dataset: (**a**) false color map; (**b**) ground-truth map.



(**a**) (**b**)

Figure 9. University of Pavia dataset: (**a**) false color map; (**b**) ground-truth map.



(**a**) (**b**)

Figure 10. Kennedy Space Center dataset: (**a**) false color map; (**b**) ground-truth map.

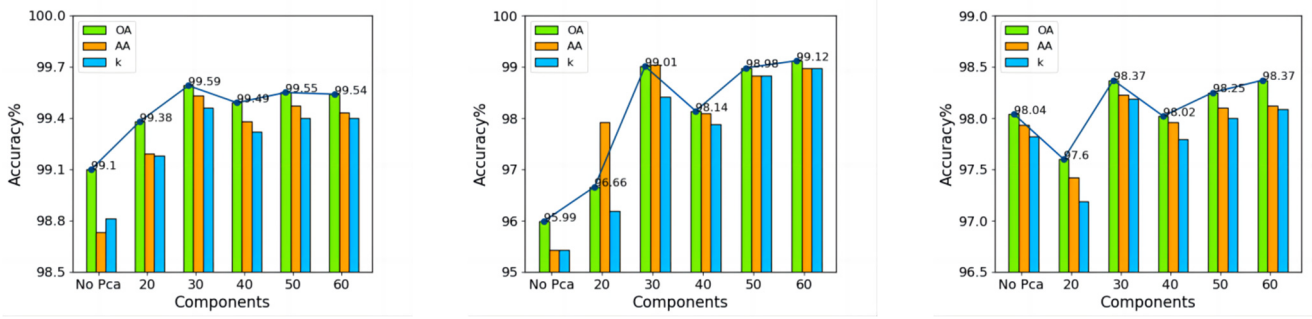
3.2. Experimental Setting

In our experiment, the batch size was set to 16, the RMSprop optimizer [54] was selected to optimize the training loss, and the number of training iterations was set to 100, saving the best model for each iteration. We employed the grid search [55,56] method to choose the best learning rate, and the learning rate was set to 0.01, 0.001, 0.003, 0.0003, 0.0005, and 0.00005, respectively. The results showed that the optimal learning rate was 0.0003 on the three datasets.

3.2.1. Effect of Principal Components

In the PCA test, the number of principal components significantly impacts the classification results, and the components were set to 20 to 60; the spectral dimensions after PCA were selected at intervals of 10 to conduct five sets of experiments on the three datasets. Experimental results without PCA were used as control experiments.

We can see from Figure 11 that as the number of principal components increases, the values of three metrics on the three datasets continue to rise, reaching the highest values when the number of principal components was 30 and then decreasing. On the IP dataset, when the number of principal components was 60, the OA reached 99.12%, which was 0.11% higher than the number of principal components at 30. However, the number and amount of parameters, training, and testing time increased along with the principal components. Additionally, more principal components will cause noise and reduce the classification accuracy of low-resolution hyperspectral images. The experimental accuracy without PCA was the lowest on the IP and UP datasets, and the training time was the longest. Considering the above, we set the number of principal components to 30.



(**a**) (**b**) (**c**)

Figure 11. Classification results under PCA test: (**a**) IP dataset; (**b**) UP dataset; (**c**) KSC dataset.

3.2.2. Effect of Different Spatial Size Inputs

The spatial size of the input sample influences the classification accuracy of the HSI greatly. In order to choose the best spatial size of CA-FFDN, five sets of experiments were conducted with spatial sizes of 9×9, 11×11, 13×13, 15×15, and 17×17. The classification results are shown in Figure 12. For the IP dataset, Figure 12a shows that it

reached the highest AA of 99.47% at a spatial size of 13 × 13 and the highest OA of 99.42% at a spatial size of 15 × 15, and then all values of the three metrics started to decrease. For the UP dataset, as shown in Figure 12b, the accuracy reached close to 99% at a spatial size of 13 × 13. It reached the highest OA, AA, and K values at 15 × 15, but its running time also significantly increased. For the KSC dataset, Figure 12c shows that the classification accuracy started to fall after 13 × 13. Considering the above, we chose the spatial size of 13 × 13 as the input for CA-FFDN.

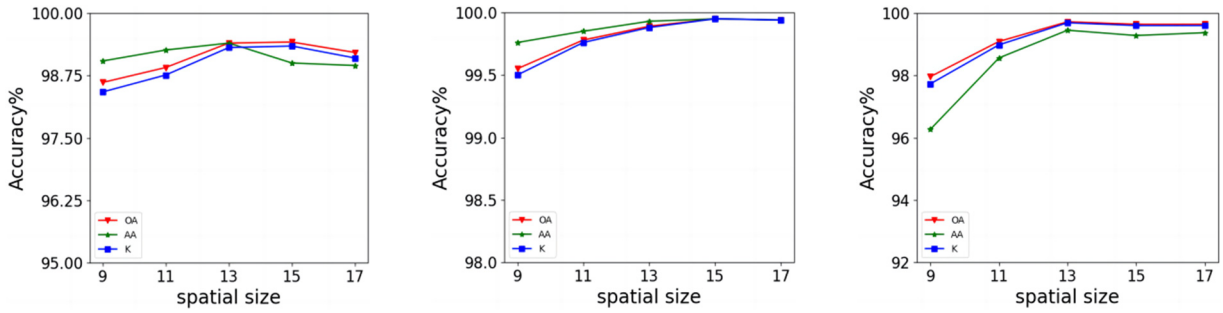


Figure 12. Accuracy under different spatial size inputs: (**a**) IP dataset; (**b**) UP dataset; (**c**) KSC dataset.

3.2.3. Effect of the Feature Extraction Structure Depth

The depth of the feature extraction structure also dramatically impacts the accuracy. As shown in Figure 13, the classification results improved with the increasing depth of the feature extraction structure. OA reached the highest on three datasets when the depth was 4, which was 99.51%, 99.91%, and 99.89%, respectively. For all the datasets, the accuracies began to fall after depth 4. As shown in Figure 12b,c, the classification accuracies on UP and KSC datasets are unstable. When the depth was 5, the OA decreased by 0.39% and 0.11% but improved by 0.14% and 0.04% when the depth was 6. Thus, we chose the optimal depth of the feature extraction structure of CA-FFDN to be 4.

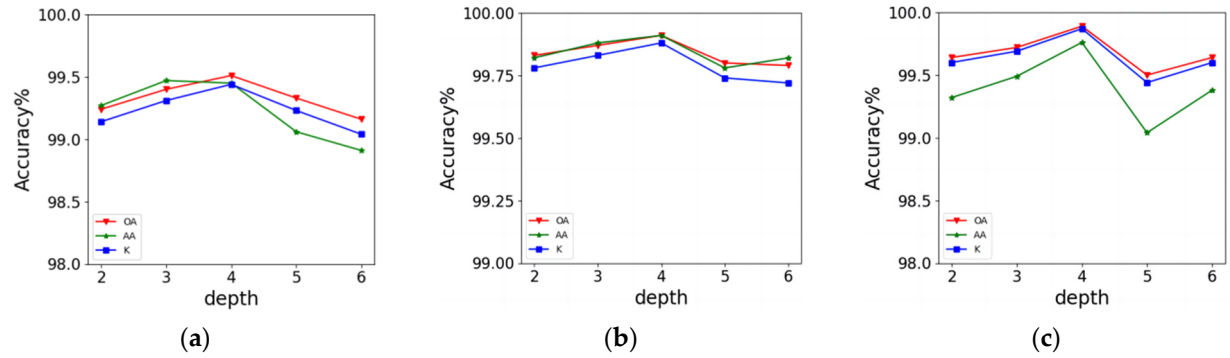


Figure 13. Accuracy under different feature extraction structure depths: (**a**) IP dataset; (**b**) UP dataset; (**c**) KSC dataset.

3.2.4. Ratios of Training Dataset

The bigger the division ratio of the training dataset, the higher the classification accuracy. Different ratios of the training dataset are discussed in Table 2 to select the optimal one. For all datasets, OA and training time increased along with the ratio of the training set, validation set, and testing set. For the IP, UP, and KSC datasets, OA reached 99% when the ratio was at 2:1:7, 1:1:8, and 2:1:7, respectively, which already met with the expected results of the experiment. In addition, OA reached 99.9% when the ratio was 5:1:4 for the IP dataset and 3:1:6 for the UP dataset. However, then came the increase in training time cost. Training the network took almost double the time when the training samples

rose by 10%. Finally, we selected a ratio of 2:1:7 for the IP and KSC datasets and 1:1:8 for the UP dataset.

Table 2. OA, training time, and test time under different training dataset ratios.

	IP (13 × 13 × 30; Depth = 4)			UP (13 × 13 × 30; Depth = 4)			KSC (13 × 13 × 30; Depth = 4)		
	OA	**Train (s)**	**Test (s)**	**OA**	**Train (s)**	**Test (s)**	**OA**	**Train (s)**	**Test (s)**
1:1:8	95.80	428.3	6.4	99.91	1868.4	25.6	98.84	253.8	3.2
2:1:7	99.51	908.5	4.8	99.96	3642.5	22.8	99.89	438.3	2.8
3:1:6	99.74	1446.7	2.6	99.98	5298.7	19.7	99.93	629.3	2.4
4:1:5	99.82	1781.5	1.4	100	6824.6	16.4	100	833.8	2.0
5:1:4	99.94	2275.2	0.8	100	8541.8	13.9	100	996.9	1.5

3.3. Ablation Experiment

To verify the effectiveness of SC-bottleneck (S), ECA-FF (E), and transition layer (T), we performed six groups of ablation experiments. As shown in Table 3, the ECA was removed from the ECA-FF module in Group 1; the SC-bottleneck was not used in Group 2; the transition layer was not applied in Group 3; both the ECA mechanism and the SC-bottleneck were removed in Group 4, but the transition layer was retained; the transition layer was removed in Group 5 based on Group 4; and the sixth group of experiments is the proposed network in this paper. Each experiment was carried out ten times, and the average value was taken.

Table 3. Classification results under six different groups of methods.

	Method			**Dataset**								
				IP (20%)			**UP (10%)**			**KSC (20%)**		
Group	**S**	**E**	**T**	**OA (%)**	**AA (%)**	**K (%)**	**OA (%)**	**AA (%)**	**K (%)**	**OA (%)**	**AA (%)**	**K (%)**
1	√		√	99.36	99.36	99.27	99.89	99.89	99.85	99.45	99.33	99.38
2		√	√	99.25	99.22	99.14	99.90	99.89	99.87	99.72	99.40	99.69
3	√	√		99.33	99.28	99.23	99.79	99.80	99.73	99.53	99.01	99.47
4			√	99.33	99.17	99.23	99.87	99.85	99.82	99.67	99.37	99.63
5				99.21	98.99	99.10	99.89	99.88	99.88	99.67	99.50	99.63
6	√	√	√	99.51	99.45	99.44	99.91	99.91	99.88	99.89	99.76	99.87

The experimental results on three datasets of Groups 1, 2, and 3 were compared with the results of Group 6, respectively, showing that the SC-bottleneck, ECA-FF module, and transition layers can improve the classification results. And the comparison of the results between the fourth and sixth groups shows that networks without transition layers decrease in accuracy, which verifies the applicability of adding transition layers in CA-FFDN. For the IP dataset, the fifth group of experiments achieved the worst classification accuracy, with a 0.3% decrease from the best result. For the UP dataset, Group 3 achieved the worst classification result. Still, in terms of the high spatial resolution of the UP dataset and the small number of mixed pixels, the classification accuracy reached more than 99.7%. Beyond that, the effectiveness of the ECA-FF module and the SC-bottleneck can be illustrated based on the comparison with Group 5. For the KSC dataset, the network lacking the ECA mechanism achieved the worst classification accuracy, and the OA was reduced by 0.44% compared with the best setting, and the ECA mechanism will significantly improve the classification accuracy of the KSC dataset.

To further prove that our attentional feature fusion strategy is more robust than other attention mechanisms. We employed channel attention mechanism (CAM), spatial attention mechanism (SAM), CBAM, and SENet to replace the ECANet in the ECA-FF module, respectively. It is observed from Table 4 that SENet obtained the worst results among all attention mechanisms, and the reason was that the process of dimensionality reduction in

the fully connected layers brought side effects to the extraction of channel information. All attentional feature fusion methods' OA reached 99%, reflecting our network's robustness. Meanwhile, attentional feature fusion based on ECANet outperformed other attention methods because ECANet can capture the cross-channel information to make full use of the semantic information.

Table 4. Classification results under different attention mechanisms.

Method	Dataset								
	IP (20%)			UP (10%)			KSC (20%)		
	OA (%)	AA (%)	K (%)	OA (%)	AA (%)	K (%)	OA (%)	AA (%)	K (%)
CAM [36]	99.20	99.28	98.98	99.78	99.78	99.71	99.61	99.26	99.57
SAM [36]	99.27	98.99	99.17	99.83	99.83	99.79	99.47	99.04	99.41
CBAM [36]	99.10	99.02	98.98	99.84	99.81	99.77	99.56	99.06	99.51
SENet [35]	99.00	99.09	98.87	99.66	99.62	99.60	99.46	99.11	99.41
ECANet [40]	99.51	99.45	99.44	99.91	99.91	99.88	99.89	99.76	99.87

3.4. Comparative Analysis of Classification Results

Five advanced networks were selected for comparative analysis, including 3D-CNN [57], HybridSN [58], 3D-SE-DenseNet [34], MDSSAN [50], and MSDN [32]. The 3D-CNN contains three 3D convolutional layers and two global-pooling layers. Meanwhile, the dropout strategy is added to prevent overfitting. HybridSN is a 3D and 2D convolution combined network, which is more efficient than simple 3D-CNN networks. In the 3D-SE-DenseNet, each dense block is followed by an SENet structure, and both the dense blocks and dense layers were set to 3 in this paper. MDSSAN applies separable convolution in the bottleneck to reduce the training parameters. The depth of MSDN remained the same as ours. In order to ensure the fairness of the experiments, all experimental data were measured under the same environment, and the sample division ratio of different datasets, as well as input size for each network, were the same as ours. Furthermore, the optimal parameter settings of each network are consistent with those of the references. Tables 5–7 show the classification results of the experiments, and Figures 14–16 present the classification maps on three datasets.

Table 5. Classification results of each method for IP dataset with 20% training samples.

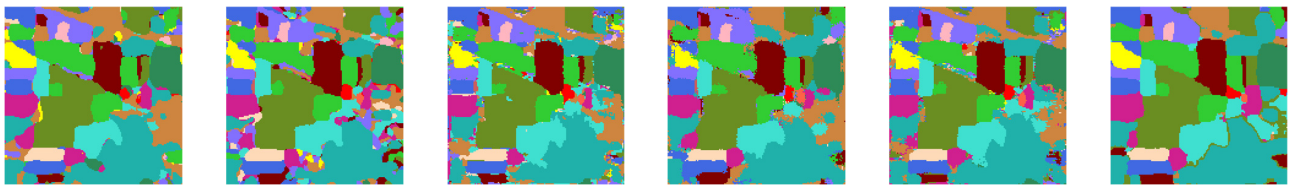
No.	3D-CNN	HybridSN	3D-SE-DenseNet	MDSSAN	MSDN	Proposed
1	85.71	100.00	83.33	94.59	94.44	97.22
2	99.59	99.41	96.47	98.40	99.29	98.43
3	92.39	99.15	98.44	97.91	98.28	100.00
4	99.36	100.00	94.90	99.31	90.76	100.00
5	95.27	99.31	97.71	98.27	100.00	100.00
6	98.81	98.03	99.80	99.61	99.80	100.00
7	100.00	100.00	100.00	100.00	100.00	100.00
8	99.09	100.00	98.49	100.00	100.00	100.00
9	100.00	100.00	100.00	100.00	100.00	100.00
10	99.54	98.96	99.39	99.10	99.40	100.00
11	98.07	99.65	98.89	99.11	99.58	99.64
12	94.30	92.87	97.53	94.57	98.30	99.28
13	100.00	100.00	99.28	94.52	99.28	100.00
14	98.64	95.17	99.43	99.54	99.88	99.54
15	99.26	98.61	97.50	96.81	97.16	98.56
16	98.48	98.30	91.89	100.00	97.01	98.55
OA(%)	97.78 ± 0.24	98.37 ± 0.57	98.25 ± 0.74	98.53 ± 0.46	99.06 ± 0.67	99.51 ± 0.12
AA(%)	97.41 ± 0.31	98.71 ± 0.61	97.06 ± 0.62	98.23 ± 0.23	98.32 ± 0.72	99.47 ± 0.25
K × 100	97.47 ± 0.68	98.14 ± 0.32	98.01 ± 0.75	98.33 ± 0.25	98.93 ± 0.34	99.44 ± 0.14

Table 6. Classification results of each method for UP dataset with 10% training samples.

No.	3D-CNN	HybridSN	3D-SE-DenseNet	MDSSAN	MSDN	Proposed
1	99.31	99.58	97.75	99.26	99.37	99.84
2	99.60	99.59	99.97	99.96	99.98	99.95
3	97.30	97.88	97.83	99.87	98.05	100.00
4	99.40	98.13	99.79	99.08	100.00	100.00
5	99.72	100.00	98.44	99.71	100.00	100.00
6	99.82	99.92	99.92	99.97	99.90	100.00
7	100.00	100.00	99.39	99.81	98.66	100.00
8	95.30	94.72	98.74	97.92	98.48	99.52
9	95.72	100.00	99.20	100.00	99.86	99.86
OA(%)	99.00 ± 0.82	99.04 ± 0.19	99.31 ± 0.15	99.60 ± 0.18	99.61 ± 0.17	99.91 ± 0.03
AA(%)	98.46 ± 0.54	98.87 ± 0.26	99.00 ± 0.37	99.51 ± 0.27	99.37 ± 0.24	99.91 ± 0.02
K × 100	98.67 ± 0.64	98.73 ± 0.11	99.09 ± 0.28	99.47 ± 0.16	99.49 ± 0.14	99.88 ± 0.07

Table 7. Classification results of each method for KSC dataset with 20% training samples.

No.	3D-CNN	HybridSN	3D-SE-DenseNet	MDSSAN	MSDN	Proposed
1	100.00	100.00	100.00	97.24	100.00	100.00
2	94.79	100.00	99.29	99.37	100.00	100.00
3	82.62	93.61	96.70	100.00	96.70	100.00
4	97.93	92.61	93.25	95.45	100.00	98.82
5	100.00	96.15	100.00	100.00	98.18	98.18
6	100.00	100.00	100.00	98.65	100.00	100.00
7	96.38	95.23	100.00	100.00	100.00	100.00
8	92.54	97.70	100.00	99.33	99.28	100.00
9	100.00	100.00	93.06	97.91	94.94	100.00
10	100.00	98.95	96.25	100.00	99.64	100.00
11	100.00	100.00	99.32	100.00	100.00	100.00
12	99.70	100.00	99.41	99.12	100.00	100.00
13	100.00	100.00	99.69	100.00	100.00	100.00
OA(%)	97.88 ± 0.76	98.81 ± 0.13	98.24 ± 0.75	98.92 ± 0.75	99.14 ± 0.47	99.89 ± 0.09
AA(%)	97.23 ± 0.64	98.02 ± 0.81	98.23 ± 0.62	99.00 ± 0.63	99.13 ± 0.54	99.76 ± 0.12
K × 100	97.64 ± 0.66	98.68 ± 0.23	98.04 ± 0.48	99.00 ± 0.67	99.05 ± 0.46	99.87 ± 0.08



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 14. Classification maps for the IP dataset: (**a**) 3D-CNN; (**b**) HybridSN; (**c**) 3D-SE-DenseNet; (**d**) MDSSAN; (**e**) MSDN; (**f**) Proposed.

The proposed CA-FFDN provides the best average results along with the highest OA, AA, and K values on all three datasets. For the IP dataset, compared with the five types of networks, 3D-CNN, HybridSN, 3D-SE-DenseNet, MDSSAN, and MSDN, the OA of CA-FFDN increased by 1.73%, 1.14%, 1.26%, 0.98%, and 0.45%, respectively; for the UP dataset, the OA of CA-FFDN increased by 0.91%, 0.87%, 0.60%, 0.31%, and 0.3%, respectively. For the KSC dataset, the OA of CA-FFDN increased by 2.01%, 1.08%, 1.65%, 0.97%, and 0.75%, respectively. We can note that 3D-CNN has the lowest classification accuracy compared to other networks on the three datasets. The reason is that the network structure of 3D-CNN is too simple to extract fine spatial–spectral features in most cases. The HybridSN combined comprehensive spatial and spectral information in the form of 3D

and 2D convolution. Thus, it performed well in specific land cover, such as alfalfa and corn, on the IP dataset. 3D-SE-DenseNet and MDSSAN gradually refined the extraction of HSI features due to the consecutive dense block structure. Compared with 3D-CNN, the OA improved by 0.47% and 0.75% on the IP dataset, 0.31% and 0.60% on the UP dataset, and 0.36% and 1.04% on the KSC dataset, which indicated the effectiveness of dense networks, but they achieved lower accuracy in the classification of small sample categories, such as grass-pasture-mowed and sheets. The OA of MSDN was reduced by 0.45%, 0.30%, and 0.75% compared to ours on three datasets, respectively. By observing the experimental results, our network outperformed other networks, achieved the highest accuracy, and could quickly converge in fewer training iterations because introducing an efficient channel attention mechanism makes up for the underfitting of the network and preserves as much of the original semantics as possible by compensating for feature loss.

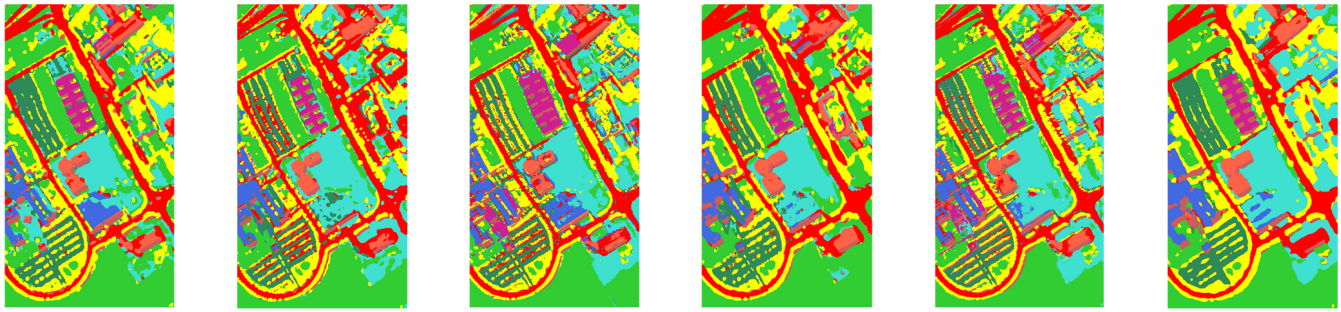


Figure 15. Classification maps for the UP dataset: (**a**) 3D-CNN; (**b**) HybridSN; (**c**) 3D-SE-DenseNet; (**d**) MDSSAN; (**e**) MSDN; (**f**) Proposed.

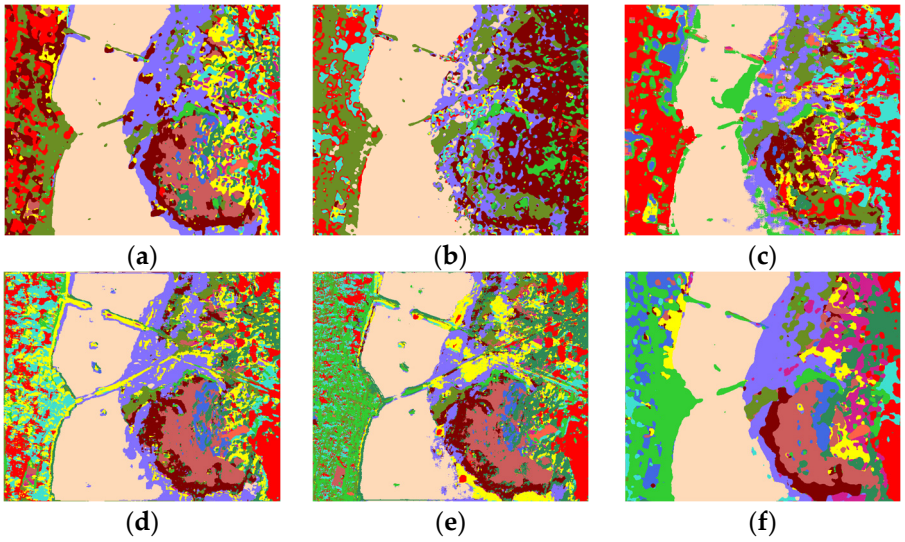


Figure 16. Classification maps for the KSC dataset: (**a**) 3D-CNN; (**b**) HybridSN; (**c**) 3D-SE-DenseNet; (**d**) MDSSAN; (**e**) MSDN; (**f**) Proposed.

Figures 14–16 present the classification maps on three datasets. The completeness of the classification maps remained broadly consistent with the classification results. In particular, 3D-CNN, HybridSN, 3D-SE-DenseNet, and MDSSAN had more noise and misclassification situations on the IP and KSC datasets. The MSDN achieved the fine classification of ground objects, while noise was also significantly reduced compared with the rest of the networks. CA-FFDN was the best regarding ground object classification, generating smoother results with few misclassified samples in classification maps. Due to the large sample size of the UP dataset, the obtained classification maps were relatively good, and there were few noticeable differences between the classification maps.

3.5. Comparative Analysis of Convergence Performance

To further evaluate the convergence performance of the proposed network, more experiments were conducted on each network on three datasets to obtain the accuracy and loss variation between the training and validation sets during the training process. Figures 17–19 portray that CA-FFDN had the slightest loss fluctuation in training and validation sets, and the network converged the fastest. In contrast, 3D-CNN had the slowest convergence speed during the training process because it generally lost important and detailed semantic information, which made it hard to converge fast. In particular, HybridSN was set with the largest batch size of 256 among all networks. Thus, more original data can be trained within a shorter time, which will lead to a fast convergence process. Both 3D-SE-DenseNet and MDSSAN adopt dense connection and attention mechanisms, and the accuracy and loss convergence speed of the training set were greatly improved compared with 3D-CNN.

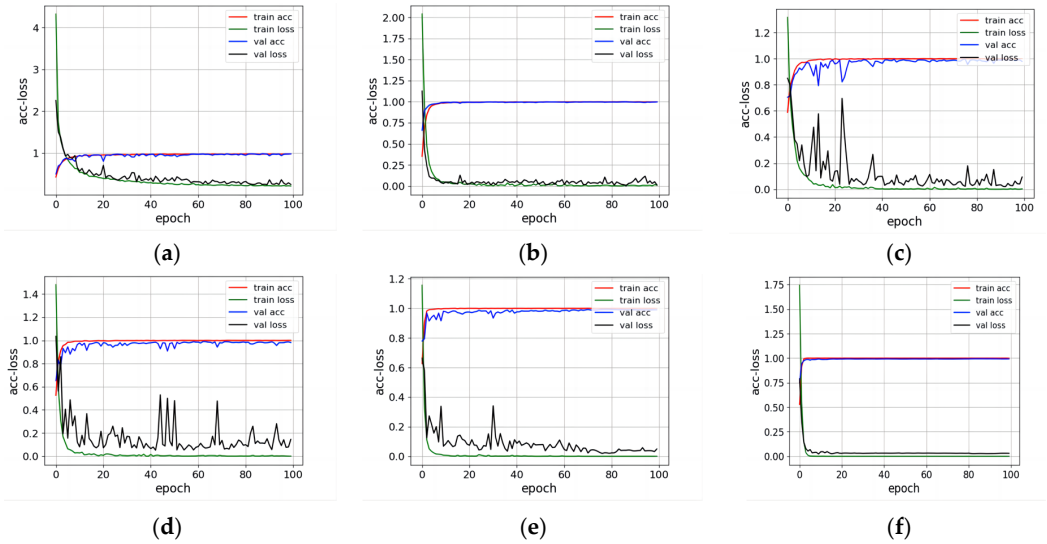


Figure 17. Accuracy loss variation in the training and validation sets on IP dataset: (**a**) 3D-CNN; (**b**) HybridSN; (**c**) 3D-SE-DenseNet; (**d**) MDSSAN; (**e**) MSDN; (**f**) Proposed.

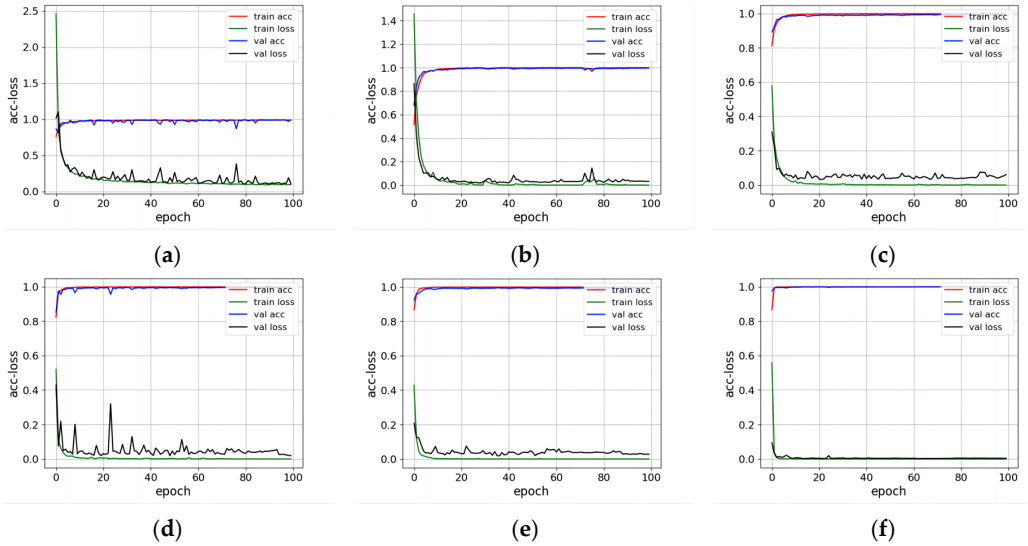


Figure 18. Accuracy loss variation in the training and validation sets on UP dataset: (**a**) 3D-CNN; (**b**) HybridSN; (**c**) 3D-SE-DenseNet; (**d**) MDSSAN; (**e**) MSDN; (**f**) Proposed.

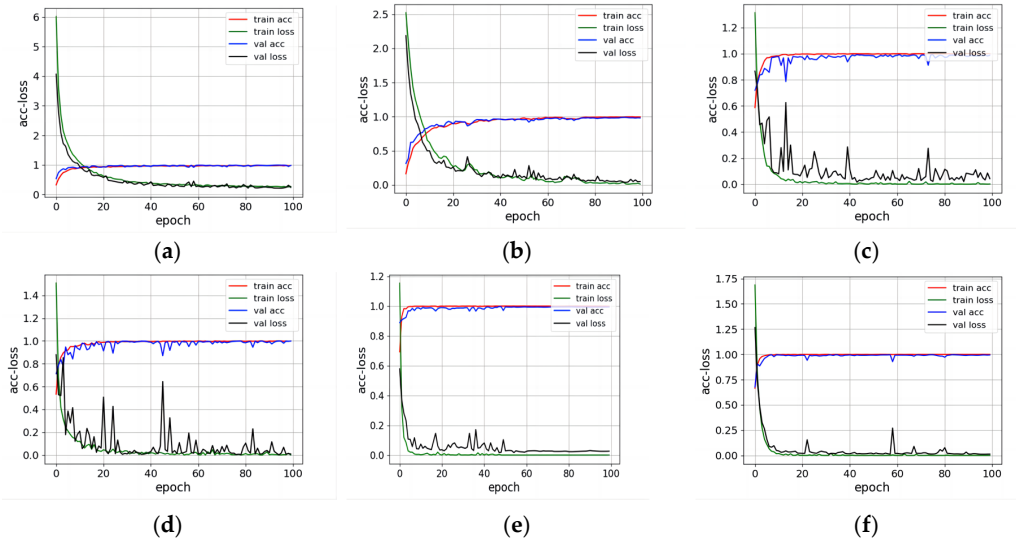


Figure 19. Accuracy loss variation in the training and validation sets on KSC dataset: (**a**) 3D-CNN; (**b**) HybridSN; (**c**) 3D-SE-DenseNet; (**d**) MDSSAN; (**e**) MSDN; (**f**) Proposed.

Many factors can lead to loss fluctuations. The convergence speed of the MSDN was second only to CA-FFDN. Still, there were inevitable fluctuations in the validation set due to its complex network structure with a large number of parameters, so the network's stability was not good enough. The reason why 3D-SE-DenseNet and MDSSN fluctuated severely on the IP and KSC datasets, compared with other networks, was unbalanced data division as well as small batch size. In other words, insufficient validation set samples and small batch size will lead to loss fluctuations. We also recorded each network's parameters and training time in the training process, as shown in Table 8. Compared with MSDN before improvement, CA-FFDN reduced the network parameters by more than half, and the training time was greatly reduced. At the same time, the accuracy and convergence speed of the network was guaranteed, which showed the effectiveness of the proposed network.

Table 8. Training time, test time, and parameters of each network.

Dataset	Time/Parameter	3D-CNN	HybridSN	3D-SE-DenseNet	MDSSAN	MSDN	CA-FFDN
IP	Training time (s)	561.71	36.74	1248.63	454.11	2234.05	807.66
	Test time (s)	4.86	0.32	11.32	2.72	20.94	5.34
	Parameter	27,863,448	796,800	1,171,280	565,167	1,662,134	771,902
UP	Training time (s)	1126.90	173.91	1848.60	999.60	3042.95	1806.45
	Test time (s)	11.33	3.77	31.95	13.36	54.84	25.42
	Parameter	27,862,041	519,417	1,169,985	563,872	1,652,399	770,103
KSC	Training time (s)	299.01	22.41	571.37	241.06	1007.59	459.25
	Test time (s)	2.46	0.18	5.13	1.42	9.05	2.96
	Parameter	27,862,845	796,413	1,170,725	564,612	1,657,779	771,131

3.6. Comparative Analysis of Different Percentages of Training Samples

To further evaluate the generalizability of the proposed CA-FFDN, different percentages of the training samples were tested, 5%, 7%, 9%, 15%, and 20% for the IP and KSC datasets, and 0.5%, 1%, 3%, 5%, and 10% for the UP datasets. It can be observed from Figure 20 that as the percentage of training samples increases, the overall accuracy of each network improves. In detail, MSDN performed worst when training samples were less than 15% on the IP dataset. 3D-SE-DenseNet also had the worst classification results in the case of small samples on UP and KSC datasets. Nevertheless, different networks performed differently on three datasets. But, in general, our CA-FFDN had the most robust

performance under small training samples, which achieved the highest OA among all networks on three datasets.

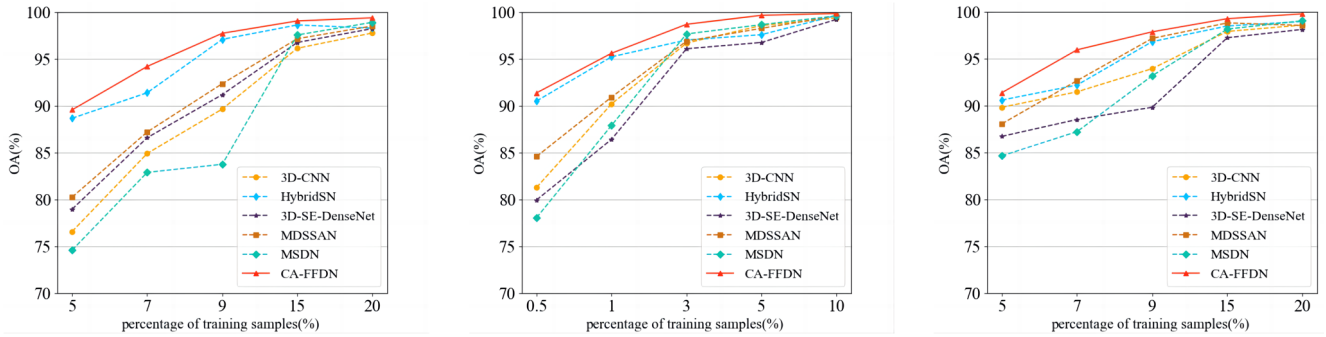


Figure 20. OA of different networks with different training sample percentages: (**a**) IP dataset; (**b**) UP dataset; (**c**) KSC dataset.

4. Conclusions

In this paper, we propose a novel dual-scale dense network, CA-FFDN, to deal with the problems of slow convergence and low classification accuracy caused by insufficient spatial–spectral feature extraction in HSI classification tasks. Our CA-FFDN significantly reduces the interference of data noise by using PCA technology, which can be used to manage large hyperspectral data cubes. Numerous experimental results demonstrate that the CA-FFDN realizes the best classification results by providing a state-of-the-art SC-bottleneck-based dense network and ECA-based feature fusion strategy. Finer spatial–spectral features can be extracted by applying efficient attentional feature fusion between the dual-scale layers, and the highest classification metrics can be obtained by using ECANet rather than employing the compared attention mechanisms in this paper. The results of multiple comparison experiments with five advanced networks show that the overall accuracies of the proposed network under a ratio of 2:1:7 on the IP, 1:1:8 on the UP dataset, and 2:1:7 on the KSC dataset reached 99.51%, 99.91%, and 99.89%, respectively, achieving the highest classification accuracy as well as obtaining the smoothest classification maps with the least noise. Furthermore, our CA-FFDN converged the fastest in the training process, and the loss fluctuated minimally on the training and validation sets. Also worthy of mention is that our CA-FFDN outperformed the other five advanced networks, even with small training samples. In the future, we intend to conduct further research on unsupervised learning algorithms in HSI classification tasks with small samples.

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

Funding: This research was funded by Shanghai Science and Technology Innovation Action Planning, No. 20dz1203800.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are grateful to the editor and reviewers for their constructive comments, which have significantly improved this work.

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

References

1. Li, Z.; Huang, L.; He, J. A Multiscale Deep Middle-level Feature Fusion Network for Hyperspectral Classification. *Remote Sens.* **2019**, *11*, 695. [CrossRef]
2. Yadav, C.S.; Pradhan, M.K.; Gangadharan, S.M.P.; Chaudhary, J.K.; Singh, J.; Khan, A.A.; Haq, M.A.; Alhussen, A.; Wechtaisong, C.; Imran, H.; et al. Multi-Class Pixel Certainty Active Learning Model for Classification of Land Cover Classes Using Hyperspectral Imagery. *Electronics* **2022**, *11*, 2799. [CrossRef]
3. Boshkovski, B.; Doupis, G.; Zapolska, A.; Kalaitzidis, C.; Koubouris, G. Hyperspectral Imagery Detects Water Deficit and Salinity Effects on Photosynthesis and Antioxidant Enzyme Activity of Three Greek Olive Varieties. *Sustainability* **2022**, *14*, 1432. [CrossRef]
4. Pande, C.B.; Moharir, K.N. Application of hyperspectral remote sensing role in precision farming and sustainable agriculture under climate change: A review. In *Climate Change Impacts on Natural Resources, Ecosystems and Agricultural Systems*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 503–520.
5. Liu, C.; Xing, C.; Hu, Q.; Wang, S.; Zhao, S.; Gao, M. Stereoscopic hyperspectral remote sensing of the atmospheric environment: Innovation and prospects. *Earth-Sci. Rev.* **2022**, *226*, 103958. [CrossRef]
6. Mukundan, A.; Tsao, Y.-M.; Cheng, W.-M.; Lin, F.-C.; Wang, H.-C. Automatic Counterfeit Currency Detection Using a Novel Snapshot Hyperspectral Imaging Algorithm. *Sensors* **2023**, *23*, 2026. [CrossRef] [PubMed]
7. Huang, H.-Y.; Hsiao, Y.-P.; Mukundan, A.; Tsao, Y.-M.; Chang, W.-Y.; Wang, H.-C. Classification of Skin Cancer Using Novel Hyperspectral Imaging Engineering via YOLOv5. *J. Clin. Med.* **2023**, *12*, 1134. [CrossRef]
8. Mukundan, A.; Huang, C.-C.; Men, T.-C.; Lin, F.-C.; Wang, H.-C. Air Pollution Detection Using a Novel Snap-Shot Hyperspectral Imaging Technique. *Sensors* **2022**, *22*, 6231. [CrossRef] [PubMed]
9. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
10. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 645–657. [CrossRef]
11. Camps-Valls, G.; Gomez-Chova, L.; Muñoz-Marí, J.; Vila-Francés, J.; Calpe-Maravilla, J. Composite kernels for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 93–97. [CrossRef]
12. Hasan, H.; Shafri, H.Z.; Habshi, M. A comparison between support vector machine (SVM) and convolutional neural network (CNN) models for hyperspectral image classification. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *357*, 012035. [CrossRef]
13. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4085–4098. [CrossRef]
14. 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.* **2011**, *50*, 809–823. [CrossRef]
15. Vaddi, R.; Manoharan, P. Hyperspectral image classification using CNN with spectral and spatial features integration. *Infrared Phys. Technol.* **2020**, *107*, 103296. [CrossRef]
16. Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1264. [CrossRef]
17. 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]
18. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
19. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962.
20. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [CrossRef]
21. Fang, L.; Liu, Z.; Song, W. Deep hashing neural networks for hyperspectral image feature extraction. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1412–1416. [CrossRef]
22. 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]
23. 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, 27–30 June 2016; pp. 770–778.
24. 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.
25. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [CrossRef]
26. Wang, W.; Dou, S.; Jiang, Z.; Sun, L. A fast dense spectral–spatial convolution network framework for hyperspectral images classification. *Remote Sens.* **2018**, *10*, 1068. [CrossRef]
27. Tu, C.; Liu, W.; Jiang, W.; Zhao, L. Hyperspectral Image Classification Based on Residual Dense and Dilated Convolution. *Infrared Phys. Technol.* **2023**, *131*, 104706. [CrossRef]

28. Mou, L.; Ghamisi, P.; Zhu, X.X. Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 391–406. [CrossRef]
29. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens* **2018**, *56*, 5046–5063. [CrossRef]
30. Mou, L.; Lu, X.; Li, X.; Zhu, X.X. Nonlocal graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8246–8257. [CrossRef]
31. Xu, Y.; Du, B.; Zhang, L. Robust self-ensembling network for hyperspectral image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. [CrossRef]
32. Zhang, C.; Li, G.; Du, S. Multi-scale dense networks for hyperspectral remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9201–9222. [CrossRef]
33. 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]
34. Li, G.; Zhang, C.; Lei, R.; Zhang, X.; Ye, Z.; Li, X. Hyperspectral remote sensing image classification using three-dimensional-squeeze-and-excitation-DenseNet (3D-SE-DenseNet). *Remote Sens. Lett.* **2020**, *11*, 195–203. [CrossRef]
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
36. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
37. 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/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
38. 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]
39. Qing, Y.; Liu, W. Hyperspectral image classification based on multi-scale residual network with attention mechanism. *Remote Sens.* **2021**, *13*, 335. [CrossRef]
40. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
41. Prasad, S.; Bruce, L.M. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 625–629. [CrossRef]
42. Qing, Y.; Huang, Q.; Feng, L.; Qi, Y.; Liu, W. Multiscale Feature Fusion Network Incorporating 3D Self-Attention for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 742. [CrossRef]
43. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
44. Lin, Z.; Feng, M.; Santos, C.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv* **2017**, arXiv:1703.03130.
45. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
46. 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.* **2021**, *60*, 5518615. [CrossRef]
47. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–Spatial Feature Tokenization Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522214. [CrossRef]
48. Yang, L.; Yang, Y.; Yang, J.; Zhao, N.; Wu, L.; Wang, L.; Wang, T. FusionNet: A Convolution–Transformer Fusion Network for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 4066. [CrossRef]
49. Dai, Y.; Gieseke, F.; Oehmcke, S.; Wu, Y.; Barnard, K. Attentional feature fusion. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–9 January 2021; pp. 3560–3569.
50. Wang, X.; Fan, Y. Hyperspectral image classification based on modified DenseNet joint spatial spectrum attention mechanism. *Laser Optoelectron. Prog.* **2022**, *3*, 5.
51. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
52. Landgrebe, D.A. Available online: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 1 June 2023).
53. Bai, Y.; Xu, M.; Zhang, L.; Liu, Y. Pruning Multi-Scale Multi-Branch Network for Small-Sample Hyperspectral Image Classification. *Electronics* **2023**, *12*, 674. [CrossRef]
54. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
55. Liashchynskyi, P.; Liashchynskyi, P. Grid search, random search, genetic algorithm: A big comparison for NAS. *arXiv* **2019**, arXiv:1912.06059.
56. Alibrahim, H.; Ludwig, S.A. Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; pp. 1551–1559.

57. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
58. 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]