electronics

MDPI

Article

Hybrid Convolutional Network Combining 3D Depthwise Separable Convolution and Receptive Field Control for Hyperspectral Image Classification

Chengle Lin, Tingyu Wang, Shuyan Dong, Qizhong Zhang, Zhangyi Yang and Farong Gao *

School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China
* Correspondence: frgao@hdu.edu.cn

Abstract: Deep-learning-based methods have been widely used in hyperspectral image classification. In order to solve the problems of the excessive parameters and computational cost of 3D convolution, and loss of detailed information due to the excessive increase in the receptive field in pursuit of multi-scale features, this paper proposes a lightweight hybrid convolutional network called the 3D lightweight receptive control network (LRCNet). The proposed network consists of a 3D depthwise separable convolutional network and a receptive field control network. The 3D depthwise separable convolutional network uses the depthwise separable technique to capture the joint features of spatial and spectral dimensions while reducing the number of computational parameters. The receptive field control network ensures the extraction of hyperspectral image (HSI) details by controlling the convolution kernel. In order to verify the validity of the proposed method, we test the classification accuracy of the LRCNet based on three public datasets, which exceeds 99.50% The results show that compare with state-of-the-art methods, the proposed network has competitive classification performance.

Keywords: image classification; feature extraction; classification accuracy; depthwise separable; receptive field; LRCNet; hyperspectral image

Citation: Lin, C.; Wang, T.; Dong, S.; Zhang, Q.; Yang, Z.; Gao, F. Hybrid Convolutional Network Combining 3D Depthwise Separable Convolution and Receptive Field Control for Hyperspectral Image Classification. *Electronics* **2022**, *11*, 3992. https://doi.org/10.3390/electronics11233992

Academic Editor: Jakub Nalepa

Received: 9 November 2022
Accepted: 29 November 2022
Published: 1 December 2022

1. Introduction

Hyperspectral images (HSIs) contain a large amount of spectral and spatial data, which provide abundant information based on the spectral characteristics of the objects and retain the overall shape of an object and its association with the surrounding objects [1]. Considering the characteristics of HSI data, it is important to analyze and extract the spectral and spatial features. HSI processing technology possesses the capability to satisfy military and civilian needs, such as medical image processing, agriculture and geological exploration, and sea resource investigation [2–8]. Consequently, hyperspectral image classification (HSIC) has become a research hotspot in image processing and remote sensing [9].

In the early works of HSIC, convolutional neural networks (CNNs) were usually used to extract the features [10–14]. Cheng et al. proposed a simple, effective method in order to extract hierarchical deep spatial features for HSI classification by exploring the power of off-the-shelf CNN models [15]. Makantasis et al. exploited a convolutional neural network to encode pixels' spectral and spatial information, and a multi-layer perceptron to conduct the classification task [16]. Many deep neural networks have been developed to handle HSIC tasks. Jiao et al. applied fully convolutional networks (FCNs) to the HSIC task for the first time by combining the weighted extracted features and spectral information [17]. Sun et al. introduced a supervised network for better performance and proposed a fully convolutional segmentation network (FCSN) [18]. Kang et al. believed that the CNN-based methods are unable to effectively extract the discriminant features and proposed a dual-path network (DPN) by combining a residual network and dense convolutional network to perform HSIC [19]. In order to obtain additional neighborhood information, Soucy et al. proposed the clustering ensemble U-Net (CEU-Net) by combining clustering methods with

U-Net [20]. Si et al. used DeepLab v3+ technology and a support vector machine (SVM) classifier for HSI feature extraction and classification recognition [21].

The aforementioned methods perform 2D convolution-based operations. However, some researchers [22] believe that 2D CNNs cannot effectively extract the features because they do not consider the correlation information between channels [22]. On the contrary, 3D convolution effectively combines the spatial and spectral features to improve the accuracy. Based on a 3D CNN for feature extraction [13], Hamida et al. presented an efficient method that enables a joint spectral and spatial information process [23]. He et al. proposed a multi-scale 3D deep convolutional neural network (M3D-DCNN) in order to jointly learn both 2D multi-scales and 1D spectral features from HSI data [24]. Zhong et al. introduced additional residual blocks by using 3D convolutional layers and proposed the spectral–spatial residual network (SSRN) [25]. Roy et al. proposed HybridSN by combining 2D and 3D convolutions to obtain a higher classification accuracy [22]. Zhu et al. proposed the 3D deep capsule network based on the abundant feature representation capability [26]. Sun et al. proposed the cubic capsule network (EMAP-Cubic-Caps) in order to overcome the shortcomings, including the inability to capture fine spatial features, and loss of important information of PCA dimensionality reduction [27].

Despite the fact that 3D convolution effectively obtains the joint features from spatial and spectral dimensions, it has its own limitations [22,28,29]. A network that incorporates 3D convolutions has a large number of parameters, leading to a higher computational cost [22]. In addition, most networks using 3D convolution do not consider the impact of controlling the receptive field size on the classification accuracy, and they emphasize expanding the receptive fields to obtain a better performance [30,31]. Actually, due to the low spatial resolution of HSIs, there is a considerable loss of detail in large receptive fields [9,32]. Although multiple pooling operations are useful for acquiring multi-scale features, they also have an adverse effect on the classification accuracy due to the loss of detailed features, creating confusion among similar category features [31]. However, if the receptive field is too small, it is not possible to consider the multi-scale features. This may lead to underfitting in the network. Therefore, a suitable receptive field is essential for enhancing the network's classification accuracy.

Recently, researchers have tried to address the above problems. Considering the low spatial resolution of HSIs, Pan et al. proposed the dilated semantic segmentation network (DSSNet) [31]. The authors presented a concept for controlling the receptive field of convolution kernels at 13×13 [31]. However, it is difficult to focus on the joint features of the space and spectrum because DSSNet still extracts features using 2D convolutions. Li et al. argued that existing networks do not effectively combine 2D and 3D convolutions, so they alternately used 2D and 3D units to solve the redundancy of the model structure [33]. Although the method proposed by them can reduce the size of the model, it does not specifically reduce the consumption of 3D convolution. Howard et al. proposed depthwise separable convolution in order to reduce the number of parameters and computations in the 2D convolution process. The authors used the proposed convolution in the lightweight network Mobilenetv1 [34]. Fırat et al. introduced 2D depthwise separable convolution in HSIC tasks to decrease the computational cost [35]. Sandler et al. upgraded the depthwise separable convolution and proposed the inverted residual structure [36]. These methods lessen the number of parameters and computational costs that convolutions introduce. Additionally, these methods are aimed at 2D convolution instead of 3D convolution. We also found that in other research fields, researchers have modified and introduced 3D depthwise separable convolution to reduce the computational cost [37–39]. In order to effectively obtain the multi-scale information from HSIs, Gong et al. proposed the multi-scale squeeze-and-excitation pyramid pooling network (MSPN) [28]. The classification accuracy of this network is affected due to the introduction of a pooling layer without controlling the size of the receptive field. Although there are solutions available for addressing the aforementioned issues, it is difficult to resolve the defects.

From the previous work, two problems can be summarized as follows. First, although 3D convolutions effectively capture the spectral and spatial features, the number of parameters and computations introduced during the training process is large [22]. Second, the spatial resolution of HSIs is usually low and some details are presented only based on a few pixels [31,32]. It is noteworthy that the details may disappear after multiple pooling operations, and the lost details cannot be retrieved by up-sampling [40]. If the network is too deep, it may lose some details during the feature extraction process due to the large receptive field of the convolution kernel. As a result, the classification accuracy is affected.

For the purpose of solving the above problems, the 3D lightweight receptive control network (LRCNet) is proposed in this paper. We combine 2D and 3D convolution to effectively integrate the features from the spatial and spectral dimensions. Next, in order to lower the computational cost and reduce the number of parameters, we employ depthwise separable convolution and convert it from 2D to 3D format. In order to reduce the negative impact of a low spatial resolution, we control the size of the receptive field based on dilated convolutions. Below is a summary of this work's contributions:

1. The application of 3D depthwise separable convolution decreases the computational costs of 3D convolution. Additionally, 3D depthwise convolution can effectively capture spatial and spectral features, while pointwise convolution can extract information from adjacent spectral bands, improving the learning ability of the spectral domain.
2. The receptive field control strategy is adopted. To prevent the loss of detailed information when learning multi-scale features, the receptive filed is gradually increased through dilated convolution. Moreover, the receptive field is left unchanged during 3D convolution to enhance the robustness of the model and lower the risk of overfitting.
3. The experimental results show that the proposed method has a better classification accuracy in three public datasets, indicating that the model is competitive.

The rest of this paper is organized as follows: The LRCNet architecture and the functional block are presented in Section 2. The experimental results and analysis are discussed in Section 3, and the conclusion is presented in Section 4.

2. Methods

The proposed LRCNet's architecture is depicted in Figure 1. For the input HSI, we use principal component analysis (PCA) to reduce the dimensions of the data. Next, a 3D depthwise separable convolutional network comprising three 3D-DW modules is used. Afterwards, a reshape operation is applied, and the resulting data are used as the input of the receptive field control network. This network is followed by a fully connected (FC) module, which consists of three FC layers. Finally, the classification results are obtained.

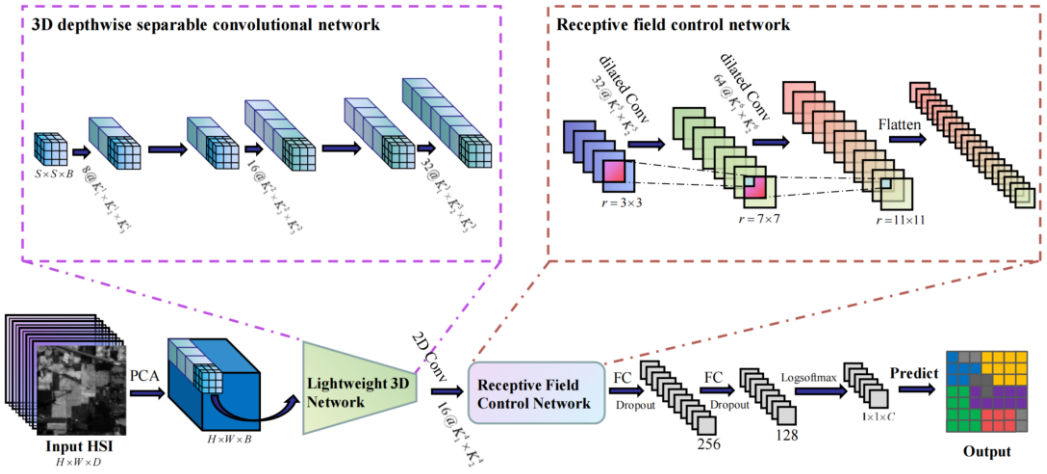


Figure 1. The architecture of the proposed LRCNet.

2.1. Initial Data Input and Processing

As an HSI contains mixed land categories, there is a similarity between different categories. Additionally, a significant percentage of spectral bands exhibit redundancy, which makes it difficult to train models [16]. As shown in Figure 1, in order to reduce the impact of redundant HSI data during the training process, we use PCA before further processing [41]. Assume that the initial hyperspectral image is represented by $I \in \Re^{H \times W \times D}$, where I represents the input HSI data, H and W represent the height and width of the input data, respectively, and D represents the number of bands in the input images. The data cube after dimension reduction based on PCA is $X \in \Re^{H \times W \times B}$, where X represents the data cube and B represents the number of bands after dimension reduction. Next, we divide X into equal sizes and obtain $P \in \Re^{S \times S \times B}$, where P represents the data cube after partition, B is the number of bands, and $S \times S$ represents the height and width of P.

2.2. 3D Depthwise Separable Convolutional Network

Depthwise separable convolution was first proposed by Howard et al. and used in Mobilenetv1 [34]. The standard convolution is split into two parts through the depthwise separable convolution. The first part is the depthwise convolution, which is utilized to extract the features from each input channel separately. The second part is the pointwise convolution, which uses 1×1 convolution to combine the output of the depthwise convolution.

Compared with the standard convolution, the depthwise separable convolution significantly reduces the number of parameters and the computational complexity of the convolution layer. We assume that the size of the input feature map is $H \times W \times C_{in}$ and the parameters of a standard convolution layer are $K_{2D} \times K_{2D} \times C_{in} \times C_{out}$, where H and W represent the height and width of the input data, respectively, C_{in} denotes the number of channels in the input feature map, K_{2D} represents the size of the convolution kernel for performing 2D convolutions, and C_{out} represents the number of output channels. If the feature map size is still $H \times W$, we set $Cost_S$ as the computational complexity of the standard 2D convolution. Next, $Cost_S$ is calculated as follows [34]:

$$Cost_S = K_{2D} \cdot K_{2D} \cdot C_{in} \cdot C_{out} \cdot H \cdot W \tag{1}$$

If 2D depthwise separable convolution is adopted, we assume its computational cost is $Cost_{DW}$. $Cost_{DW}$ consists of two parts. The first part denotes the computational cost of the 2D depthwise convolution, and the second part denotes the computational cost of the 2D pointwise convolution. The costs are represented by $Cost_D$ and $Cost_P$, respectively. In order to compare the computational costs of the 2D depthwise separable convolution and standard 2D convolution, we assume that the size of the convolution kernel is K_{2D}, the numbers of input and output channels are C_{in} and C_{out}, respectively, and the height and width of the input data are H and W, respectively. Next, $Cost_{DW}$ is calculated as follows:

$$\begin{aligned} Cost_{DW} &= Cost_D + Cost_P \\ &= K_{2D} \cdot K_{2D} \cdot C_{in} \cdot H \cdot W + C_{in} \cdot C_{out} \cdot H \cdot W \end{aligned} \tag{2}$$

By comparing the computational costs of the two convolutions, the ratio of the computation is obtained as follows:

$$\begin{aligned} \frac{Cost_{DW}}{Cost_S} &= \frac{K_{2D} \cdot K_{2D} \cdot C_{in} \cdot H \cdot W + C_{in} \cdot C_{out} \cdot H \cdot W}{K_{2D} \cdot K_{2D} \cdot C_{in} \cdot C_{out} \cdot H \cdot W} \\ &= \frac{1}{C_{out}} + \frac{1}{K_{2D}{}^2} \end{aligned} \tag{3}$$

For convenience, we define the computational cost factor ξ_{2D} as the ratio of the computational cost of the current 2D convolution to that of the standard 2D convolution, as shown in Equation (4):

$$\xi_{2D} = \frac{Cost_{DW}}{Cost_S} \tag{4}$$

Generally, the values of C_{out} and K_{2D} are greater than 2; thus, $\xi_{2D} < 1$ can be obtained from Equations (3) and (4), which shows that 2D depthwise separable convolution can effectively decrease the computational costs. If a convolution kernel of size 3×3 is used, the computational cost of 2D depthwise separable convolution can be reduced by about 9 times as compared with the standard 2D convolution. Therefore, a lightweight network can be created using depthwise separable convolution, which can also increase the network's training effectiveness.

In the 2D depthwise convolution part, the features are extracted separately from each input channel. If 2D depthwise convolution is adopted, the connection between different bands of the same pixel is ignored, and the spectral features cannot be learned completely. Moreover, it is easy to ignore the relationship between spatial and spectral features in channel-by-channel convolutions. Although pointwise convolution addresses this defect, there are still many features that cannot be obtained.

Considering the limitations of 2D depthwise separable convolution, we propose the 3D depthwise separable convolution technique, which can fully extract the spatial–spectral features and learn joint features from multiple bands to enhance the classification performance. As each 3D convolution convolves a data block, it is possible to capture the features of adjacent groups of bands. Figure 2 depicts the structure of the proposed 3D depthwise separable convolution (3D-DW) module. The proposed technique also splits the standard 3D convolution into halves, including 3D depthwise convolution and 3D pointwise convolution.

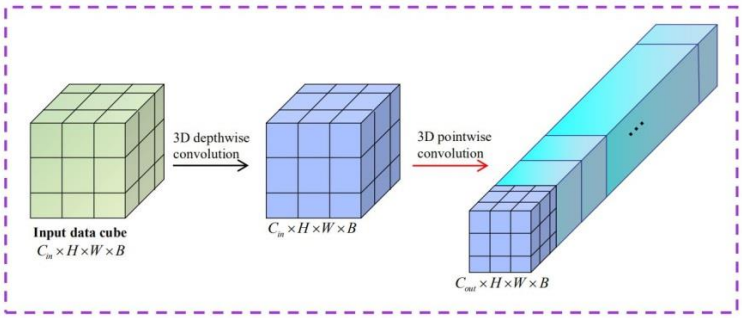


Figure 2. The proposed 3D-DW module.

In addition, the proposed 3D depthwise separable convolution retains the advantages of 2D depthwise separable convolutions. Note that the computational complexity of 3D depthwise separable convolution is lower as compared to the standard 3D convolution.

Assume that the size of the input data cube is $C_{in} \times H \times W \times B$, where C_{in} is the number of input channels, B is the number of bands, and H and W are the height and width of the data cube, respectively. The number of parameters in a standard 3D convolution is $K_{3D} \times K_{3D} \times K_{3D} \times C_{in} \times C_{out}$, where K_{3D} is the size of the 3D convolution kernel and C_{out} is the number of output channels. If the space size of the output data cube remains unchanged, we consider $Cost_{3D-S}$ as the computational cost of the standard 3D convolution. $Cost_{3D-S}$ is computed as follows:

$$Cost_{3D-S} = K_{3D} \cdot K_{3D} \cdot K_{3D} \cdot C_{in} \cdot C_{out} \cdot B \cdot H \cdot W \tag{5}$$

If 3D depthwise separable convolution is adopted, we assume its computational cost is $Cost_{3D-DW}$. $Cost_{3D-DW}$ consists of two parts, i.e., computational cost of 3D depthwise convolution, and computational cost of 3D pointwise convolution, which are denoted as $Cost_{3D-D}$ and $Cost_{3D-P}$, respectively. To compare the computational costs of 3D depthwise separable convolution with those of the standard 3D convolution, we assume that the size of the convolution kernel is K_{3D}, the numbers of input channels and output channels

are C_{in} and C_{out}, respectively, and the height and width of the input data are H and W, respectively. Next, $Cost_{3D-DW}$ is calculated as follows:

$$\begin{aligned} Cost_{3D-DW} &= Cost_{3D-D} + Cost_{3D-P} \\ &= K_{3D} \cdot K_{3D} \cdot K_{3D} \cdot C_{in} \cdot H \cdot W \cdot B + C_{in} \cdot C_{out} \cdot H \cdot W \cdot B \end{aligned} \tag{6}$$

To compare the computational costs of the convolutions, we define the computational cost factor ξ_{3D} as follows:

$$\begin{aligned} \xi_{3D} &= \frac{Cost_{3D-S}}{Cost_{3D-DW}} \\ &= \frac{K_{3D} \cdot K_{3D} \cdot K_{3D} \cdot C_{in} \cdot H \cdot W \cdot B + C_{in} \cdot C_{out} \cdot B}{K_{3D} \cdot K_{3D} \cdot K_{3D} \cdot C_{in} \cdot C_{out} \cdot B \cdot H \cdot W} \\ &= \frac{1}{C_{out}} + \frac{1}{K_{3D}{}^2} \end{aligned} \tag{7}$$

Since $C_{out} \geq 2$ and $K_{3D} \geq 2$, $\xi_{3D} < 1$ is obtained using Equation (11). Therefore, it is evident that 3D depthwise separable convolution greatly reduces the computational cost.

Figure 3 shows the difference between the filters of the 3D depthwise separable convolution and the filters of the standard 3D convolution. Since each input layer channel is convolved separately in depthwise convolution, it is difficult to efficiently utilize the feature information from many channels in the same spatial position. The convolution kernels of the 3D depthwise convolution have three dimensions, so each convolution kernel extracts features from a group of adjacent bands, effectively avoiding the defects of depthwise convolution. Additionally, the number of channels is adjusted, and features are captured again using 3D pointwise convolution. Note that the size of the convolution kernel is only $1 \times 1 \times 1$. Therefore, as compared with the standard convolution, the 3D depthwise separable convolution has significantly fewer parameters and a lower computational cost.

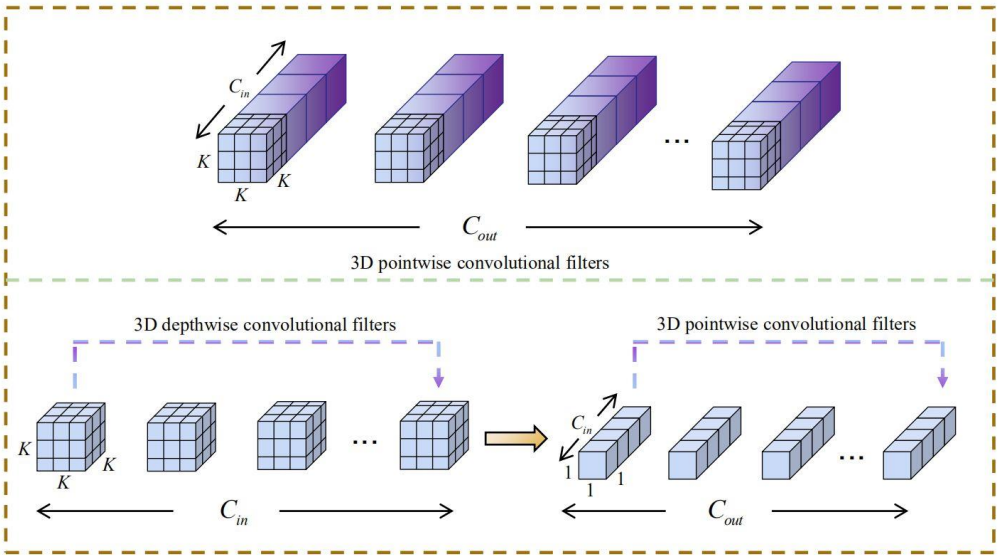


Figure 3. A comparison of a 3D standard convolution kernel and a 3D depthwise separable convolution kernel.

The 3D depthwise separable convolutional network contains three 3D-DW modules. After each depthwise convolution and pointwise convolution, batch normalization (BN) is applied, along with the ReLU activation function. The parameters of the three 3D-DW modules are different. Since all the bands corresponding to each pixel in the HSI image collectively reflect the features of a pixel, it is necessary to aggregate the information from multiple bands as much as possible when extracting the features. Therefore, we set the size of the convolution kernels for the 3D depthwise convolution to $(7,3,3)$, $(5,3,3)$, and $(3,3,3)$.

In addition, the stride and padding parameters of the depthwise convolution and pointwise convolution are set to 1 and 0, respectively. As a result, the number of channels can be increased without changing the height and width of the input images. Due to the low spatial resolution of HSIs, it is easy to lose small features if the data size is compressed too early. This operation ensures that the receptive field of the convolution kernels does not increase during the 3D convolution and that spectral and spatial dimension information can be aggregated.

The essence of 3D depthwise convolution is still 3D convolution. For pixels with spatial position (x, y, z) in the jth feature map of the ith layer, we assume that the activation value $v_{i,j}^{x,y,z}$ is expressed as follows [22]:

$$v_{i,j}^{x,y,z} = f(b_{i,j} + \sum_{\tau=1}^{d_{l-1}} \sum_{\lambda=-\eta}^{\eta} \sum_{\rho=-\gamma}^{\gamma} \sum_{\sigma=-\delta}^{\delta} \omega_{i,j,\tau}^{\sigma,\rho,\lambda} \times v_{i-1,j}^{x+\sigma,y+\rho,z+\lambda}) \tag{8}$$

where f represents the ReLU activation function, $b_{i,j}$ represents the bias parameter for the jth feature map of the ith layer, d_{l-1} denotes the number of feature maps in the $(l-1)$th layer and the depth of kernel $\omega_{i,j}$ for the jth feature map of the ith layer, $2\gamma + 1$ is the width of the convolution kernel, $2\delta + 1$ is the height of the convolution kernel, and $2\eta + 1$ is the depth of the convolution kernel along the spectral dimension.

2.3. Receptive Field Control Network

This network includes a standard 2D convolution layer and two dilated convolution layers. The BN operation and ReLU activation function are also applied after each convolution operation. Since the data format output by the 3D convolution includes four dimensions, we multiply the number of bands and channels to reshape the data into three dimensions. However, this operation results in too many channels of data input. To avoid the impact of data redundancy on the training results, we compress the number of channels using a standard 2D convolution layer. Two dilated convolution layers are added to increase the receptive field for obtaining the multi-scale features. The dilation convolution can also obtain the features between neighbors, which can help to improve the classification accuracy. The stride parameter of the two dilated convolutions is 1, the padding parameter is 0, and the dilation rate is 2. The lower side of each 2D convolution layer in Figure 1 also shows the size of their receptive fields. It can be seen that the size of the final receptive field is 11×11.

Assume that the receptive field after convolution is r_{out}, and the receptive field for introducing the dilated convolution operation is [31]

$$r_{out} = (r_{in} - 1) \cdot stride + (t \times 2 + 1) \tag{9}$$

where r_{in} represents the size of the receptive field of the upper layer, *stride* is the stride parameter of the convolution layer, and t represents the dilation parameter of the dilated convolution.

2.4. Fully Connected Module

The proposed LRCNet consists of three fully connected (FC) layers. The first FC layer converts the feature map output by the last dilation convolution layer into a 1D vector with 256 nodes. Due to the similarity between the classes of HSI data, we further compress 256 nodes into 128 nodes by using an FC layer. As a result, the influence of the feature location on the classification results is reduced in order to improve the network's robustness. To reduce the risk of overfitting, a dropout layer is added after each FC layer. Finally, we use an FC layer with the number of nodes equal to the number of classes in the dataset.

Suppose that the 1D vector output by the FC layer is $\overrightarrow{A} = (a_1, a_2, a_3 \ldots a_{i-1}, a_i)$, where a_i represents the ith element of $\overrightarrow{A}$. Next, a_i is calculated as follows:

$$a_i = b_i + \sum_{\kappa=1}^{q} (G_\kappa \cdot W_{i,\kappa}) \tag{10}$$

where G_κ represents the κth feature map, $W_{i,\kappa}$ denotes the weight matrix of the κth feature map of the ith element, and q denotes the total number of feature maps output by the receptive field control network.

2.5. Classification Result Output

After the third FC layer, we map the output to $(-\infty, 0]$ by using the logsoftmax function. Since the softmax activation function performs exponential operations, overflow or underflow may occur during the calculation; therefore, by using the logsoftmax activation function, problems can be avoided, data stability can be improved, and the operation can be sped up [31]. Assuming that $x_h \in \Re^{1\times C}$ is the output vector of pixel h after passing through the FC layer, where C is the number of object categories in the dataset, the output is expressed as:

$$\widehat{y}_c = \log \frac{e^{x_h(c)}}{\sum_{n=1}^{C} e^{x_h(n)}} \tag{11}$$

where $\widehat{y}_c$ represents the possibility that x_h belongs to category c, and $x_h(c)$ is the cth element in x_h.

The cross-entropy loss is chosen as the loss function. Assuming the loss of classifying pixel h is $loss_{CE}(h, c_t)$, $loss_{CE}(h, c_t)$ can be calculated as:

$$loss_{CE}(h, c_t) = -\widehat{y}_{c_t} = -\log \frac{e^{x_h(c_t)}}{\sum_{n=1}^{C} e^{x_h(n)}} \tag{12}$$

where c_t represents the correct class of pixel h, and $x_h(c_t)$ is the element in x_h that belongs to class c_t.

3. Results

3.1. Dataset Introduction

In this work, we used three public datasets to verify the performance of LRCNet in HSIC tasks [42], including Indian Pines (IndianP), Pavia University (PaviaU), and Salinas Valley (SalinasV).

3.2. Experimental Setup

We used a GTX 1080 Ti with 10GB of memory for training the network. The hyperparameters of LRCNet were set as follows. We set the learning rate to 0.00008, epochs to 100, and batch size to 128. We divided the input image into small windows of 25×25 pixels and reduced the band number to 30. The Adam algorithm was adopted to optimize the learning rate. The cross-entropy function was selected as the loss function. We reserved 30% of the data for testing and 70% of the data for training the network. In this paper, we used the OA, kappa, and AA metrics to evaluate the classification performance. OA represents the number of correctly classified test samples, AA represents the average accuracy of each class, and kappa combines the diagonal and non-diagonal terms of the confusion matrix and is a robust measure of consistency.

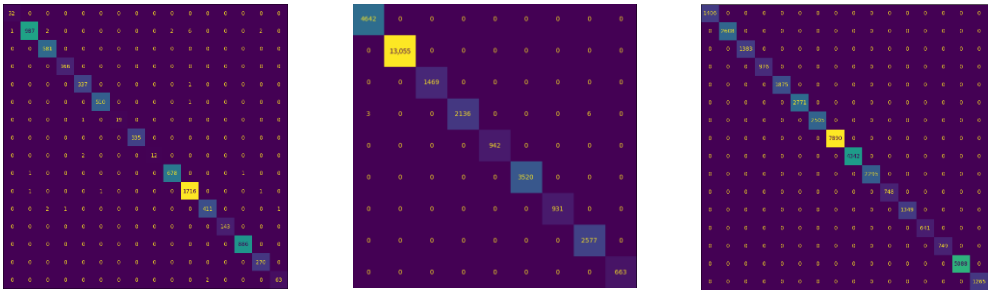
3.3. Classification Results and Analysis

We compared the proposed LRCNet with other methods in detail, including two classical methods, i.e., SVM and 2D-CNN [16,43], five 3D convolutional networks, i.e., 3D-CNN, HybridSN, M3D-CNN, SSRN, and 3D-Caps [22–26], a method for controlling the receptive field named DSSNet [31], and some state-of-the-art methods such as EMAP-Cubic-Caps (EMAP-C-C), MSPN, and SST-M [27,28,44].

Table 1 shows the OA, AA, and kappa values of the different methods based on the three public datasets. The proposed LRCNet clearly performs well on the three datasets, and its classification accuracy has a certain competitiveness. Based on the PaviaU and SalinasV datasets, the classification accuracy of LRCNet is close to 100. However, the classification performance obtained using the IndianP dataset reaches the ideal result, and the AA index score is only 98.40%. Based on the confusion matrix shown in Figure 4, we find that the proposed LRCNet easily misjudges the Soybean-mintill class as the Corn-notill class and Grass-pasture class. The additional observations of the ground truth map of IndianP show that the three classes are very close in the image. We infer that the details of the three categories are wrongly fused together during feature learning.

Table 1. The classification accuracy comparison for the IndianP, PaviaU, and SalinasV datasets (%).

Method	IndianP			PaviaU			SalinasV		
	OA	**Kappa**	**AA**	**OA**	**Kappa**	**AA**	**OA**	**Kappa**	**AA**
SVM [43]	85.30	83.10	79.03	94.36	92.50	92.98	92.95	92.11	94.60
2D-CNN [16]	89.48	87.96	86.14	97.86	97.16	96.55	97.38	97.08	98.84
3D-CNN [23]	91.10	89.98	91.58	96.53	95.51	97.57	93.96	93.32	97.01
M3D-CNN [24]	95.32	94.70	96.41	95.76	94.50	95.08	94.79	94.20	96.25
SSRN [25]	99.19	99.07	98.93	99.90	99.87	99.91	99.98	99.97	99.97
HybridSN [22]	99.56	99.51	98.50	99.85	99.80	99.88	**100**	**100**	**100**
3D-Caps [26]	90.20	90.15	93.00	88.34	84.93	90.14	88.95	87.74	94.35
DSSNet [31]	97.61	97.27	96.31	99.62	99.50	99.22	98.51	98.34	97.56
EMAP-C-C [27]	98.20	96.72	97.95	98.81	98.42	98.49	98.55	98.38	99.08
MSPN [28]	96.09	95.53	91.53	96.56	95.42	94.55	97.00	96.66	97.33
SST-M [44]	99.08	98.95	**99.01**	99.61	99.48	99.23	/	/	/
LRCNet	**99.60**	**99.54**	98.40	**99.97**	**99.96**	**99.95**	**100**	**100**	**100**



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

Figure 4. The confusion matrix obtained using the proposed method by using the IndianP, PaviaU, and SalinasV datasets in the first, second, and third matrices, respectively.

Figures 5–7 show the classification results obtained using LRCNet, HybridSN, and DSSNet on three public datasets. The results show that the classification results obtained with the proposed LRCNet are close to the ground truth.

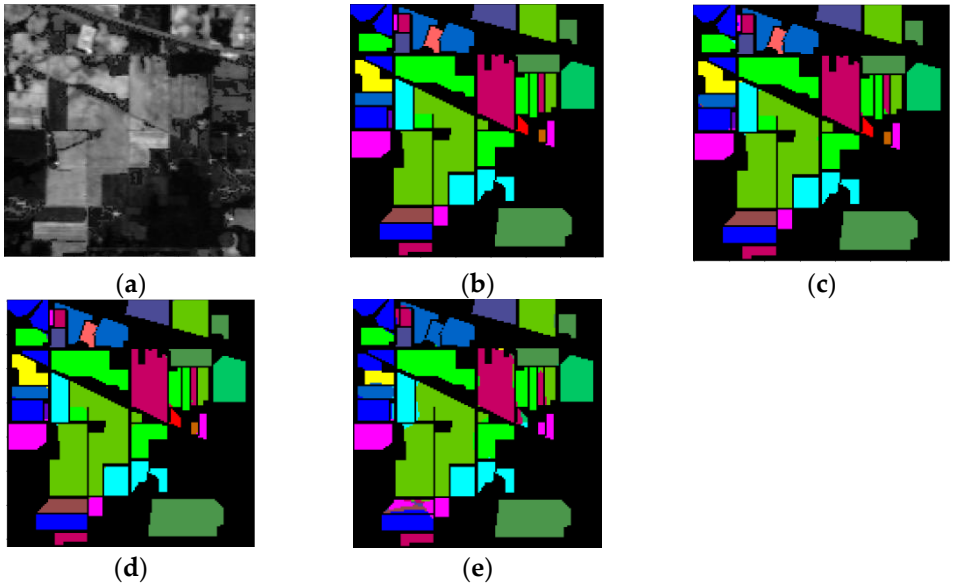


Figure 5. The classification map for the IndianP dataset: (**a**) original image, (**b**) ground truth, and (**c**–**e**) classification maps obtained using LRCNet, HybridSN, and DSSNet, respectively.

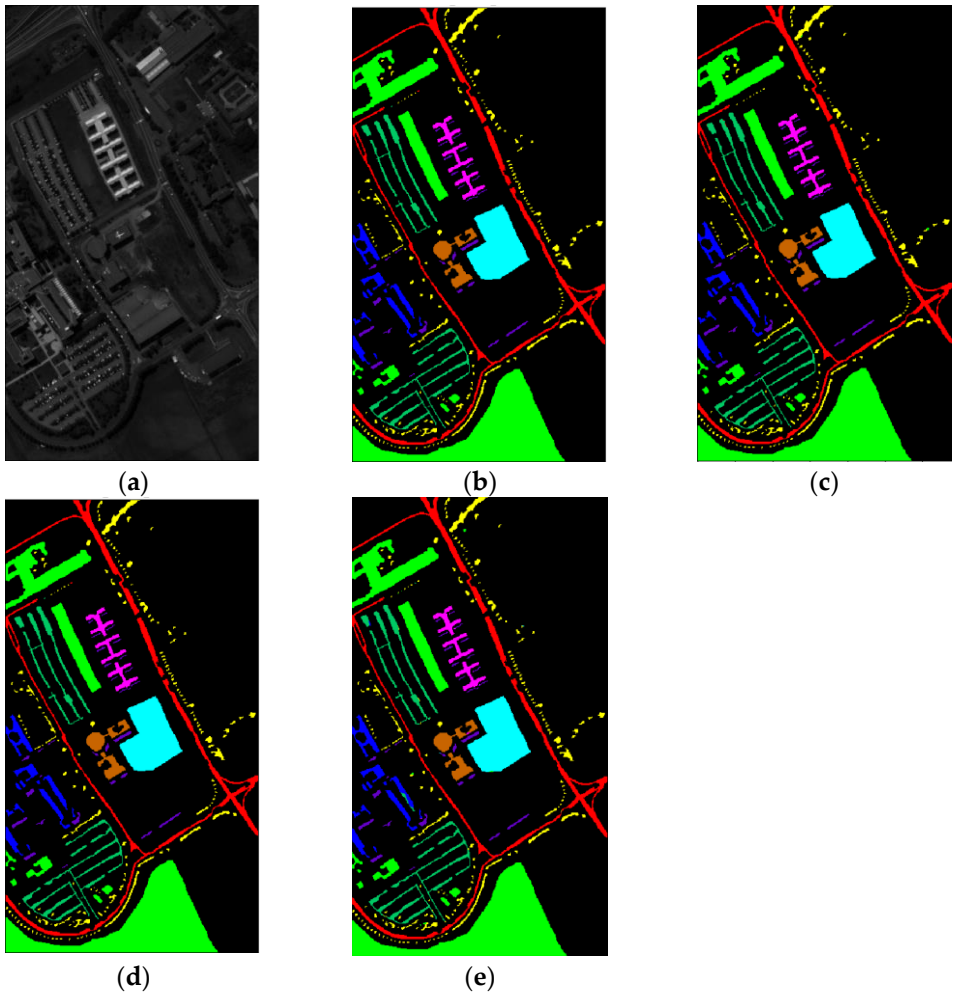


Figure 6. The classification map for the PaviaU dataset: (**a**) original image, (**b**) ground truth, and (**c**–**e**) classification maps obtained using LRCNet, HybridSN, and DSSNet, respectively.

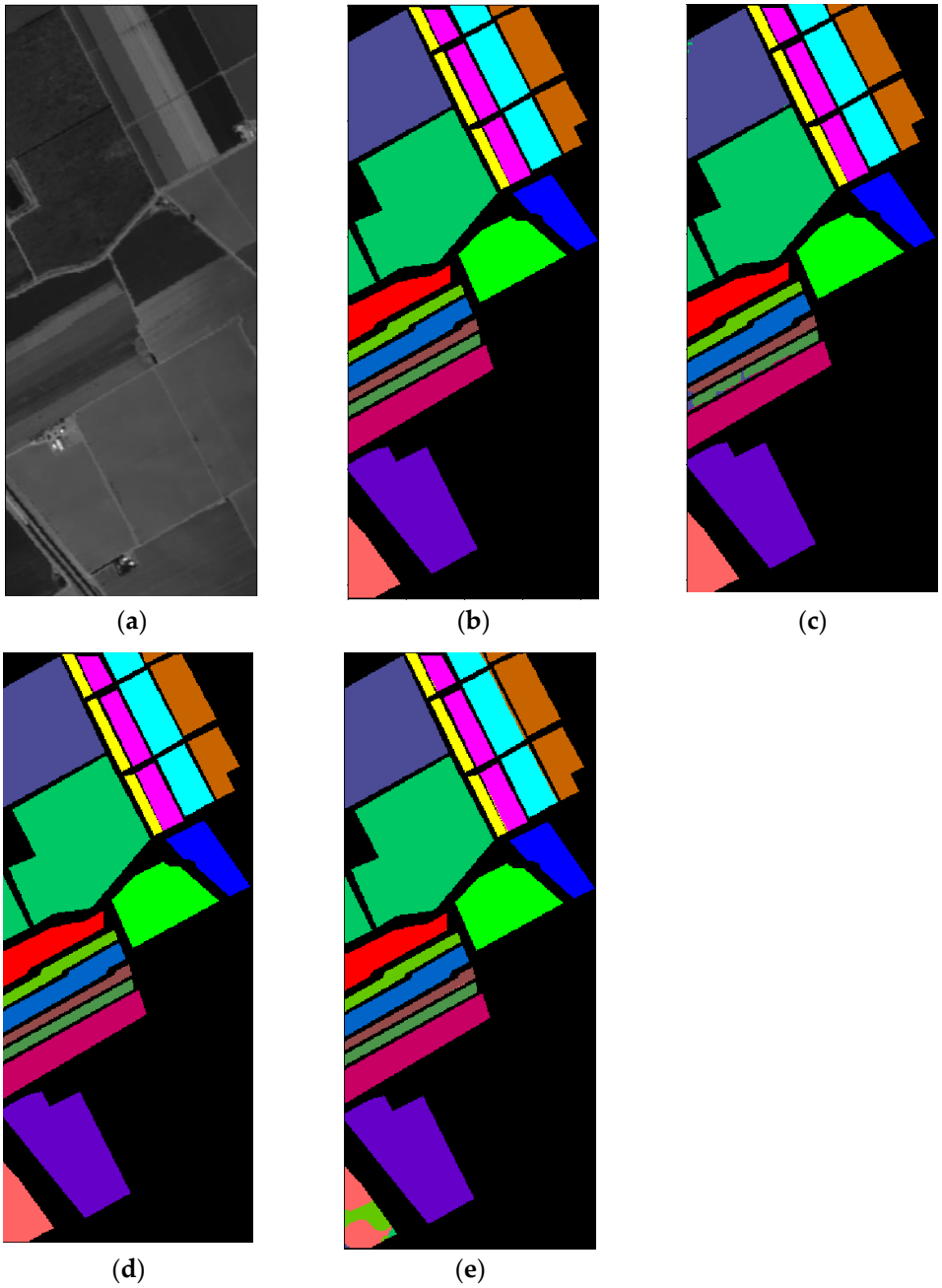


Figure 7. The classification map for the SalinasV dataset: (**a**) original image, (**b**) ground truth, and (**c**–**e**) classification maps obtained using LRCNet, HybridSN, and DSSNet, respectively.

In order to further verify the LRCNet's high classification performance, we performed a one-sample t-test experiment to study if the mean values of OA, kappa, and AA were substantially different from those in Table 1. We repeated 10 tests on three datasets, and the results are shown in Table 2. The results in Tables 1 and 2 do not significantly differ from one another. It can be inferred that the outstanding classification performance of LRCNet is not an accidental result.

Table 2. The classification accuracy of LRCNet repeated experiments on three datasets (%).

Datasets	OA	Kappa	AA
IndianP	99.31 ± 0.34	99.22 ± 0.38	98.40 ± 0.72
PaviaU	99.95 ± 0.04	99.93 ± 0.06	99.92 ± 0.04
SalinasV	99.99 ± 0.01	99.99 ± 0.01	99.98 ± 0.02

Table 3 displays the proposed LRCNet's classification performance under different training set partition ratios. IndianP was used in the experiment. It can be found that the decline in the OA and kappa indicators is small. When the training set only accounts for 10%, the score of the OA indicator also reaches 97.90%. It is evident that the proposed LRCNet learns most of the spatial and spectral features by using a small number of training samples.

Table 3. The classification accuracies obtained using a small number of training samples (%).

Proportion of Training Samples	OA	Kappa	AA
30%	99.60	99.54	98.40
20%	98.68	98.50	95.10
10%	97.90	97.60	88.16

Table 4 shows the number of parameters and the computational cost of training the proposed LRCNet and HybridSN by using the IndianP dataset. It is evident from Table 4 that the proposed method effectively reduces the number of parameters and the computational cost of 3D convolution. The LRCNet has only 3,857,330 parameters, and the floating point operations per second (Flops) is only 95.71MB, which is 152.91MB less than that of HybridSN. Therefore, it can be verified that the 3D depthwise separable technology ensures classification accuracy and reduces the calculation cost.

Table 4. The comparison of the number of parameters and computational cost.

Method	Params	Flops (MB)
LRCNet	3,857,330	95.71
HybridSN	5,122,176	248.62

3.4. Ablation Experiments

In order to confirm the impact of the 3D-DW module on the classification accuracy, we performed ablation experiments. We contrasted the proposed LRCNet and the two modified networks in terms of classification performance. In Net1, we replaced the third 3D-DW module with a 2D-DW module. Similarly, in Net2, we replaced the second and third 3D-DW modules with two 2D-DW modules.

We tested the classification performance of the three networks based on IndianP. Table 5 displays the results. It is evident from the results that the classification performance reduces significantly after the 3D-DW module is replaced with the 2D-DW module, and the AA index decreases the most. When only one 3D-DW module is used, the score of the AA index is only 84.29%. When the number of 3D-DW modules is increased to 2, the score of the AA index increases to 93.75%. For some classes with a small number, it is difficult to learn the corresponding features using the 2D-DW module, and they are often classified as other classes during classification. Although the scores of the OA and kappa indicators do not drop considerably, the scores of the AA indicator are much lower. This shows the validity of the 3D-DW module, which learns the joint features of the spectral and spatial dimensions.

Table 5. The classification accuracy obtained using a small number of training samples (%).

Networks	Architecture of 3D-DW Part	OA	Kappa	AA
LRCNet	three 3D-DW modules	99.60	99.54	98.40
Net1	two 3D-DW modules and one 2D-DW module	97.16	96.75	93.75
Net2	one 3D-DW module and two 2D-DW modules	97.09	96.68	84.29

In order to verify the effectiveness of the receptive field control strategy, we also conducted ablation experiments. We changed the dilation rate of the dilated convolution in the receptive field control network by using three different values, i.e., 1, 3, and 4. The corresponding networks are Net3, Net4, and Net5.

We tested the classification accuracies of the three networks and compared them with LRCNet. The results are presented in Table 6. It is evident from the results that no matter whether the size of the receptive field continues to increase or decrease, the classification accuracy decreases by varying degrees. When the receptive field is 7×7, it is too small to capture the multi-scale information. Therefore, for smaller categories, the network is unable to learn all the features and is prone to incorrect classification results. When the receptive field is increased to 15×15 and 19×19, OA decreases to 98.68% and 98.45%, respectively. It can be inferred that a large receptive field loses some details. More details are lost when the receptive field is larger. Therefore, the strategy for controlling the receptive field of LRCNet is successful.

Table 6. The classification accuracy of different receptive fields (%).

Networks	Dilation Rate	Receptive Field	OA	Kappa	AA
LRCNet	2	11×11	99.60	99.54	98.40
Net3	1	7×7	99.37	99.28	98.06
Net4	3	15×15	98.68	98.49	97.25
Net5	4	19×19	98.45	98.24	92.94

4. Conclusions

In this work, we proposed the LRCNet for performing HSIC tasks, which is an end-to-end framework comprising two functional modules, including a 3D depthwise separable convolutional network, which is used to reduce the computational cost of the convolution and the number of parameters, and the receptive field control network, which is used to control the receptive field for capturing multi-scale features. In the 3D convolutional network, we propose the 3D depthwise separable convolution technique to decrease the number of parameters and the computational costs. In the receptive field control network, we use dilated convolutions to control the receptive field for capturing the multi-scale features and to avoid the loss of detailed features. In the ablation study, we found that the strategy of mixing 3D and 2D convolutions and controlling the receptive field can enhance the classification accuracy. In order to verify the classification performance of the proposed LRCNet, we tested it on three public datasets and obtained competitive results. The proposed method can be applied to accurately identify and classify ground objects in hyperspectral images.

Author Contributions: All authors contributed substantially to this study. Individual contributions were as follows: conceptualization, Q.Z.; methodology, C.L. and F.G.; software, C.L. and T.W.; validation, T.W. and S.D.; formal analysis, C.L.; investigation, Q.Z. and Z.Y.; resources, F.G.; writing—original draft preparation, C.L. and Q.Z.; writing—review and editing, F.G. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Zhejiang Provincial Innovative Incubation Projects for University Students (Emerging Artists Talent Program), grant number 2021R407016, and the Open Foundation of Key Laboratory of Submarine Geosciences, MNR, grant number KLSG2002.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank those who provided help in this study.

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

References

1. Zhang, H.; Zou, J.; Zhang, L. EMS-GCN: An End-to-End Mixhop Superpixel-Based Graph Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5526116. [CrossRef]
2. Feng, J.; Zhao, N.; Shang, R.; Zhang, X.; Jiao, L. Self-Supervised Divide-and-Conquer Generative Adversarial Network for Classification of Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5536517. [CrossRef]
3. Bayramoglu, N.; Kaakinen, M.; Eklund, L.; Heikkila, J. Towards virtual H&E staining of hyperspectral lung histology images using conditional generative adversarial networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 64–71.
4. Han, Y.; Shi, X.; Yang, S.; Zhang, Y.; Hong, Z.; Zhou, R. Hyperspectral Sea Ice Image Classification Based on the Spectral-Spatial-Joint Feature with the PCA Network. *Remote Sens.* **2021**, *13*, 2253. [CrossRef]
5. Hu, K.; Weng, C.; Zhang, Y.; Jin, J.; Xia, Q. An Overview of Underwater Vision Enhancement: From Traditional Methods to Recent Deep Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 241. [CrossRef]
6. Zhou, J.; Yang, T.; Zhang, W. Underwater vision enhancement technologies: A comprehensive review, challenges, and recent trends. *Appl. Intell.* **2022**, 1–28. [CrossRef]
7. Ye, P.; Han, C.; Zhang, Q.; Gao, F.; Yang, Z.; Wu, G. An Application of Hyperspectral Image Clustering Based on Texture-Aware Superpixel Technique in Deep Sea. *Remote Sens.* **2022**, *14*, 5047. [CrossRef]
8. Zhang, Q.; Zheng, E.; Wang, Y.; Gao, F. Recognition of ocean floor manganese nodules by deep kernel fuzzy C-means clustering of hyperspectral images. *J. Image Graph.* **2021**, *26*, 1886–1895.
9. Li, S.; Song, W.; Fang, L.; Chen, Y.; Benediktsson, J.A. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
10. Wang, Q.; Meng, Z.; Li, X. Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2077–2081. [CrossRef]
11. Yuan, Y.; Lin, J.; Wang, Q. Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization. *IEEE Trans. Cybern.* **2015**, *46*, 2966–2977. [CrossRef]
12. Pan, B.; Shi, Z.; Xu, X. Hierarchical guidance filtering-based ensemble classification for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4177–4189. [CrossRef]
13. 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]
14. Pan, B.; Shi, Z.; Xu, X. MugNet: Deep learning for hyperspectral image classification using limited samples. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 108–119. [CrossRef]
15. Cheng, G.; Li, Z.; Han, J.; Yao, X.; Guo, L. Exploring hierarchical convolutional features for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6712–6722. [CrossRef]
16. 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.
17. Jiao, L.; Liang, M.; Chen, H.; Yang, S.; Liu, H.; Cao, X. Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5585–5599. [CrossRef]
18. Sun, H.; Zheng, X.; Lu, X. A supervised segmentation network for hyperspectral image classification. *IEEE Trans. Image Process.* **2021**, *30*, 2810–2825. [CrossRef]
19. Kang, X.; Zhuo, B.; Duan, P. Dual-path network-based hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 447–451. [CrossRef]
20. Soucy, N.; Sekeh, S.Y. CEU-Net: Ensemble Semantic Segmentation of Hyperspectral Images Using Clustering. *arXiv* **2022**, arXiv:2203.04873.
21. Si, Y.; Gong, D.; Guo, Y.; Zhu, X.; Huang, Q.; Evans, J.; He, S.; Sun, Y. An Advanced Spectral–Spatial Classification Framework for Hyperspectral Imagery Based on DeepLab v3+. *Appl. Sci.* **2021**, *11*, 5703. [CrossRef]
22. 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.* **2019**, *17*, 277–281. [CrossRef]
23. Hamida, A.B.; Benoit, A.; Lambert, P.; Amar, C.B. 3-D deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434. [CrossRef]
24. He, M.; Li, B.; Chen, H. Multi-scale 3D deep convolutional neural network for hyperspectral image classification. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3904–3908.
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. Zhu, K.; Chen, Y.; Ghamisi, P.; Jia, X.; Benediktsson, J.A. Deep convolutional capsule network for hyperspectral image spectral and spectral-spatial classification. *Remote Sens.* **2019**, *11*, 223. [CrossRef]
27. Sun, L.; Song, X.; Guo, H.; Zhao, G.; Wang, J. Patch-wise semantic segmentation for hyperspectral images via a cubic capsule network with EMAP features. *Remote Sens.* **2021**, *13*, 3497. [CrossRef]
28. Gong, H.; Li, Q.; Li, C.; Dai, H.; He, Z.; Wang, W.; Li, H.; Han, F.; Tuniyazi, A.; Mu, T. Multiscale information fusion for hyperspectral image classification based on hybrid 2D-3D CNN. *Remote Sens.* **2021**, *13*, 2268. [CrossRef]
29. Ghaderizadeh, S.; Abbasi-Moghadam, D.; Sharifi, A.; Zhao, N.; Tariq, A. Hyperspectral image classification using a hybrid 3D-2D convolutional neural networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7570–7588. [CrossRef]
30. Xu, H.; Yao, W.; Cheng, L.; Li, B. Multiple spectral resolution 3D convolutional neural network for hyperspectral image classification. *Remote Sens.* **2021**, *13*, 1248. [CrossRef]
31. Pan, B.; Xu, X.; Shi, Z.; Zhang, N.; Luo, H.; Lan, X. DSSNet: A simple dilated semantic segmentation network for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1968–1972. [CrossRef]
32. Yokoya, N.; Chan, J.C.-W.; Segl, K. Potential of resolution-enhanced hyperspectral data for mineral mapping using simulated EnMAP and Sentinel-2 images. *Remote Sens.* **2016**, *8*, 172. [CrossRef]
33. Li, Q.; Wang, Q.; Li, X. Exploring the relationship between 2D/3D convolution for hyperspectral image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 8693–8703. [CrossRef]
34. 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.
35. Firat, H.; Asker, M.E.; Hanbay, D. Hybrid 3D Convolution and 2D Depthwise Separable Convolution Neural Network for Hyperspectral Image Classification. *Balk. J. Electr. Comput. Eng.* **2022**, *10*, 35–46. [CrossRef]
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018*; pp. 4510–4520.
37. Jiang, Y.; Han, W.; Ye, L.; Lu, Y.; Liu, B. Two-Stream 3D MobileNetV3 for Pedestrians Intent Prediction Based on Monocular Camera. In Proceedings of the International Conference on Neural Computing for Advanced Applications, Jinan, China, 8–10 July 2022; pp. 247–259.
38. Hou, B.; Liu, Y.; Ling, N.; Liu, L.; Ren, Y. A Fast Lightweight 3D Separable Convolutional Neural Network With Multi-Input Multi-Output for Moving Object Detection. *IEEE Access* **2021**, *9*, 148433–148448. [CrossRef]
39. Alalwan, N.; Abozeid, A.; ElHabshy, A.A.; Alzahrani, A. Efficient 3D deep learning model for medical image semantic segmentation. *Alex. Eng. J.* **2021**, *60*, 1231–1239. [CrossRef]
40. Stergiou, A.; Poppe, R. Adapool: Exponential adaptive pooling for information-retaining downsampling. *arXiv* **2021**, arXiv:2111.00772.
41. 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]
42. Graña, M.; Veganzons, M.A.; Ayerdi, B. Hyperspectral Remote Sensing Scenes. Available online: https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 5 August 2022).
43. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
44. Bai, J.; Wen, Z.; Xiao, Z.; Ye, F.; Zhu, Y.; Alazab, M.; Jiao, L. Hyperspectral Image Classification Based on Multibranch Attention Transformer Networks. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5535317. [CrossRef]