Multiscale Information Fusion for Hyperspectral Image Classification Based on Hybrid 2D-3D CNN

Hang Gong 1, Qixia Li 1, Chunlai Li 2, Haishan Dai 3, Zhiping He 2, Wenjing Wang 1, Haoyang Li 1, Feng Han 1, Abdusalamu Tuniyazi 1 and Tingkui Mu 1,*

1 School of Physics, Xi’an Jiaotong University, Xi’an 710049, China; gh0000@stu.xjtu.edu.cn (H.G.); lqx0324@stu.xjtu.edu.cn (Q.L.); wjwang@stu.xjtu.edu.cn (W.W.); libaoyang@stu.xjtu.edu.cn (H.L.); jhhanfeng@stu.xjtu.edu.cn (F.H.); abdusalam1112@stu.xjtu.edu.cn (A.T.)
2 Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China; lichunlai@mail.sitp.ac.cn (C.L.); hjping@mail.sitp.ac.cn (Z.H.)
3 Shanghai Institute of Satellite Engineering, Shanghai Academy of Spaceflight Technology, Shanghai 201109, China; dlhs1314@126.com
* Correspondence: tkmu@mail.xjtu.edu.cn

Abstract: Hyperspectral images are widely used for classification due to its rich spectral information along with spatial information. To process the high dimensionality and high nonlinearity of hyperspectral images, deep learning methods based on convolutional neural network (CNN) are widely used in hyperspectral classification applications. However, most CNN structures are stacked vertically in addition to using a onefold size of convolutional kernels or pooling layers, which cannot fully mine the multiscale information on the hyperspectral images. When such networks meet the practical challenge of a limited labeled hyperspectral image dataset—i.e., “small sample problem”—the classification accuracy and generalization ability would be limited. In this paper, to tackle the small sample problem, we apply the semantic segmentation function to the pixel-level hyperspectral classification due to their comparability. A lightweight, multiscale squeeze-and-excitation pyramid pooling network (MSPN) is proposed. It consists of a multiscale 3D CNN module, a squeezing and excitation module, and a pyramid pooling module with 2D CNN. Such a hybrid 2D-3D-CNN MSPN framework can learn and fuse deeper hierarchical spatial–spectral features with fewer training samples. The proposed MSPN was tested on three publicly available hyperspectral classification datasets: Indian Pine, Salinas, and Pavia University. Using 5%, 0.5%, and 0.5% training samples of the three datasets, the classification accuracies of the MSPN were 96.09%, 97%, and 96.56%, respectively. In addition, we also selected the latest dataset with higher spatial resolution, named WHU-Hi-LongKou, as the challenge object. Using only 0.1% of the training samples, we could achieve a 97.31% classification accuracy, which is far superior to the state-of-the-art hyperspectral classification methods.

Keywords: convolutional neural network (CNN); hyperspectral image classification; multiscale information

1. Introduction

Remote sensing via hyperspectral imaging is very powerful to capture a set of continuous images of a scene at each resolved narrow band over a wide spectral range. The obtained hyperspectral images are rich in spatial information and spectral information, which can be effectively applied for the classification and recognition of targets over the scene [1–3]. However, hyperspectral remote sensing classification remains difficult due to the problems of highly dimensional, highly nonlinear [4], small samples.

Early research in hyperspectral classification or pattern recognition mainly focused on feature extraction using model-driven algorithms. Commonly used algorithms for
local features are scale-invariant feature transform (SIFT) [5], histogram of oriented gradient (HOG) [6], local binary patterns (LBP) [7], multinomial logistic regression [8], active learning [9], k-nearest neighbor (KNN) [10], support vector machine (SVM) [11], and spectral angle mapping (SAM) [12], etc. Other algorithms put an emphasis on feature discrimination enhancement or dimensionality reduction, such as principal component analysis (PCA) [13], independent component analysis (ICA) [14], linear discriminant analysis (LDA) [15], etc. These methods should follow classifiers to obtain the final results. Although model-driven algorithms make full use of spectral information, the classification maps still contain non-negligible noise due to the underutilization of spatial contextual information. The addition of spatial information can greatly improve the classification accuracy [16]. For example, a 3D Gabor feature-based collaborative representation (3GCR) approach can extract multiscale spatial structured features [17]. However, the above manually designed feature extraction algorithms lack robustness to geometric transformations and photometric variations between unbalanced intraclass samples, which could result in limited resolution of hyperspectral classification. Orthogonal complement subspace projection (OCSP) [18] tries to solve the constraint problem of sample labeling via unsupervised learning. Although hyperspectral classification methods based on hand-designed feature extraction have made great progress in recent years, the complex modeling process limits the further improvement of performance. It also affects the efficiency and model scalability of practical application.

In recent years, deep learning methods have made breakthroughs in the fields of image classification [19], target detection [20], natural language processing [21], etc. Deep learning methods use a hierarchical structure to extract higher-level abstract features from raw data, enabling nonlinear mapping from feature space to label space. They have powerful data mining and feature extraction capabilities and automatically learn features to achieve a more essential portrayal of data, which greatly saves human and material resources and effectively improves the recognition rate or classification accuracy. In particular, deep learning is considered an effective feature extraction method for the hyperspectral classification process [22,23]. For example, a stacked autoencoder (SAE) learns shallow and deep features from hyperspectral images using a single-layer autoencoder and a multilayer stacked autoencoder, respectively [24]. This spatial-dominated information-based deep learning framework yields a higher accuracy compared to model-driven methods based on spectral information. A deep belief network (DBN) extracts the depth and invariant features of hyperspectral data and then uses the learned features in logistic regression to solve the classification problem of hyperspectral data [25]. However, both SAE and DBN first represent the spatial information as vectors before the pretraining phase, which inevitably causes the loss of spatial information.

In contrast, a convolutional neural network (CNN) has the potential to fully extract and exploit spatial–spectral features simultaneously. In the early stage, hundreds of spectral channels of hyperspectral images usually are represented as 1D arrays. Hu et al. [26] used a 1D-CNN for hyperspectral classification, concentrating only on the spectral signatures without considering the spatial correlation. Subsequently, the 2D spatial information was emphasized using a 2D-CNN [27]. Randomized PCA is introduced to reduce the spectral dimensionality while keeping the spatial information unchanged. Thus far, several kinds of 2D-CNN models with innovative network structures have been employed for hyperspectral classification. For example, a deformable 2D-CNN that introduced deformable convolutional sampling locations [28], a dilated 2D-CNN that utilized dilated convolution to avoid resolution reduction due to pooling [29], and a two-stream 2D-CNN network that combined spectral features and spatial features [30]. Furthermore, a deep feature fusion network (DFFN) counteracts negative effects such as overfitting, gradient disappearance, and accuracy degradation in excessive network depth by considering the correlation information between different layers and introducing residual learning [31]. Due to the success of the above feature fusion 2D-CNN models in hyperspectral classification, the joint use of spatial–spectral information has become a mainstream trend.
However, most 2D shape-based filters sacrifice spectral information during feature extraction, which eventually prevents further performance improvement in the classification of complex scenes. Cheng et al. [32] proposed a 3D-CNN framework to extract both spectral and spatial information. In addition, L2 regularization and dropout strategies were used to deal with the overfitting problem caused by limited training samples. Zhong et al. designed an end-to-end spectral–spatial residual network (SSRN) [33] with consecutive residual blocks to learn spectral and spatial representations separately. The network designedly overfits the dataset and then uses residual connectivity to address the gradient disappearance phenomenon. Batch normalization (BN) and dropout were added as regularization strategies to improve the classification performance. However, this deep network structure with a large number of trainable parameters is computationally more expensive than common CNNs. To fully exploit the spatial contextual information of hyperspectral images, He et al. [34] replaced the multiscale block with a typical 3D convolution layer to improve the performance significantly. Sellami et al. [35] kept the initial spectral–spatial features by automatically selecting relevant spectral bands and associated with a 3D-CNN model. Paoletti et al. [36] used pyramidal bottleneck residual blocks to gradually increase the feature map dimension at all convolutional layers for involving more locations. These pyramidal bottleneck residual units designed for hyperspectral images can extract more robust spectral–spatial representations even though they are still computationally expensive. To reduce the complexity of the net framework, Sivalpa K. R. et al. [37] proposed a HybridSN network to combine the complementary spectral–spatial information in the form of 3D and 2D convolutions. The simple hybrid model is more computationally efficient than either the 3D-CNN model or 2D-CNN model and also shows superior performance in terms of the “small sample problem”. On this basis, Feng et al. [38] proposed a R-HybridSN network with skip connections and depth-separable convolution and achieved better classification results than all the contrast models using very few training samples. Ge et al. [39] also proved the effectiveness of 2D-3D CNN with multibranch feature fusion.

Although multiple convolutional layers of feature extraction and subsequent classifiers can be tied together to form end-to-end hierarchical networks, the limited training samples would lead to obvious overfitting problems. While data augmentation is one solution, optimization of the network structure is another important direction. Multiscale feature extraction and information fusion modules can greatly improve network performance. From the perspective of joint use of spatial–spectral information, a multiscale 3D-CNN is effective in extracting features [34]. From the perspective of channels, the channel attention mechanism can focus on more informative channel features. As for representation, a squeeze-and-excitation (SE) block can improve the quality of feature representations by reintegrating spatial–spectral information over the channels [40]. From the perspective of spatial information, an unbalanced distribution of “samples would degrade the capability of discriminating for small sample problem”. Fortunately, pyramid pooling layers can retain global information at different scales [41] and can make better use of features than single-scale pooling layers can.

In this paper, we will combine multiscale 3D convolution, an SE block, and pyramid pooling layers to fully extract and exploit the spatial–spectral information in hyperspectral images. The contributions of this paper are summarized as follows:

1. To overcome the “small sample problem” of hyperspectral pixel-level classification, we designed a multiscale information fusion hybrid 2D-3D CNN named the multiscale squeeze-and-excitation pyramid pooling network (MSPN) model to improve the classification performance. The model can not only deepen the network vertically, but can also expand the multiscale spectral–spatial information horizontally. In terms of being lightweight, the frequently used fully connected layer at the tail is replaced for the first time by a global average pooling layer to reduce model parameters.
2. The proposed MSPN was trained on a small sample with 5%, 0.5%, and 0.5% of the labeled data from the three public datasets of Indian Pine, Salinas, and Pavia University, respectively. The prediction accuracies reached up to 96.09%, 97%, and 96.56%, respectively. The MSPN achieved high classification performance on the small sample.

This paper is organized as follows: Section 2 describes the proposed framework of the MSPN. Section 3 introduces the datasets used in the experiment. Section 4 describes the comparison experiments with existing methods and analyzes and discusses the results. Section 5 concludes the paper and looks at future research directions.

2. Methodology

Figure 1 shows the whole framework of the hyperspectral image classification based on the MSPN. First, dimension reduction is conducted on raw hyperspectral data using PCA and a relatively small number of principal components are kept. The number of retained principal components should have an error of no more than 0.01 in projecting the data from higher to lower dimensions, which means that 99% of the information is retained in the reduced dimensional data. Hyperspectral data can be regarded as a 3D cube with input space \( X \subseteq \mathbb{R}^{W \times H \times B} \), where \( W, H, \) and \( K \) are the width, height, and band, respectively. After selecting the first \( B \) principal components, the hyperspectral data can be denoted as \( \mathbb{R}^{W \times H \times B} \). Next, the hyperspectral image cube is cut into small nearest-neighbor patches \( P_{i,j} \subseteq \mathbb{R}^{r \times r \times B} \), where \( P_{i,j} \) represents the central pixel at \((i,j)\) with a window size of \( r \times r \) covering all spectral dimensions \( B \), composed of pixels from \((i-(r/2),j-(r/2))\) to \((i+(r/2),j-(r/2))\) and from \((i-(r/2),j+(r/2))\) to \((i+(r/2),j+(r/2))\). The head of the MSPN is a multiscale 3D-CNN and accepts the 3D hyperspectral image patch as input. The output space for pixel-level classification is represented as \( Y \subseteq \mathbb{R}^{C \times d} \), where \( L = \{y_0, y_1, \ldots, y_k\} \) is a set of possible categories. In order to determine whether the model has converged in the training stage, the difference between the predicted class labels \( \hat{L} = \{\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_k\} \) and the ground truth class labels \( L = \{y_0, y_1, \ldots, y_k\} \) is estimated using a cross-entropy loss function:

\[
C(\hat{L},L) = \sum_{i=0}^{k} y_i \left( \sum_{j=0}^{k} e^{y_j} - \hat{y}_i \right)
\]  

(1)
The model would be optimized by iteratively updating the parameters through a backpropagation algorithm. The MSPN mainly includes three modules: a multiscale 3D-CNN module, an SE block, and a pyramid pooling module. The multiscale 3D-CNN, consisting of three parallel 3D convolution kernels, is used to extract preliminary spatial–spectral features. The data dimension from the 3D-CNN should be reshaped in order to input into the SE block, which focuses on the attention redistribution of spatial–spectral channel information. The pyramid pooling module further integrates spatial context information. Finally, the global pooling layer is used to replace the fully connected layer and then reduce trainable parameters for avoiding overfitting. Each module is described in detail as follows.

2.1. Multiscale 3D-CNN

Multiscale convolution means that convolution kernels of different sizes are used simultaneously to learn multiscale information [42]. Multiscale information has been applied to the classification due to its rich context information in multiscale structures [43]. The multiscale convolution block can construct a more powerful network model for hyperspectral detection and classification [44]. Our proposed multiscale 3D-CNN consists of \( n = 3 \) parallel convolution kernels with different sizes \( k \times k \times d \), and would extract three feature cubes with the size of \( r \times r \times B' \), such as \( 13 \times 13 \times 16 \), in our experiment. Simultaneously, the position \((x, y, z)\) of each feature cube \( V \) is calculated as follows:

\[
V_{xyz}^{rr} = \mu( \sum_{m=0}^{M-1} \sum_{b=0}^{B-1} \sum_{i=0}^{W-1} K_{ijm}^{rr} \cdot v_{i j m}^{r (r+b)(y+w)(z+b)} + R_{i,j} )
\]

where \( K \) and \( R \) are the parameters of the convolution kernel and the bias term, respectively; \( i, j, m \) are the indexes of input layer, output layer, and feature map, respectively; and \( \mu \) is the activation function, such as the widely used rectified linear unit (ReLU). The input of the 3D-CNN is the small nearest-neighbor patches with the size of \( r \times r \times B \), the subsampling steps of the 3D-CNN is \((s_x, s_y, s_z)\). Then, the output spatial width is \( r' = [1 + (r - k + 1) / s_x] \), and the spectral depth is \( B' = [1 + (B - d + 1) / s_z] \). Let \( s_x = s_y = 1 \) and add the same padding operation to ensure that the output size is equal to the input size. Then, we concatenate the outputs of the three multiscale 3D-CNNs on the channel dimension and feed them into the SE block.

2.2. SE Block

The original generation of the SE block is used to improve network performance by explicitly simulating interdependencies between channels and adaptively recalibrating the characteristic response in terms of channels [45]. Due to the lightweight advantage of the SE block, it also can reduce model parameters and increase detection speed. A learning framework with an SE block can well characterize channel-wise spectral–spatial features [46]. Usually, an SE block includes two steps of squeeze and excitation. The squeeze operation firstly compresses the features along the spatial dimension. It turns each 2D feature channel into a real number, which has a global receptive field to some extent. The statistic \( Z \) of channel dimensions is generated by reducing the spatial dimension \( H \times W \) of the reshaped feature cube \( V \) using global average pooling. The element of \( Z \) is calculated as follows:

\[
Z = f_{S_{SE}}(V) = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} V(\alpha, \beta)
\]

Once the global spatial information is embedded into the feature vector \( Z \in \mathbb{R}^{1 \times B} \), the excitation operation is used to convert the feature vector \( Z \) to another feature vector \( \beta \) as follows:
where \( \mu \) and \( \sigma \) are the ReLU activation function and sigmoid activation function, respectively; \( \omega_1 \) and \( \omega_2 \) represent the weights of two consecutive fully connected layers. The weights are automatically generated by learning the correlation between feature channels explicitly. The output feature channels of \( S \) match the input feature channels of \( Z \). The above two operations can let the output vector \( S \) obtain the global information distribution. They can select the feature channels with more information by turning weights. Finally, the original feature cube \( V \) is recalibrated on the channel dimension by weighting the feature vector \( S \) via multiplication as follows:

\[ V' = f_{pool}(V, S) = S \cdot V \]

Accordingly, the SE block can extract the important information of feature channels automatically and then enhance the selected features and suppress the unusable features.

### 2.3. Pyramid Pooling Module

The pyramid pooling module can further integrate spatial information. It is usually used for scene parsing in image semantic segmentation [47] or pattern recognition [48], since it can aggregate multiple receptive fields in different scales. Furthermore, different scales of receptive field can gather richer spatial context information. In terms of this point, it is reasonable for us to employ it to address the limitations of single-size receptive fields in “small sample problems” of the hyperspectral image classification task and to improve classification accuracy. In this paper, the framework and parameters of the pyramid pooling module are specified as follows. A 2D convolutional layer with a kernel size of \( 3 \times 3 \) and 128 channels is used to extract a feature map before pyramid pooling layers. Then, four pool layers with different receptive fields are used to downsample the feature map to different scales. The sizes of the four pooling layers are \( 13 \times 13 \), \( 7 \times 7 \), \( 5 \times 5 \), and \( 3 \times 3 \). Subsequently, the \( 1 \times 1 \) convolution layer is used to change the number of channels. Then, the four downsampled feature maps are restored to the original size by the deconvolution layer. The original feature map and the restored four feature maps are concatenated in the channel dimension. The last 2D convolution layer with a kernel size of \( 3 \times 3 \) and 256 channels is used to extract the final features, which provide rich global context information for pixel-level classification, even if there is the “small sample problem”.

### 3. Datasets and Details

To verify the proposed model, we used four publicly available hyperspectral datasets, namely the Indian Pine, Salinas, Pavia University, and WHU-Hi-LongKou datasets [49]. Some open codes were used for comparative experiments on the same training and test data. We adjusted the hyperparameters of the MSFPN model using only the three classic datasets and then used the optimal model to challenge the latest high-resolution WHU-Hi-LongKou dataset. The open codes are available online at https://github.com/gokriznastic/HybridSN (March 4, 2021) and https://github.com/eecn/Hyperspectral-Classification (March 4, 2021). The WHU-Hi-LongKou dataset is available online at http://rsidea.whu.edu.cn/resource_WHUHi_sharing.htm (May 25, 2021). The classes and the number of samples for the four datasets are listed in Table 1.
| Order | Class        | Indian Pine Samples | Salinas Class                                      | Samples | Pavia University Class | Samples | WHU-Hi-LongKou Class    | Samples |
|-------|--------------|---------------------|---------------------------------------------------|---------|------------------------|---------|------------------------|---------|
|       | Alfalfa      | 46                  | Brocoli_green_weeds_1                             | 2009    | Asphalt                | 6631    | Corn                   | 34511   |
|       | Corn-notill  | 1428                | Brocoli_green_weeds_2                             | 3726    | Meadows                | 18649   | Cotton                 | 8374    |
|       | Corn-mintill | 830                 | Fallow                                           | 1976    | Gravel                 | 2099    | Sesame                 | 3031    |
|       | Corn         | 237                 | Fallow_rough_plow                                | 1394    | Trees                  | 3064    | Broad-leaf soybean     | 63212   |
|       | Grass-pasture| 483                 | Fallow_smooth                                    | 2678    | Paintedmetalsheets     | 1345    | Narrow-leaf soybean    | 4151    |
|       | Grass-trees  | 730                 | Stubble                                          | 3959    | Bare Soil              | 5029    | Rice                   | 11854   |
|       | Grass-pasture- | 28                  | Celery                                          | 3579    | Bitumen                | 1330    | Water                  | 67056   |
|       | mowed Hay-wind-rowed | 478         | Grapes_untrained                                | 11271   | Self-Blocking Bricks   | 3682    | Roads and houses       | 7124    |
|       | Oats         | 20                  | Soil_vinyard_develop_Corn_se-nesced_green_weeds | 6203    | Shadows                | 947     | Mixed weed             | 5229    |
|       | Soybean-notill| 972                | Soybean_notill                                  | 3278    |                        |         |                        |         |
|       | Soybean-mintill| 2455            | Soybean_notill                                  | 1068    |                        |         |                        |         |
|       | Soybean-clean | 593                | Soybean_clean                                  | 1927    |                        |         |                        |         |
|       | Wheat        | 205                 | Wheat                                            | 916     |                        |         |                        |         |
|       | Woods        | 1265                | Woods                                            | 1070    |                        |         |                        |         |
|       | Buildings-Grass-Trees-Drives | 386 | Brick                                            | 7268    |                        |         |                        |         |
|       | Stone-Steel-Towers | 93            | Stone-Steel-Towers                              | 1807    |                        |         |                        |         |

1. Each Indian Pine hyperspectral image is composed of 145 × 145 pixels with a spatial resolution of 20 m. The bands covering the water-absorbing area are removed, and the remaining 200 bands are used for classification. Indian Pine landscape mainly includes different types of crops, forests, and other perennial plants. The ground truth values are specified into 16 classes. The number of available sample points for all classes is 10,249. Each class includes a minimum of 20 and a maximum of 2455 sample points. As such, the distribution of sample points is very uneven. In addition, the crops are divided into two classes due to the different levels of tillage. For
example, the first class is no-till corn and the second class is low-till corn. Both of them are cornfields.

2. Each Salinas hyperspectral image consists of 512 × 217 pixels with a spatial resolution of 3.7 m. Similar to the Indian Pine images, the water absorption band is discarded and 204 bands remain. The Salinas scenes mainly include vegetation, bare soil, and vineyards. A total of 54,129 sample points are divided into 16 groups. Per class, the minimum number of sample points is 916 and the maximum number is 11,271, which is rather uneven. The Salinas dataset differs from the Indian Pine dataset in that there are more available labeled sample points and the spatial resolution is higher. There is also a similarity between several classes, e.g., classes 11, 12, 13, and 14 are longleaf lettuces, but they are divided into four classes depending on growth time. The advantage of this dataset is the high spatial resolution, which can help to improve the classification effect.

3. Each Pavia University hyperspectral image consists of 610 × 340 pixels with a spatial resolution of 1.3 m. The dataset includes 103 spectral bands. The labeled samples are divided into nine classes. The scenes mainly include urban features, such as metal sheets, roofs, asphalt pavements, etc. Per class, the minimum number of sample points is 947 and the maximum number is 18,649. The total number of labeled sample points is 42,776.

4. The WHU-Hi-LongKou dataset was acquired in 2018 in Longkou Town, China. Each hyperspectral image consists of 550 × 400 pixels with 270 bands from 400 to 1000 nm, and the spatial resolution of the hyperspectral imagery is about 0.463 m. It contains nine classes, namely Corn, Cotton, Sesame, Broad-leaf soybean, Narrow-leaf soybean, Rice, Water, Roads and houses, and Mixed weed. The minimum number of sample points is 3031 and the maximum number is 67,056. The total number of labeled sample points is 204,542.

For the supervised hyperspectral image classification, to our knowledge, the number of training sample points has a significant influence on the classification accuracy. To verify the effectiveness of the MSPN model with the “small sample problem”, we randomly selected 5% of sample points from the Indian Pine dataset, 0.5% of sample points from the Salinas dataset, and 0.5% of sample points from the Pavia University dataset as training samples and ensures that all classes were included. Since the Indian Pine dataset has an extremely uneven distribution of sample points, more training sample points (i.e., 5%) should be used to ensure that all classes are considered. For the other two datasets, since the sample points are rich and relatively uniform, use of fewer training sample points (i.e., 0.5%) is acceptable. For example, both the HybridSN [36] and R-HybridSN [37] models used 1% of sample points from these two datasets to train the networks and achieve a better classification accuracy. However, for the high-resolution WHU-Hi-LongKou dataset, since it has more training sample points, we took only 0.1% of them as training samples to highlight the performance of our model in regard to the “small sample problem”.

4. Results of Experiment

In order to evaluate the performance of the proposed MSPN model and compare it with other supervised deep learning methods, the following five experiments were implemented.

1. The first experiment evaluated the classification performances of the proposed MSPN and other models in terms of overall accuracy (OA), average accuracy (AA), and kappa coefficient, using a different number of training samples of the three classic datasets.

2. The second experiment compared the classification performances of the original MSPN model and its variants, such as by removing the multiscale 3D-CNN, SE block, or pyramid pooling module.
3. The third experiment verified the influence of the selection of principal components on the model performance.
4. The fourth experiment determined the proper number of convolutional kernels and the number of parallel convolutional layers.
5. The fifth experiment aimed to challenge the latest high-resolution remote sensing images in the WHU-Hi-LongKou dataset. We compared our proposed MSPN model with other methods by taking 0.1% of the sample points as a training sample. At the same time, a confusion matrix was used to visualize the classification results of our proposed model.

The performance was evaluated by the indicators OA, AA, and kappa coefficient, where OA is the ratio of correctly classified pixels to total pixels, AA is the average value of classification accuracy, and the kappa coefficient is a metric of statistical measurement that can measure the consistency between a predicted map and the ground truth. The experiments were implemented with Tensorflow. We used mini-batches of size 128 for training the network. The optimizer was Adam, and the learning rate was set to 0.001. All experiments were repeated 10 times and the average value was taken as the final classification accuracy.

For the first experiment, Tables 2–4 list the classification performance of the different methods on all classes, and Figures 2–4 show the ground truths and the predicted classification maps of the different models. Evidently, the classification maps and accuracy of the MSPN are better than those of the other models. It should be noted that the MSPN uses fixed hyperparameters for each dataset. That is, the MSPN has a good generalization ability for different hyperspectral images. The OA, AA, and kappa coefficient obtained by the MSPN were the highest relative to other models: the OA of the MSPN was 96.09% on the Indian Pine dataset, 97% on the Salinas dataset, and 96.56% on the Pavia University dataset. It is interesting to note that the hybrid 3D-CNN and 2D-CNN are superior to either the 2D-CNN or 3D-CNN alone. The addition of multiscale modules further improves the performance.

![Figure 2](image-url)  
**Figure 2.** The predicted classification maps for 5% samples from the Indian Pine dataset with the method of (a) 2D-CNN, (b) M3D-CNN, (c) HybridSN, (d) R-HybridSN, and (e) MSPN and (f) ground truth.

![Figure 3](image-url)  
**Figure 3.** The predicted classification map for 0.5% samples from the Salinas dataset with the method of (a) 2D-CNN, (b) M3D-CNN, (c) HybridSN, (d) R-HybridSN, and (e) MSPN and (f) ground truth.
Figure 4. The predicted classification map for 0.5% samples from the Pavia University dataset with the method of (a) 2D-CNN, (b) M3D-CNN, (c) HybridSN, (d) R-HybridSN, and (e) MSPN and (f) ground truth.

Table 2. Accuracy comparison of different methods for 5% samples of Indian Pine.

| Order | Class           | 2D-CNN | M3D-CNN | HybridSN | R-HybridSN | MSPN |
|-------|-----------------|--------|---------|----------|------------|------|
| 1     | Alfalfa         | 0.85   | 1.00    | 0.96     | 1.00       | 0.91 |
| 2     | Corn-notill     | 0.85   | 0.90    | 0.92     | 0.90       | 0.95 |
| 3     | Corn-mintill    | 0.91   | 0.94    | 0.97     | 0.98       | 0.95 |
| 4     | Corn            | 0.92   | 0.87    | 0.98     | 0.92       | 1.00 |
| 5     | Grass-pasture   | 0.91   | 0.99    | 0.94     | 0.98       | 0.97 |
| 6     | Grass-trees     | 0.48   | 0.97    | 1.00     | 1.00       | 1.00 |
| 7     | Grass-pasture-mowed | 1.00 | 1.00    | 1.00     | 1.00       | 0.90 |
| 8     | Hay-windrowed   | 0.90   | 1.00    | 0.95     | 0.91       | 1.00 |
| 9     | Oats            | 0.00   | 0.93    | 0.75     | 1.00       | 0.93 |
| 10    | Soybean-notill  | 0.98   | 0.92    | 0.88     | 0.87       | 0.90 |
| 11    | Soybean-mintill | 0.92   | 0.89    | 0.92     | 0.96       | 0.97 |
| 12    | Soybean-clean   | 0.89   | 0.84    | 0.97     | 0.96       | 0.95 |
| 13    | Wheat           | 0.98   | 1.00    | 1.00     | 0.95       | 0.99 |
| 14    | Woods           | 1.00   | 0.97    | 0.99     | 0.93       | 0.98 |
| 15    | Buildings-Grass-Trees-Drives | 0.87 | 0.98    | 0.98     | 0.94       | 0.97 |
| 16    | Stone-Steel-Towers | 0.93 | 0.92    | 0.97     | 0.80       | 0.99 |

OA/% 85.56 92.65 94.19 93.66 96.09
Kappa/% 83.49 91.57 93.34 92.77 95.53
AA/% 71.43 86.83 86.79 87.57 91.53

Table 3. Accuracy comparison of different methods for 0.5% samples of Salinas.

| Order | Class                      | 2D-CNN | M3D-CNN | HybridSN | R-HybridSN | MSPN |
|-------|---------------------------|--------|---------|----------|------------|------|
| 1     | Brocoli_green_weeds_1     | 0.94   | 1.00    | 0.98     | 1.00       | 1.00 |
| 2     | Brocoli_green_weeds_2     | 0.92   | 1.00    | 0.97     | 1.00       | 1.00 |
| 3     | Fallow                    | 1.00   | 0.95    | 1.00     | 1.00       | 1.00 |
| 4     | Fallow_rough_plow         | 0.48   | 0.90    | 0.96     | 0.98       | 0.98 |
| 5     | Fallow_smooth             | 1.00   | 0.98    | 1.00     | 0.91       | 0.97 |
| 6     | Stubble                   | 0.99   | 0.99    | 1.00     | 1.00       | 1.00 |
| 7     | Celery                    | 1.00   | 1.00    | 1.00     | 0.99       | 1.00 |
| 8     | Grapes_untrained          | 0.64   | 0.92    | 0.99     | 0.95       | 0.94 |
| 9     | Soil_vinyard_develop      | 0.91   | 0.99    | 0.95     | 1.00       | 0.99 |
Table 4. Accuracy comparison of different methods for 0.5% samples of Pavia University.

| Order | Class                  | 2D-CNN | M3D-CNN | HybridSN | R-HybridSN | MSPN |
|-------|------------------------|--------|---------|----------|------------|------|
| 1     | Asphalt                | 0.94   | 0.76    | 0.78     | 0.94       | 0.95 |
| 2     | Meadows                | 0.90   | 0.85    | 0.94     | 0.99       | 0.98 |
| 3     | Gravel                 | 0.71   | 0.68    | 0.80     | 0.74       | 0.87 |
| 4     | Trees                  | 1.00   | 0.93    | 0.93     | 0.94       | 0.96 |
| 5     | Painted metal sheets   | 0.26   | 1.00    | 0.93     | 0.99       | 1.00 |
| 6     | Bare Soil              | 0.97   | 0.99    | 0.93     | 0.96       | 0.99 |
| 7     | Bitumen                | 0.96   | 0.75    | 0.78     | 0.86       | 0.99 |
| 8     | Self-Blocking Bricks   | 0.46   | 0.67    | 0.81     | 0.71       | 0.93 |
| 9     | Shadows                | 0.35   | 0.54    | 0.97     | 0.84       | 0.95 |
| OA/%  | 74.77                  | 81.95  | 89.10   | 92.54    | 96.56      |      |
| Kappa/%| 66.53                  | 75.21  | 85.35   | 90.12    | 95.42      |      |
| AA/%  | 61.10                  | 68.25  | 76.92   | 89.50    | 94.55      |      |

Figure 5 presents the convergence accuracy and loss on training phase over 50 epochs. Table 5 presents the computational efficiency of the MSPN in terms of training and testing time. As shown, the MSPN model is more efficient than the R-HybridSN model. Convergence was achieved after just 30 epochs, mainly because the full connection layer is replaced with the global pooling layer. In addition, relative to other models, the 3D-CNN layers are reduced, while the 2D-CNN layers are increased, and so the network becomes lightweight. There is no regularization strategy such as batch normalization and dropout.

Figure 5. Training loss and accuracy convergence for (a) Indian Pine, (b) Salinas, and (c) Pavia University.
In the second experiment, the contribution of each module was explored in detail. Removal of the multiscale module means letting the multiscale module become a single-scale one, and only the parameters of the middle 3D-CNN are copied. Removal of the SE block or pyramid pooling module means removing them from the network completely. Figure 6 shows the performance of the MSPN and its variants. The results show that the pyramid pooling module contributes the most to the model. When it is removed, the OA decreases the most, mainly because the module contains the most convolution layers and multiscale information. The SE block has the least impact because it only recalibrates the features from the spectral dimension to a certain extent. Compared with other modules, it only plays an auxiliary role in feature learning.

Table 5. Speed comparison of different methods in the Pavia University dataset with 0.5% training samples (processor: 1.8 GHz Intel Core i5, no GPU acceleration).

| Method          | Train time/s | Test time/s |
|-----------------|--------------|-------------|
| 2D-CNN          | 418.4        | 1512.4      |
| R-HybridSN      | 332.3        | 1336.1      |
| MSPN            | 310.7        | 1035.5      |

Figure 6. OA of the MSPN while removing each module separately.

In the third experiment, the effect of the number of principal components on the classification performance was investigated. The fewer the principal components were, the fewer the spectral features and the shorter the computation time were, and vice versa. Figure 7 shows the classification OA as a function of the number of principal components. As expected, the OA increases with the number of principal components. However, the optimal principal components are different for different datasets. Considering generalization and stability, $k = 16$ is recommended. It should be noted that PCA in preprocessing may lose part of the original hyperspectral data cube information. Therefore, it is necessary to explore new dimensionality reduction methods to reduce the parameters and retain the original information as much as possible in the future.
For the fourth experiment, Figure 8 shows the performance of different convolution kernels and parallel layers used in the MSPN. This experiment did not consider the SE block because it does not contain convolution layers. Based on the single variable principle, we used OA as the evaluation criterion. For the multiscale 3D-CNN, we changed the number of convolutional kernels per layer in the order of 2, 4, 8, and 16 in the case of determining three parallel convolutional layers. We changed the number of parallel convolution layers by copying or removing the middle branching layer in the case of determining eight kernels per layer. For the pyramid pooling module, the setting of variables was similar. The best parameters were determined in the multiscale 3D-CNN having three parallel layers with eight kernels per layer, and the pyramid pooling module having four parallel layers with 128 convolution kernels per layer. It should be noted that all the above determined optimal parameters were employed in the first experiment for comparison, and thus, the best classification performance was achieved.
multiscale 3D-CNN module. (c) The number of kernels in the pyramid pooling module. (d) The number of parallel layers in the pyramid pooling module.

For the fifth experiment, Table 6 lists the classification performance of different methods on all classes, and Figure 9 shows the ground truths and the predicted classification maps of the different models. Figure 10 shows the visualization of the confusion matrix. From these charts, we can see that our proposed model still performs better on the latest high-resolution WHU-Hi-LongKou dataset.

Table 6. Accuracy comparison of different methods for 0.1% samples from the WHU-Hi-LongKou dataset.

| Order | Class              | 2D-CNN | M3D-CNN | HybridSN | R-HybridSN | MSPN |
|-------|--------------------|--------|---------|----------|------------|------|
| 1     | Corn               | 0.95   | 0.95    | 0.98     | 0.95       | 0.99 |
| 2     | Cotton             | 0.91   | 0.83    | 0.93     | 0.75       | 0.94 |
| 3     | Sesame             | 0.93   | 0.97    | 1.00     | 0.71       | 1.00 |
| 4     | Broad-leaf soybean | 0.97   | 0.97    | 0.95     | 0.99       | 0.96 |
| 5     | Narrow-leaf soybean| 0.76   | 0.96    | 0.81     | 0.96       | 0.84 |
| 6     | Rice               | 0.97   | 0.99    | 0.96     | 0.99       | 1.00 |
| 7     | Water              | 0.97   | 0.98    | 1.00     | 1.00       | 1.00 |
| 8     | Roads and houses   | 0.84   | 0.85    | 0.84     | 0.83       | 0.91 |
| 9     | Mixed weed         | 0.91   | 0.93    | 0.86     | 0.90       | 0.86 |
| OA/%  |                    | 95.49  | 96.04   | 96.39    | 96.03      | 97.31|
| Kappa/%|                   | 94.05  | 94.76   | 95.24    | 94.79      | 96.45|
| AA/%  |                    | 84.32  | 84.87   | 86.62    | 87.81      | 89.82|

Figure 9. The predicted classification map for 0.1% samples from the WHU-Hi-LongKou dataset with the method of (a) 2D-CNN, (b) M3D-CNN, (c) HybridSN, (d) R-HybridSN, and (e) MSPN and (f) ground truth.
5. Conclusions

In this paper, to overcome the “small sample problem”, we proposed a multiscale squeeze-and-excitation pyramid pooling network (MSPN) model for hyperspectral image classification. The model includes a multiscale 3D-CNN module, an SE block, and a pyramid pooling module. The multiscale 3D-CNN can better integrate spatial–spectral information, and the SE block automatically relearns the information of the channel. The pyramid pooling module uses four pooling layers of different sizes and the corresponding convolution and deconvolution layers to extract spatial feature information, which can improve the robustness of the model to spatial layout and object variability. We implemented experiments on three public datasets, whereby it was proved that the combination of different multiscale modules can better learn spatial–spectral information in the case of small training samples. The contribution of each module to the model was defined. The pyramid pooling module contains the most convolution layers and multiscale information, therefore contributing the most, followed by the multiscale 3D-CNN. The SE block only plays an auxiliary role in the learning of features and contributes the least. The number of principal components, as well as the number of convolution layers and kernels on the test, was also determined. Overall, $k = 16$ principal components, a multiscale 3D-CNN module with three parallel layers and eight kernels per layer, and a pyramid pooling module with four parallel layers and 128 convolution kernels per layer are preferred. The proposed MSPN model shows competitive advantages in training and testing time and convergence speed and presents a superior performance on limited training samples compared with other similar methods. Compared with the existing methods, the purpose of our model is to utilize various effective modules to retrieve more information from limited training samples. In the future, transfer learning can be explored to improve the proposed model.

Author Contributions: Conceptualization, H.G. and T.M.; methodology, H.G. and Q.L.; software and experiments, H.G. and H.L.; validation, W.W., F.H., and A.T.; writing—original draft preparation, H.G.; writing—review and editing, T.M.; funding acquisition, T.M., C.L., H.D., and Z.H. All authors contributed to the results analysis and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.
**Funding:** This research was funded by the National Natural Science Foundation of China (NSFC) under Grant no. 61775176, in part by the National Major Special Projects of China under Grant GFZX04014308, in part by the Shaanxi Province Key Research and Development Program of China under Grants 2020GY-131 and 2021SF-135, in part by the Fundamental Research Funds for the Central Universities under Grant xjhl20202021, and in part by the Natural Science Foundation of Shang- hai under Grant 18ZR1437200.

**Acknowledgments:** The authors would like to thank M. Graña, MA. Veganzons, and B. Ayerdi for collecting the hyperspectral datasets and free downloads from http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 21 June 2019). 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. Huang, R.; He, M.; Band selection based on feature weighting for classification of hyperspectral data. IEEE Geosci. Remote Sens. Lett. 2005, 2, 156-159, doi: 10.1109/lgrs.2005.844658.
2. Hung, L.; Zhang, L.; Tao, D.; Huang, X.; Du, B. Hyperspectral Remote Sensing Image Subpixel Target Detection Based on Supervised Metric Learning. IEEE Trans. Geosci. Remote Sens. 2013, 52, 4953-4965, doi:10.1109/tgrs.2013.2286195.
3. Luo, B.; Yang, C.; Chanussot, J.; Zhang, L. Crop yield estimation based on unsupervised linear unmixing of multidate hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. 2013, 51, 162-173.
4. He, L.; Li, J.; Plaza, A.; Li, Y. Discriminative Low-Rank Gabor Filtering for Spectral-Spatial Hyperspectral Image Classification. IEEE Trans. Geosci. Remote Sens. 2017, 55, 1381-1395.
5. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vis. 2004, 60, 91-110, doi:10.1023/b:visi.0000029664.99615.94.
6. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20-25 June 2005; pp. 886-893.
7. Pietikäinen, M. Local Binary Patterns. Scholarpedia 2010, 5, 9775, doi:10.4249/scholarpedia.9775.
8. 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.
9. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Hyperspectral image segmentation using a new Bayesian approach with active learning. IEEE Trans. Geosci. Remote Sens. 2011, 49, 3947-3960.
10. Huang, K.; Li, S.; Kang, X.; Fang, L. Spectral-Spatial Hyperspectral Image Classification Based on KNN. Sens. Imaging Int. J. 2016, 17, 1–13, doi:10.1007/s11220-015-0126-z.
11. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans. Geosci. Remote Sens. 2004, 42, 1778–1790, doi:10.1109/tgrs.2004.831865.
12. Arvelyna, Y.; Shiuichi, M.; Atushi, M.; Nguno, A.; Mhopjeni, K.; Muvangua, E. Hyperspectral mapping for rock and alteration mineral with Spectral Angle Mapping and Neural Network classification method: Study case in Warbad district, south of Namibia. In Proceedings of the 2011 IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 24–29 July 2011; pp. 1752–1754.
13. Liciardi, G.; Marpu, P.R.; Chanussot, J.; Benediktsson, J.A. Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles. IEEE Geosci. Remote Sens. Lett. 2012, 9, 447–451.
14. A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, Hyper- spectral image classification with independent component discriminant analysis. IEEE Trans. Geosci. Remote Sens. 2011, 49, 4865–4876.
15. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of hyperspectral images with regularized linear discriminant analysis. IEEE Trans. Geosci. Remote Sens. 2009, 47, 862–873.
16. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. Proc. IEEE 2013, 101, 652–675.
17. Jia, S.; Shen, L.; Li, Q. Gabor feature-based collaborative repre- sentation for hyperspectral imagery classification. IEEE Trans. Geosci. Remote Sens. 2015, 53, 1118–1129.
18. Li, J.; Du, Q.; Li, Y.; Li, W. Hyperspectral image classification with imbalanced data based on orthogonal complement subspace projection. IEEE Trans. Geosci. Remote Sens. 2018, 56, 3838–3851.
19. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
47. Kim, J.H.; Lee, H.; Hong, S.J.; Kim, S.; Park, J.; Hwang, J.Y.; Choi, J.P. Objects Segmentation From High-Resolution Aerial Images Using U-Net With Pyramid Pooling Layers. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 115–119, doi:10.1109/lgrs.2018.2868880.

48. Gao, X.; Sun, X.; Zhang, Y.; Yan, M.; Xu, G.; Sun, H.; Jiao, J.; Fu, K. An End-to-End Neural Network for Road Extraction from Remote Sensing Imagery by Multiple Feature Pyramid Network. *IEEE Access* **2018**, *6*, 39401–39414.

49. Zhong, Y.; Hu, X.; Luo, C.; Wang, X.; Zhao, J.; Zhang, L. WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF. *Remote Sens. Environ.* **2020**, *250*, 112012, doi:10.1016/j.rse.2020.112012.