remote sensing

MDPI

Article

A Multiscale Deep Middle-level Feature Fusion Network for Hyperspectral Classification

Zhaokui Li [1,*], Lin Huang [1] and Jinrong He [2]

[1] School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China; 13667295568@163.com

[2] College of Mathematics and Computer Science, Yan'an University, Yan'an 716000, China; hejinrong@yau.edu.cn

* Correspondence: lzk@sau.edu.cn; Tel.: +86-18809878619

Received: 17 February 2019; Accepted: 19 March 2019; Published: 22 March 2019

Abstract: Recently, networks consider spectral-spatial information in multiscale inputs less, even though there are some networks that consider this factor, however these networks cannot guarantee to get optimal features, which are extracted from each scale input. Furthermore, these networks do not consider the complementary and related information among different scale features. To address these issues, a multiscale deep middle-level feature fusion network (MMFN) is proposed in this paper for hyperspectral classification. In MMFN, the network fully fuses the strong complementary and related information among different scale features to extract more discriminative features. The training of network contains two stages: the first stage obtains the optimal models corresponding to different scale inputs and extracts the middle-level features under the corresponding scale model. It can guarantee the multiscale middle-level features are optimal. The second stage fuses the optimal multiscale middle-level features in the convolutional layer, and the subsequent residual blocks can learn the complementary and related information among different scale middle-level features. Moreover, the idea of identity mapping in residual learning can help the network obtain a higher accuracy when the network is deeper. The effectiveness of our method is proved on four HSI data sets and the experimental results show that our method outperforms the other state-of-the-art methods especially with small training samples.

Keywords: hyperspectral image classification; multiscale; middle-level feature fusion; deep network

1. Introduction

Hyperspectral Imaging (HSI) has hundreds of continuous spectral bands and high spatial correlation, so it contains abundant spectral and spatial information which is useful for the classification of different materials. HSI has been applied to many fields, including environment management [1], geological mapping [2], mineral exploitation [3], and scene recognition [4]. Although HSI contains rich spectral information, it is difficult to obtain enough training samples in practice, which often leads to the "curse of dimensionality". In addition, the neighboring bands of HSI are of high correlation, which means that only a few bands play a critical role. This increases the computational complexity and affects the following classification process. Therefore, dimensionality reduction (DR) is necessary for HSI classification preprocessing, which can reduce computational complexity and retain useful information of classification [5–8]. Feature selection and feature extraction are the traditional methods to implement DR [9]. Feature selection aims to find more discriminative bands from the raw HSI data to represent the entire image and this method can remain the physical meaning of original data [10–15]. Some clustering-based methods [16,17] and ranking-based methods [18,19] find the representative bands to classify distinct classes. Compared with feature selection, feature extraction [20–26] finds

more useful features through mathematical transformation to improve the classification accuracy. These features are, for example, multilinear principal component ananlysis (PCA) [27] and Fisher's linear discriminant analysis [28] etc., but these methods can only extract low-level features which have limited representation capacity to express the abundant information of spectral and spatial features.

Recently, about the above issue, many deep learning models have been proposed and they can learn more distinguished features with the goal of high classification accuracy [29,30]. In the typical deep learning model, stacked autoencoders (SAEs) can extract spatial and spectral information, then combine these features for HSI classification [10]. The potential of deep belief networks (DBN) [31] and restricted Boltzmann machines [32] is used to combine the spatial and spectral information to classify the image. These methods are intended for 1-D input and the input data misses the spatial structure information, which is important for HSI classification. A deep convolutional neural network (CNN) [33] is adopted to get the spatial feature and it has no requirement for the input. 3-D CNN is used to extract spectral-spatial features from the original image directly and gets better classification accuracy. Reference [34] proposes an end-to-end framework to learn the spectral and spatial features and this method can exploit the correlation between the spectral and spatial domains. But in this framework, the input of spectral data is 1-D dimension. It is missing the neighborhood information of spatial dimension. And the classification accuracy of these deep learning models will decrease when the network is deeper. Reference [35] proposes a supervised spectral–spatial residual network and the idea of identity mapping in residual blocks mitigates the decreasing-accuracy phenomenon, but this network firstly learns the spectral features that are used as the input to extract the spatial information, so the spatial features are found from data that has been transformed and so misses the original spatial correlation. Reference [36] applies CNN to extract multiple spatial features and then stacks with spectrum to generate the spectral-spatial feature. This method would have better performance if the spectral feature was extracted by multiscale. Song W proposes a deep fusion feature network [37] for classification. In this network, the features from the low layer, the middle layer and the high layer are respectively extracted by the residual network, and the features of different layers are fused in the fully convoluted layer to classify the image. Although the network considers the influence of different layer features on the classification, it does not consider the spectral-spatial fusion features and it directly extracts the features from the original image. Moreover, the features of fusion at the fully connected layer cannot enable the entire network to fully use the fusion features to learn more discriminative features. Most of the proposed deep learning models now consider the spectral-spatial fusion feature under single scale input and do not consider the abundant correlation between the spectral and spatial in multiscale inputs. Even though some models consider multiscale inputs, however, they cannot guarantee that each scale feature is optimal. Furthermore, these models cannot make full use of the strong complementary and related information among the multiscale fusion features because the features are fused in the fully connected layer to directly classify the image.

To solve these problems and extract more discriminative fusion features, we propose a multiscale deep middle-level feature fusion (MMFN) network for hyperspectral image classification. The training of the network contains two stages: in the first stage, each scale input is used to train a model and the optimal model is saved. The middle-level feature is extracted from the corresponding scale model and it can guarantee the multiscale middle-level features are optimal. In the second stage, the multiscale middle-level features are fused in the convolution layer and the subsequent residual learning block can fully use the strong complementary and related information among multiscale fusion features to extract more discriminative and higher-level features for classification. Furthermore, the residual learning [38] can help the network maintain a higher accuracy when the layer is deeper and make the network more robust.

The three major contributions of this paper include:

(1) The idea of multiscale features fusion is proposed, and this is an idea that contains more abundant neighborhood correlation and low-level features, such as spatial structure, and texture features, which are beneficial for classification.

(2) The training of the network consists of two stages, the first stage obtains the optimal models corresponding to different scales, and extracts the middle-level features under the corresponding scale model. It can ensure the multiscale middle-level features are optimal, which is helpful for the subsequent training stage extracting more discriminative features. The second stage fuses the optimal multiscale middle-level features in the convolutional layer to train a new model for final classification.

(3) Different scale features have strong complementary and related information. Compared with the features that are fused directly in the fully connected layer to classify the image, the multiscale deep middle-level features are fused in the convolutional layer, which can enable the network to make full use of the strong complementary and related information among multiscale fusion features. Moreover, the subsequent residual learning modules can learn the multiscale fusion features to extract more discriminative and higher-level features for classification and can help the network maintain a higher accuracy with deeper layers.

The rest of this paper is organized as follows. Section 2 introduces the detailed architecture of our method. Section 3 presents the results of classification accuracy on the four data sets, and shows the performance of all methods. Finally, the conclusion is provided in Section 4.

2. Methodology

A deep network can be regarded as a process of feature learning, which is a step-by-step abstract representation of the original input through a hidden layer. It can learn the original input data structure and find more useful features. Through feature combination, it transfers the original input into the low-layer features, middle-level features, high-level features up to the final mission objectives. Deep learning through the learning of hierarchical features can extract features from the texture information in the low-level features to the local information in the middle layer to the object information in the high-level layer. From this process, it is not difficult to find the connection between the original input and the low-level features, and the connection between the middle-level layer features and the high-level features, it is difficult to cross directly from the original input to the high-level features. In the MMFN framework, it consists of two training stages. The first stage mainly obtains the optimal model corresponding to each scale, and extracts the features of the last residual block of the corresponding scale in the optimal model. The second stage mainly fuses the multiscale features from the first stage in the convolution layer to train a new model for final classification. Because the multiscale features are extracted from the residual block, which are neither the low-level features that are close to the original image or nor the high-level features that are close to the fully connected layer, the features are defined as the middle-level features in the MMFN network, and the multiscale fusion features are used as inputs to a new model in the second stage to learn more discriminative features for classification.

2.1. Extracting Multiscale Deep Middle-Level Features

HSI data can be denoted as $\mathbf{R} \in \Re^{M \times N \times L}$, $\mathbf{R}_i \in \Re^{M \times N}$ is i the band image, M, N, L denote that the Hyperspectral Image has $M \times N$ pixels, and L bands, respectively. The main purpose of first training stage on MMFN is to extract optimal multiscale deep middle-level features and each scale 3-D data cube is used to train the corresponding model. The model contains a spectral and spatial learning module with different size of convolution filters. Let x be the input of a convolutional layer and x_i is the ith feature map of x. Supposing that the convolutional layer has k filters denoted as W and the bias parameter is b, the j output of the convolutional layer can be represented:

$$\mathbf{z}_j = \sum_{i=1}^{L} R(x_i * W_j + b_j) \qquad j = 1, 2 \ldots ., k \ , \tag{1}$$

The features from the spectral and spatial module are fused as spectral-spatial fusion features, the fusion operation is defined as:

$$Y_{fusion} = g\left\{W_i * [y_{spectral} \oplus y_{spatial}] + b_i\right\}, \tag{2}$$

$$g(x) = \max(0, x), \tag{3}$$

g represents the relu function and it is a rectified linear unit activation function which sets elements with negative numbers to zero. $y_{spectral}$ and $y_{spatial}$ represent the outputs which are found from the spectral and spatial learning module, respectively. The subsequent residual learning module can use the spectral-spatial fusion feature to learn more discriminative features and the structure of residual learning block is showed in Figure 1.

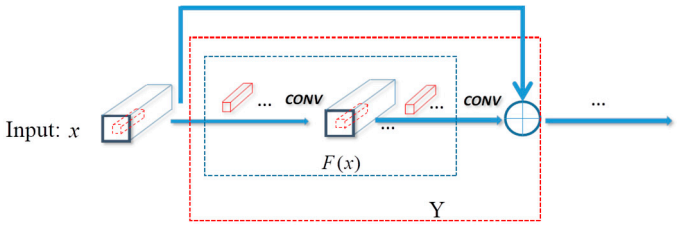


Figure 1. The structure of residual block.

In Figure 1, x represents the input of the first residual block, Y is the function learned through a two convolution layers and it is defined as:

$$Y = F(x, \{W\}) + x, \tag{4}$$

W is the convolutional kernel, F is residual function and can be written as:

$$F = R(R(x * W_1 + b_1) * W_2 + b_2), \tag{5}$$

W_1, W_2 is parameters of the first and second convolution kernel, respectively, b_1, b_2 is the next and the next two layers bias of the input layer, respectively. In residual learning, we use the batch normalization (BN) operation to regularize the learning process for every convolutional operation and BN is formulated as:

$$Y^n = R(y^{n-1} * W^n + b^n), \tag{6}$$

and Y^n represents the output of n th layer after BN operation, W^n, b^n mean the convolutional kernels and bias, respectively on the n th layer. And the y^{n-1} is defined as:

$$y^{n-1} = \frac{Y^{n-1} - E(Y^{n-1})}{Var(Y^{n-1})}, \tag{7}$$

which Y^{n-1} is the output of $(n-1)$ th layer after BN operation. After the residual block layer, the average pooling operation is done for the output of the residual block and the average pooling operation is formulated as:

$$z = \frac{1}{C} \sum_{(i,j) \in S} x_{ij}, \tag{8}$$

We suppose the S is the filter size and C is the number of elements of S, x_{ij} is the value of the corresponding position (i, j) in the input data x, and in this paper, we use global average pooling. After the average pooling, the feature is sent to the softmax layer for HSI classification. The predicted value of the framework is a vector $\hat{y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_c]$, and the truth label vector $y = [y_1, y_2, \ldots, y_c]$, c

is the number of land-cover categories. The parameters of the framework are updated through back propagating the gradients of the cross-entropy objective function which is defined as:

$$cross(\hat{y}, y) = \sum_{i=1}^{L} y_i (\log \sum_{j=1}^{L} e^{\hat{y}_j} - \hat{y}_i), \tag{9}$$

For each scale input data, the above operation is done for training a corresponding scale model. In order to get an optimal model, we use the classification accuracy of a validation set to see whether it is improved in some training epochs to determine whether the model is optimal. Through this method, one can guarantee every scale input corresponds to an optimal model, and the feature from the last residual block is extracted on every model as the deep middle-level feature. These multiscale features are calculated by the weights from the optimal trained model, so these multiscale features are the best, which is helpful for the final classification.

2.2. Fusing Multiscale Deep Middle-Level Features

In the first training stage of MMFN, we can get the optimal multiscale middle-level features and these features have different spatial sizes because of the different scales of inputs. Before fusing these features, the spatial size of features should be same. For example, there have been three different size of features which are F_1, F_2, F_3 respectively, the spatial sizes are 5×5, 7×7, 9×9. For the size of 7×7, 9×9, and we can use 3×3 and 5×5 filters to make the features with same size of 5×5, then these three features are fused. The fusion operation is formulated as:

$$X = g\{W * [F_1 \oplus A_2(F_2) \oplus A_3(F_3)] + b\}, \tag{10}$$

X represents the tensor after fusing the multiscale middle-level features in the convolution layer, A_2, A_3 is the different convolution operations to guarantee the features with the same spatial size. $\oplus$ is concatenating the outputs from the multiscale features, W, b denote the convolutional kernels and bias in the convolution layer respectively. After getting the multiscale middle-level fusion feature, the residual block is used to learn higher-level and discriminative features, which are sent to the softmax layer for the final classification.

2.3. Classifying HSI Based on the MMFN

We take the IN Data Set as an example to describe the architecture of our method in Figures 2 and 3. Figure 2 shows the first training stage of the MMFN, the size of $7 \times 7 \times 200$ as the input data are sent to the spatial learning and spectral learning module with the size of 3×3,128 and 1×1, 128, respectively and the features with the size of $7 \times 7 \times 128$ are obtained. Then these features are concatenated as the spectral-spatial fusion features to do the next convolutional and BN operation, and the size of features is constant with $7 \times 7 \times 128$. In a residual learning module, it contains two residual blocks and every block uses the size of $3 \times 3 \times 128$, 24 filters to extract features from the spectral-spatial fusion feature tensor, the feature size of $5 \times 5 \times 24$ is generated after residual learning and the BN operation is done after every convolutional layer, which can regularize the learning process and improve the classification performance. What the feature tensor gets from the residual block is an input to the average pooling layer and it can obtain a $1 \times 1 \times 24$ vector that is sent to the softmax layer for the final classification. After several epochs of training, we can get the optimal model. For the other scale of $9 \times 9 \times 200$ and $11 \times 11 \times 200$ are the inputs of the network, they are done the same as the above operation and we can get the corresponding scale of the model. From the optimal models corresponding to multiscale inputs, the features, of which the sizes are $5 \times 5 \times 24$, $7 \times 7 \times 24$, $9 \times 9 \times 24$, respectively, are extracted from the last residual block as the deep middle-level features.

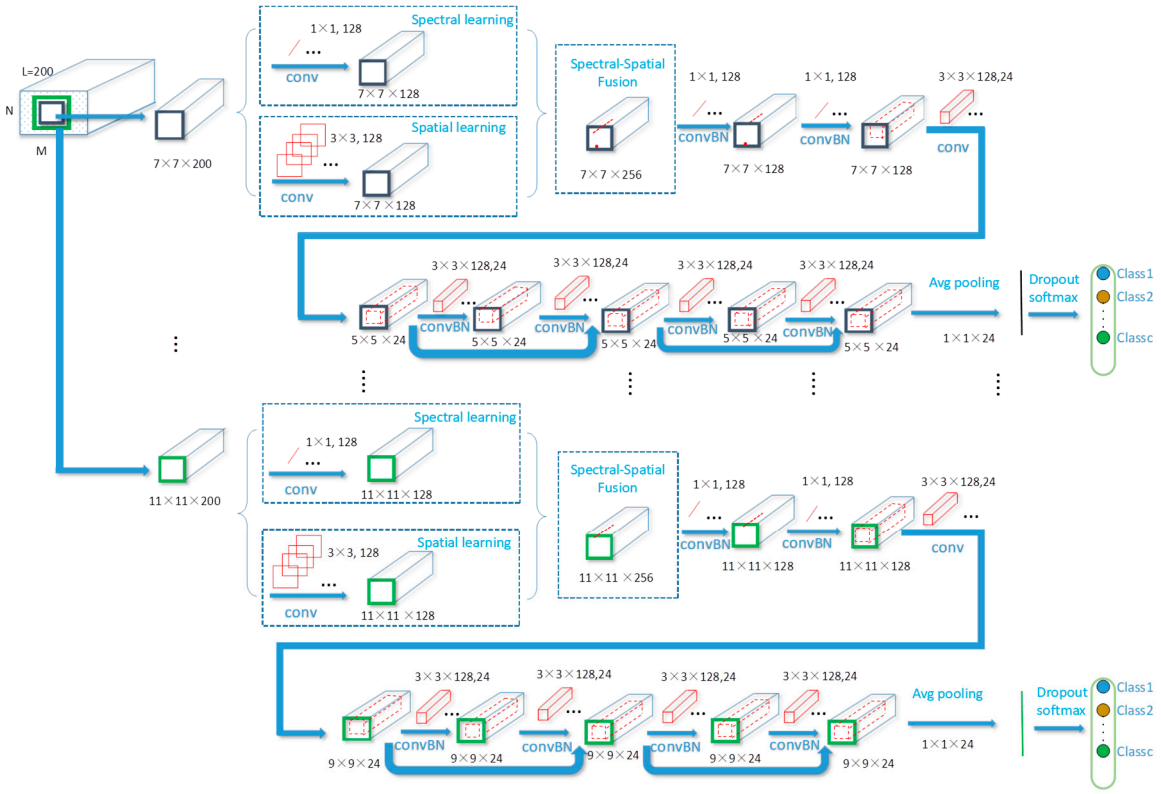


Figure 2. The framework of first training stage in a multiscale deep middle-level feature fusion network (MMFN).

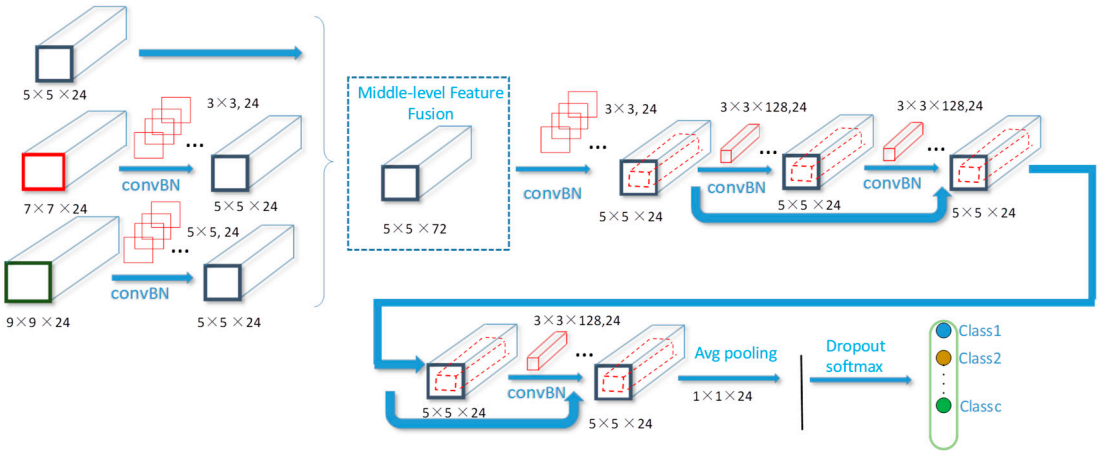


Figure 3. The framework of second training stage in MMFN.

Figure 3 shows the second training stage of the whole network and this stage fuses the multiscale middle-level features and sends them to the residual learning block to learn the higher-level and discriminative features for the final classification. Because each input scale corresponds to a separately trained model, we save the parameters of the optimal model by the classification accuracy of the validation set, and in this way, we can guarantee that the middle-level feature is calculated under the optimal parameters when the features are extracted from the corresponding scale model. It means that the multiscale middle-level features are optimal which can improve the classification performance. The middle-level features have a size of $5 \times 5 \times 24$, $7 \times 7 \times 24$, $9 \times 9 \times 24$, and we can use 3×3, 24 and 5×5, 24 kernels to convolute the features tensors of $7 \times 7 \times 24$, $9 \times 9 \times 24$, respectively and make them with the size of $5 \times 5 \times 24$. Then, these three different scales of middle-level features are concatenated to generate a tensor with the size of $5 \times 5 \times 72$. Through the size of 3×3 filters, the

fused feature with size of $5 \times 5 \times 72$ will be transformed into a tensor with size of $5 \times 5 \times 24$ as an input to the residual learning module which contains two residual blocks and every block consists of 3-D convolution operation with size of $3 \times 3 \times 128$, 24. Finally we can get the higher-level feature with size of $5 \times 5 \times 24$ as the input of the average pooling layer and a tensor with size of $1 \times 1 \times 24$ is generated for the final classification.

3. Experimental Results

3.1. Data Description and Experimental Settings

In this section, the effectiveness of our method is proved in four real-world hyperspectral remote sensing data sets which contain the Indian Pines (IN) Data Set, Pavia University (UP) Data Set, Kennedy Space Center (KSC) Data Set and Salinas Valley (Salinas) Data Set and the proposed method is compared with other state-of-the-art methods. The overall accuracy (OA) and the average accuracy (AA) are the classification metrics used to assess the classification performance of all the methods.

The Indian Pines Data Set (IN) was collected by AVIRIS in 1992 in northwestern Indiana. This commonly used data set has 16 vegetation classes and 224 bands. The spatial size is 145×145 and the spatial resolution is 20 m per pixel. To avoid the negative influence on classification due to water absorption and noise, some bands are discarded and the remaining 200 bands are adopted for analysis.

The Pavia University Data Set (UP) was captured by a Reflective Optics System Imaging Spectrometer optical sensor over an urban area surrounding the University of the Pavia. The image is of size $610 \times 340 \times 115$ with a resolution of 1.3 m per pixel and 9 urban land-cover classes are considered in this experiment. The number of remaining bands is 103 after discarding the useless bands.

The KSC Data Set was collected by AVIRIS in 1996 in Florida, and contains 512×614 pixels with spatial resolution of 18 m per pixel and the ground-truth classes are 13. After removing the noise bands, 176 bands are retained and used for the experiment.

The Salinas Data Set which was gathered by AVIRIS and it consists of 224 bands with spatial size of 512×217 pixels. The spatial resolution of the data is 3.7 m per pixel and the ground-truth classes are 16. Twenty noisy bands are removed, and 204 bands are left for the next experiment.

The information of data sets is shown in Tables 1–4. The corresponding the false-color image and ground-truth map are shown in Figures 4–7. For all data sets, the number of experiments was twenty times to reduce the influence of random effects, which are caused by randomly choosing different training samples every time. Through verifying whether the accuracy of validation set is improved in some epochs to determine whether the network model is optimal, the optimal weight values of each model were saved. We made the average effects the final results to evaluate the classification accuracy of every method. We evaluated the performance of all methods on the small training samples to prove that our proposed MMFN has strong robustness and generalization. Furthermore, the MMFN had a better performance on classification accuracy when training samples were small. In four data sets which contained IN, UP, KSC and Salinas, we split the data into training, validation, testing set, and the ratio was 5%, 10%, 85%, respectively on these four data sets.

Table 1. Land cover classes and numbers of samples in the Indian Pines data set.

No.	Class Name	Numbers of Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-tree	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93
	Total	10,249

Table 2. Land cover classes and numbers of samples in the Pavia University data set.

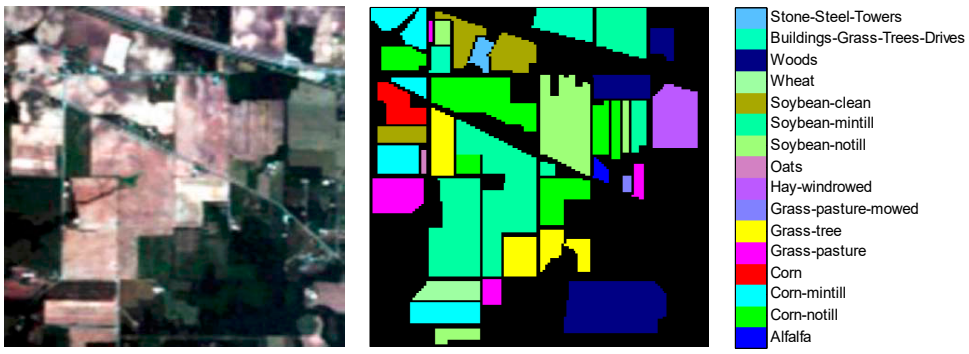
No.	Class Name	Numbers of Samples
1	Asphalt	6631
2	Meadows	18,649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947
	Total	42,776

Table 3. Land cover classes and numbers of samples in the Kennedy Space Center (KSC) data set.

No.	Class Name	Numbers of Samples
1	Scrub	761
2	Willow swamp	243
3	CP hammock	256
4	Slash pine	252
5	Oak/Broadleaf	161
6	Hardwood	229
7	Grass-pasture-mowed	105
8	Graminoid marsh	431
9	Spartina marsh	520
10	Cattail marsh	404
11	Salt marsh	419
12	Mud flats	503
13	Water	927
	Total	5211

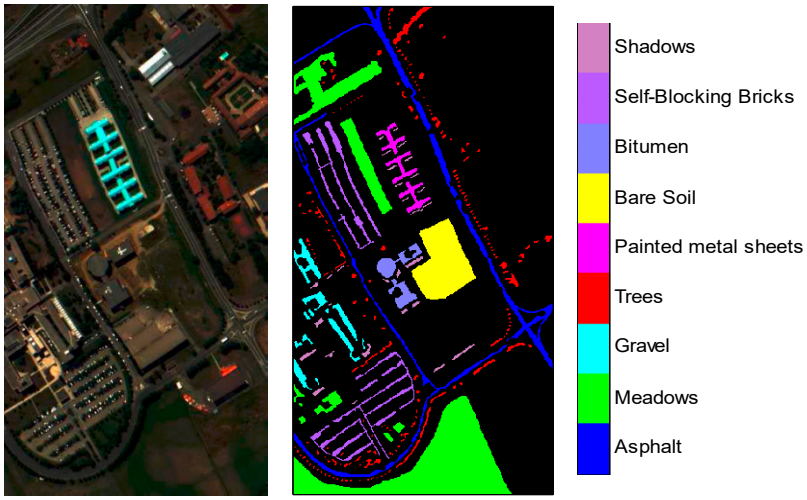
Table 4. Land cover classes and numbers of samples in the Salinas data set.

No.	Class Name	Numbers of Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11,271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807
	Total	54,129



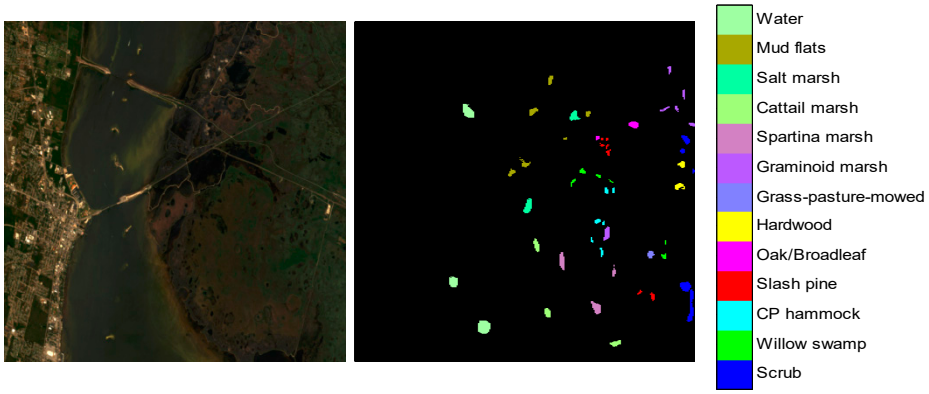
(a) (b)

Figure 4. (**a**) False-color image of the Indian Pines data, (**b**) Ground truth of the Indian Pines data.



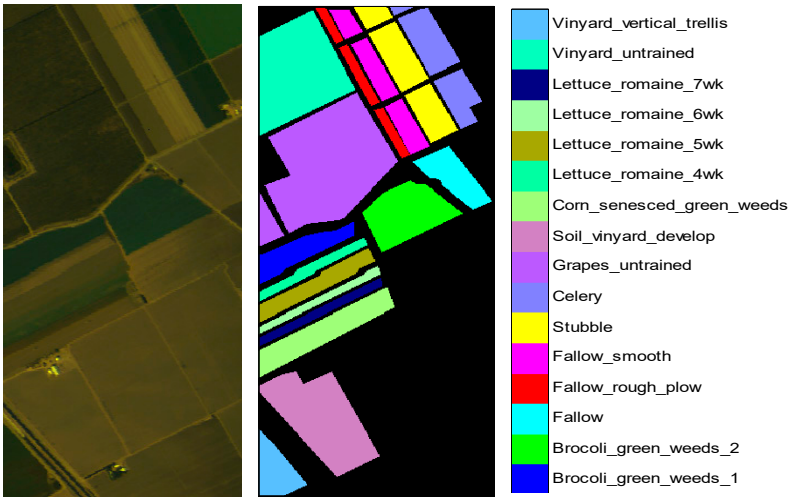
(a) (b)

Figure 5. (**a**) False-color image of the Pavia University data, and (**b**) Ground truth of the Pavia University data.



(a) (b)

Figure 6. (**a**) False-color image of the Ground-truth map of the KSC data, and (**b**) Ground truth of the KSC data.



(a) (b)

Figure 7. (**a**) False-color image of the Salinas data, (**b**) Ground truth of the Salinas data.

In our implementation, the training epoch was set to 100 and the optimizer adopted the standard stochastic gradient descent method. The batch size was set to 64, the optimum learning rates in IN, UP, KSC, Salinas data set were fixed as 0.0003, 0.0001, 0.0001, 0.0003, respectively, and the momentum was set to 0.9.

The proposed method was compared with some state-of-the-art methods including the SVM [39], ResNet [38], SAE [10], 3-D CNN [23] and Two-CNN [34], SSRN [35]. In order to compare fairly, we used the SVM using spatial information through the Gaussian filter. The framework of ResNet adopts the same residual blocks as our method and it does not contain the spectral-spatial learning module.

3.2. Influence of Parameters

3.2.1. The Selection of Multiscale Inputs

The spatial scale of input data was changed on the four data sets, and the appropriate multiscale inputs were determined through the classification accuracy achieved by the different scale inputs. In the four data sets, the data was split into 5%, 10% and 85% to comprise the training set, validation sets, and test sets, respectively. The classification results of different spatial scale inputs are shown in Table 5.

In most cases, as the spatial scale of the input became larger, the classification accuracy achieved higher results on the four data sets. It was proved that when the spatial scale was within a certain range, the input data contained more spatial structure information with a larger spatial scale, which was helpful for the network learning more discriminative features and obtaining a higher classification accuracy.

Table 5. Classification results (OA%) of our methods on the four data sets with different input sizes.

Spatial Size	IN	UP	KSC	Salinas
3×3	78.64	94.24	86.34	95.17
5×5	84.25	96.49	90.29	96.29
7×7	87.32	98.53	95.21	97.42
9×9	90.42	99.17	95.89	98.27
11×11	94.15	99.39	96.76	99.24
13×13	94.87	99.40	96.93	99.28
15×15	94.94	99.43	97.14	99.31

From Table 5, when the spatial scale of input was larger than or equal to 11×11, the improvement of classification accuracy on the four data sets was relatively small and basically stable. And when the spatial scale of the input was 7×7, the classification results on the four data sets reached a higher accuracy, which had obvious advantages over the classification accuracy achieved by 3×3 and 5×5 as spatial input scales. Therefore, in order to select a relatively small spatial scale of input and achieve a higher classification accuracy, the selection of spatial scales were 7×7, 9×9, 11×11 as the multiscale inputs of the whole network.

3.2.2. The Effectiveness of Multiscale Inputs

In order to validate the suggestion that multiscale inputs were more beneficial for HSI classification than a single input, some experiments were done on these four data sets. The experiment results are showed in Table 6, and the number of training samples for the four data sets ranges from 3% to 6% for each class, and the classification accuracy is evaluated by the overall classification accuracy (OA). It can be seen from the Table 6 that the input with a large spatial scale has a higher classification accuracy than the input with a small spatial scale. Although the number of training samples increased the classification accuracy of the input with large spatial scale and the input with small spatial scale were both improving, however the accuracy of larger spatial scale input was higher in all cases than the smaller spatial scale input. When the training samples of the four data sets were 3%, the classification accuracy obtained by multiscale inputs was higher than the single scale input. Especially in the IN data set, which was difficult to classify, the classification accuracy achieved by the multiscale inputs had an obvious advantage compared with the single scale input. From the Table 6, the bolded classification accuracies of multiscale inputs are higher than the single input, no matter how many training samples in the four data sets. It proves the multiscale inputs are more useful than single input for classification.

Table 6. Classification results (OA%) of our methods on the four data sets with different input sizes when the percentage of training samples is changing.

Data Set	Size of Input	3%	4%	5%	6%
IN	7×7	86.57	86.89	87.32	91.24
	9×9	86.94	88.14	90.42	93.6
	11×11	88.06	91.26	94.15	94.63
	$7 \times 7, 9 \times 9, 11 \times 11$	**91.81**	**93.87**	**96.59**	**97.16**
UP	7×7	96.04	97.38	98.53	98.87
	9×9	96.17	98.48	99.17	99.24
	11×11	98.68	98.88	99.39	99.43
	$7 \times 7, 9 \times 9, 11 \times 11$	**99.4**	**99.73**	**99.75**	**99.84**
KSC	7×7	92.21	93.04	95.21	95.86
	9×9	92.79	94.06	95.89	96.52
	11×11	94.62	95.48	96.76	97.48
	$7 \times 7, 9 \times 9, 11 \times 11$	**97.15**	**98.24**	**98.50**	**98.97**
salinas	7×7	94.75	95.78	97.42	97.7
	9×9	96.54	98.04	98.27	98.62
	11×11	97.25	99.2	99.24	99.33
	$7 \times 7, 9 \times 9, 11 \times 11$	**98.37**	**99.65**	**99.69**	**99.73**

Although the classification accuracy did not improve significantly in the UP and KSC data sets, and the single scale input and multiscale inputs were both reaching a higher accuracy because the UP and KSC data sets both have a higher spatial resolution, the advantages of multiscale inputs were not obvious, but in most cases the classification performance of multiscale inputs had better generalization and effectiveness with small labeled samples. Multiscale inputs of the network can generate multiscale spectral-spatial fusion features that contain abundant the correlation between spatial and spectral, spatial structure information and texture information compared with the spectral-spatial fusion feature generated by single scale input. This information can help the network learn more discriminative features for better classification. With increase of training samples, the classification accuracy was improved in most cases, and the multiscale inputs achieved better classification results than single scale input. The experimental results also proved that the idea of multiscale inputs was more suitable for deep network classification and could improve the final classification accuracy.

3.2.3. The Selection of Number of Residual Block

In the second stage of the MMFN, the residual learning module was used to learn higher-level features from the middle-level fusion features. The selection of the number of residual blocks in the network was determined by experiments in this section. In four data sets that were IN, UP, KSC, Salinas, the suitable number of residual blocks was selected through the classification accuracy achieved by changing the number of residual blocks and training samples. The result is shown in Table 7. It can be seen that by increasing the number of training samples, the classification accuracy was improved regardless of the number of residual blocks in the four data sets. It was also proved that more labeled samples are helpful for improving classification results. In comparing the different number of residual blocks in the case of the same training samples, the classification accuracy was higher when the network contained two residual blocks than zero, one and three residual blocks on the four data sets.

Table 7. The number of residual blocks selection experiment.

Data Set	The Number of Residual Block	3%	4%	5%	6%
IN	no residual block	88.44	90.28	93.37	95.59
	one residual block	91.23	92.65	94.17	96.71
	two residual block	**91.81**	**93.87**	**96.59**	**97.16**
	three residual block	91.34	93.16	95.51	96.66
UP	no residual block	99.07	99.25	99.32	99.49
	one residual block	**99.59**	99.64	99.67	99.72
	two residual block	99.40	**99.73**	**99.75**	99.84
	three residual block	99.12	99.68	99.71	**99.86**
KSC	no residual block	93.65	96.21	97.02	97.95
	one residual block	97.06	98.03	98.12	98.61
	two residual block	**97.15**	**98.24**	**98.5**	**98.97**
	three residual block	96.91	97.94	97.98	98.44
salinas	no residual block	96.26	96.89	97.31	98.28
	one residual block	97.79	98.68	99.28	99.31
	two residual block	**98.37**	**99.65**	**99.69**	99.73
	three residual block	98.25	99.36	99.44	**99.75**

Although on the UP and Salinas data sets, when the number of training samples reaches 6%, the network contains two residual blocks and the classification accuracy is slightly lower than the accuracy of three residual blocks, because the UP and Salinas data sets have a high spatial resolution with 1.3 m and 3.7 m respectively, which helps the network to achieve a high accuracy even with small training samples. So, when the number of training samples is 6% per class, the MMFN has an excellent performance regardless of the network contains zero, one, two or three residual blocks. The results of classification accuracy show that the network contained residual blocks and achieved higher accuracy than the network which did not use residual block in the four data sets and it proves that the residual block can help the network to improve classification accuracy. In most cases, the classification performance was better than other number of residual blocks when the network contained two residual blocks from the bolded classification accuracies. The result also shows that the network layer was shallow with one residual block, and the features that may be learned were not discriminative, and the three residual blocks fused features from too many layers which may introduce too much redundant information resulting in a reduction of classification accuracy. Through the experimental results, we chose two residual blocks to form the structure of MMFN in the second training stage.

3.3. Experiment Results and Analysis

In order to prove the superiority of the proposed network MMFN in the case of small label samples, we compared MMFN with other state-of-the-art methods on the four data sets, and the classification results are shown in Figure 8. Changing the number of training samples from 3% to 6% each class. Figure 8a shows the classification performance of each method on the IN data set. It can be seen from the Figure 8a that MMFN has a distinct advantage over other methods when the number of training samples was 3%, it showed that the MMFN network can learn more discriminative features to help with classifying the image even with small training samples. Figure 8b is the classification result of each method on the UP data set. Although the classification accuracy curve of the MMFN network was close to the curves of the SSRN when the number of training samples increased. However, when the number of training samples was small, the classification performance of the MMFN network was better than other methods. Figure 8c shows the performance of all methods in the KSC data set.

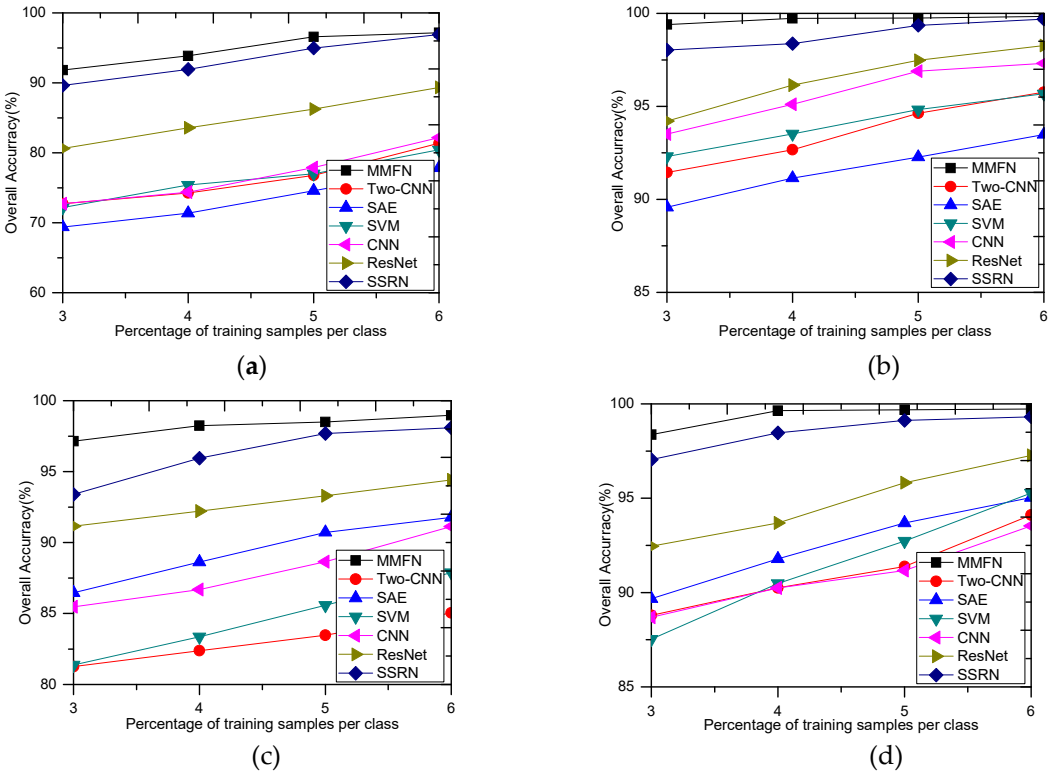


Figure 8. The Overall Accuracy of changing the percentage of training samples by all methods on the four data sets. (**a**) Overall Accuracy on IN. (**b**) Overall Accuracy on UP. (**c**) Overall Accuracy on KSC. (**d**) Overall Accuracy on Salinas.

MMFN has obvious advantages over other methods at most cases. It shows that the optimal middle-level features are helpful for the second training stage extracting more discriminative features, and the multiscale middle-level features are fused in the convolution layer can make the network to learn strong complementary and related information among multiscale features. Figure 8d is the classification result on the Salinas data set. Because the spatial resolution of the Salinas data set is high, the accuracies achieved by the MMFN, and SSRN networks are high in the case of small training samples, however it can be seen from the figure that MMFN still has obvious advantages in classification.

From the experimental results of the four data sets, the MMFN network fused the extracted multiscale middle-level features in the convolutional layer, which helped the residual network to learn more discriminative and higher-level features, however in Two-CNN, the fusion features were fused in the fully connected layer and this made the network use these fusion features only in the classification, which may have reduced the classification accuracy.

Table 8 shows the classification accuracies of different methods on the four data sets which contain IN, UP, KSC, and Salinas. Tables 9–12 list the class-specific accuracies of different methods on the four data sets. The training set, validation set, and test set are split into 5%, 10%, and 85%, respectively. It can be seen from the bolded classification accuracied in these tables that MMFN performs the better than other methods in OA and AA in most cases, and it proves the effectiveness of the network. MMFN achieved higher classification accuracy than the ResNet network on the four data sets, because the MMFN made full use of the multiscale middle-level features, and sent the fused features to the residual block instead of learning directly from the original image like ResNet.

Table 8. Classification accuracies (%) of different methods on the four data sets.

Data Set	Method	OA	AA
IN	Two-CNN	76.78 ± 0.47	75.23 ± 0.56
	SAE	74.58 ± 0.56	75.27 ± 0.62
	SVM	77.01 ± 0.54	68.73 ± 0.53
	CNN	77.89 ± 0.43	77.14 ± 0.36
	ResNet	86.25 ± 0.37	85.93 ± 0.42
	SSRN	94.97 ± 0.31	94.52 ± 0.22
	MMFN	**96.59 ± 0.26**	**96.13 ± 0.25**
UP	Two-CNN	94.63 ± 0.27	93.31 ± 0.22
	SAE	92.27 ± 0.35	92.58 ± 0.29
	SVM	94.83 ± 0.28	93.37 ± 0.32
	CNN	96.89 ± 0.17	96.75 ± 0.25
	ResNet	97.48 ± 0.20	97.03 ± 0.23
	SSRN	99.42 ± 0.26	99.05 ± 0.32
	MMFN	**99.75 ± 0.19**	**99.68 ± 0.23**
KSC	Two-CNN	83.47 ± 0.38	84.32 ± 0.43
	SAE	90.72 ± 0.35	89.51 ± 0.38
	SVM	84.37 ± 0.41	82.18 ± 0.37
	CNN	88.65 ± 0.37	86.39 ± 0.42
	ResNet	93.29 ± 0.31	92.89 ± 0.36
	SSRN	97.68 ± 0.43	97.17 ± 0.36
	MMFN	**98.50 ± 0.29**	**98.32 ± 0.31**
salinas	Two-CNN	91.38 ± 0.36	89.74 ± 0.43
	SAE	93.69 ± 0.41	93.21 ± 0.47
	SVM	92.73 ± 0.44	91.58 ± 0.48
	CNN	91.17 ± 0.37	92.16 ± 0.40
	ResNet	95.82 ± 0.30	94.56 ± 0.34
	SSRN	99.22 ± 0.26	99.31 ± 0.33
	MMFN	**99.69 ± 0.31**	**99.43 ± 0.31**

Table 9. Class-specific accuracies (%) of the Indian Pines Data Set.

Class	Training/Test	Two-CNN	SAE	SVM	CNN	ResNet	SSRN	MMFN
1	2/44	73.88	75.25	30.05	83.54	83.24	92.5	**94.63**
2	71/1357	57.71	68.67	52.11	66.48	63.19	**96.45**	95.3
3	42/788	60.28	62.51	60.17	65.85	85.34	**98.6**	95.11
4	12/225	80.92	62.48	71.22	85.47	89.15	93.13	**99.28**
5	24/459	74.09	81.42	69.19	85.46	94.94	**99.27**	98.8
6	37/657	77.46	81.76	88.63	81.54	94.34	**99.84**	98.73
7	1/27	80.0	73.64	40.21	76.49	70.5	60.32	**82.77**
8	24/454	86.45	85.38	81.54	82.67	95.5	**100**	**100**
9	1/19	100	78.26	49.46	81.43	90.5	94.36	**100**
10	49/923	62.22	67.57	76.92	71.62	85.3	90.71	**95.68**
11	123/2332	67.81	72.52	76.92	62.69	94.47	95.5	**98.41**
12	30/563	59.09	71.35	57.44	65.83	65.8	96.37	**98.54**
13	10/195	86.49	83.64	92.82	84.38	94.89	**100**	96.62
14	63/1202	82.77	80.35	96.92	83.51	94.67	**99.9**	99.53
15	19/367	88.46	83.75	80.92	86.94	95.2	95.37	**97.59**
16	5/88	66.19	75.89	76.13	70.36	77.91	**100**	87.23
OA	-	76.78	74.58	77.01	77.89	86.25	94.97	**96.59**
Kappa	-	0.7155	0.7234	0.7365	0.7628	0.8425	0.9392	**0.9458**
AA	-	75.23	75.27	68.73	77.14	85.93	94.52	**96.13**

Table 10. Class-specific accuracies (%) of the UP Data Set.

Class	Training/Test	Two-CNN	SAE	SVM	CNN	ResNet	SSRN	MMFN
1	332/6299	93.27	94.59	93.36	98.45	99.82	**100**	99.67
2	932/17717	97.14	96.44	94.67	96.37	99.62	**99.98**	99.96
3	105/1994	85.46	84.57	83.92	96.21	94.88	94.97	**99.88**
4	153/2911	98.58	97.37	97.58	97.58	96.24	98.92	**99.22**
5	67/1278	99.56	99.6	99.92	96.36	95	99.82	**100**
6	251/4778	90.58	93.39	93.88	96.51	99.83	98.74	**99.72**
7	67/1263	88.34	88.57	94.67	97.23	99.3	**100**	**100**
8	184/3498	87.17	85.66	93.57	93.47	93.73	**99.02**	98.76
9	47/900	99.74	93.04	94.05	98.57	94.87	**100**	**100**
OA	-	94.63	92.27	94.83	96.89	97.48	99.42	**99.75**
Kappa	-	0.9216	0.9235	0.9314	0.9647	0.9853	0.9943	**0.9972**
AA	-	93.31	92.58	93.37	96.75	97.03	99.05	**99.68**

Table 11. Class-specific accuracies (%) of the KSC Data Set.

Class	Training/Test	Two-CNN	SAE	SVM	CNN	ResNet	SSRN	MMFN
1	38/723	90.12	93.14	99.17	92.45	**100**	**100**	**100**
2	12/231	91.28	92.04	89.61	90.48	95.46	**99.49**	99.47
3	12/244	90.22	85.49	90.76	87.16	94.63	**100**	98.29
4	13/239	80.8	72.02	63.59	80.38	91.44	91.54	98.7
5	8/153	75.55	82.3	59.63	80.96	78.91	**85.78**	89.46
6	11/218	91.66	83.15	67.43	83.21	90.91	**100**	96.96
7	5/100	80.59	76.46	58.31	83.26	83.11	92.63	**93.69**
8	22/409	93.99	94.1	89.66	88.26	94.2	97.12	**99.46**
9	26/494	87.91	94.57	90.39	89.04	94.01	98.67	**100**
10	20/384	71.02	98.91	89.84	84.85	95.55	99.12	**100**
11	21/398	71.62	98.14	93.96	83.89	94.73	96.73	**98.26**
12	25/478	89.18	96.42	84.31	90.43	98.12	**100**	**100**
13	46/881	82.24	97.83	91.71	88.7	96.55	**100**	**100**
OA	-	83.47	90.72	84.37	88.65	93.29	97.68	**98.5**
Kappa	-	0.8104	0.9121	0.8162	0.8657	0.9391	0.9534	**0.9737**
AA	-	84.32	89.51	99.17	86.39	92.89	97.17	**98.32**

Table 12. Class-specific accuracies (%) of the Salinas Data Set.

Class	Training/Test	Two-CNN	SAE	SVM	CNN	ResNet	SSRN	MMFN
1	100/1909	93.94	92.74	94.73	94.67	**100**	**100**	**100**
2	186/3540	93.46	95.13	94.32	95.52	96.27	**100**	**100**
3	99/1877	90.9	93.62	90.48	92.74	93.83	**100**	99.67
4	70/1364	90.58	95.6	88.39	94.61	92.73	**99.68**	98.77
5	134/2544	92.5	94.39	95.27	94.69	**100**	**100**	**100**
6	198/3761	91.81	95.73	93.75	94.82	96.48	99.97	**100**
7	179/3400	94.14	96.47	95.66	97.28	94.37	**100**	**100**
8	564/10707	73.57	91.58	84.38	87.69	93.34	**99.86**	99.31
9	310/5893	89.75	95.46	94.37	90.61	94.87	99.84	**100**
10	164/3114	88.77	85.27	87.46	86.54	94.01	**99.86**	99.83
11	53/1015	86.92	92.52	93.75	89.48	92.24	**97.91**	96.48
12	96/1831	93.35	93.74	94.31	92.74	94.88	99.77	**99.94**
13	46/870	92.41	92.73	93.64	90.68	92.92	**100**	**100**
14	54/1016	92.71	95.26	94.31	93.17	94.79	**100**	**100**
15	363/6905	77.78	86.47	76.75	85.72	89.86	91.91	**99.43**
16	90/1717	93.32	94.71	93.76	93.74	92.36	**100**	99.84
OA	-	91.38	93.69	92.73	91.17	95.82	99.22	**99.69**
Kappa	-	0.9083	0.9171	0.9133	0.9058	0.9606	0.9733	**0.9961**
AA	-	89.74	93.21	91.58	92.16	94.56	99.31	**99.43**

This also shows that the optimal middle-level feature obtained by each scale input in the first stage of MMFN was beneficial for classification. This also proves the validity of extracting middle-level features in MMFN. Compared with the idea of spectral-spatial fusion in SSRN, MMFN introduces the idea of multiscale inputs, which provides the network with abundant complementary and related

information among different scale features, and the spectral and spatial learning module in MMFN is based on original image, it can extract more primitive and accurate spatial structure information.

The input of the spatial learning module in the SSRN network is based on the features extracted from the spectral learning module, the spatial learning will miss the spatial information in the original image, so the classification accuracy is lower than the MMFN. It can be seen from the table that the variance value of MMFN network classification result is smaller than other methods in most cases, which shows the stability of the network.

The training and testing times provide a direct measure of computational efficiency for MMFN. All experiments were conducted on an HP z620 workstation with GT 980Ti graphical processing unit (GPU). The loss function values on the four data sets were 0.2520(IN),0.2318(UP),0.0653(KSC) and 0.1457(Salinas), respectively. Table 13 shows the results of training and test times for all methods on four different data sets. The training set, validation set, and test set of all methods on the four data sets were split into 5%, 10%, and 85%, respectively. Two-CNN, ResNet, SAE, CNN, SSRN, MMFN were iterated 100 times and the SVM was trained 20 times. It can be seen from Table 13 that the training of the MMFN network takes the longest time, because the training of the network is divided into two phases, and the network has multiscale inputs to increase the computational time, although MMFN is longer than the training time of SSRN about 1–6 minutes in the larger data sets such as UP and Salinas, but when MMFN has small labeled samples, the classification accuracy of the network is higher than SSRN, especially in the IN data set that is difficult to classify, the advantage of the MMFN is obvious. In other methods, ResNet contains two residual blocks, so the training time of it is longer than Two-CNN, SAE, and CNN, but its classification accuracy is higher than these methods.

Table 13. Comparison of training and test time for each method on four data sets.

		IN	UP	KSC	Salinas
Two-CNN	Train (m)	3.6	13.1	2.1	12.7
	Test (s)	5.2	15.6	2.1	16.9
SAE	Train (m)	3.5	12.8	1.7	12.3
	Test (s)	3.8	15.4	1.6	18.4
SVM	Train (m)	1.6	8.0	0.8	8.4
	Test (s)	1.4	9.7	0.8	10.3
CNN	Train (m)	3.6	14.9	1.9	13.1
	Test (s)	4.8	18.6	2.0	27.2
ResNet	Train (m)	4.0	16.3	2.2	14.2
	Test (s)	4.7	18.4	2.3	26.5
SSRN	Train (m)	5.0	18.5	2.6	19.4
	Test (s)	6.3	24.7	3.2	36.4
MMFN	Train (m)	8.2	23.5	3.8	25.4
	Test (s)	6.7	25.6	3.5	37.6

3.4. Discussions

In this section, we briefly discuss the experimental results presented earlier. First, we found that the performance of MMFN model on all four data sets was generally better than other models. There are three possible reasons for such a performance improvement: 1) the multiscale model effectively fuses more abundant neighborhood correlation and low-level feature. 2) the middle-level features fusion structure can better exploit strong complementary and related information among multiscale fusion features than a high-level features fusion structure. 3) the residual learning modules can extract more discriminative and higher-level features and make deep learning models much easier to train. As can be seen in Figure 8, a residual-based network model achieved a better performance on all four data sets compared with SAE, CNN and Two-CNN. As can be seen in Figure 8 and Table 8, MMFN achieved higher classification accuracy than the ResNet and SSRN on the four data sets. MMFN made full use of the multiscale middle-level features, and sent the fused features to the residual block instead of learning directly from the original image like ResNet. SSRN also adopted residual connections, and

treated spectral features and spatial features separately in two consecutive blocks, however, if the input of the spatial block is based on the spectral block, the spatial learning will miss the spatial information.

Second, two aspects will influence the HSI classification accuracy: 1) the number and spatial size of input into the network; 2) the number of training labeled samples. MMFN uses the suitable number and spatial size of inputs through the experiments and achieves a higher accuracy. Multiscale inputs with relatively larger spatial size contained more useful and abundant information which can boost the classification performance. MMFN performs better with relatively small labeled samples and this network can be generalized to other remote-sensing scenarios because of its deep feature learning capacity.

Finally, the disadvantage of the MMFN model is that the training time is relatively long, which is mainly because the training of the network is divided into two stages, and the multiscale input increases the corresponding time. As can be seen in Table 13, the training time of MMFN is about 1–6 minutes longer than that of SSRN and 2–12 minutes longer than that of CNN, which means that MMFN is more computationally expensive than the SSRN and the CNN. Fortunately, the adoption of GPU has largely alleviated the extra computational costs and reduced the training times.

4. Conclusions

In this paper, a novel MMFN deep learning method is proposed for hyperspectral image classification. Compared with the previous networks, MMFN consists of two independent training stages. The first stage is to fuse the spectral-spatial information and effectively extract the spectral-spatial fusion feature. The second stage fuses the complementary yet correlated information among the regions of different scales and effectively extracts the multiscale fusion feature, which can further improve the classification accuracy. In addition, multiscale deep middle-level feature fusion network works better for spectral–spatial feature learning as compared to single scale network. The multiscale deep middle-level features are fused in the convolutional layer rather than the fully connected layer, which can enable the network to make full use of the strong complementary and related information among multiscale fusion features. Finally, the residual-based network model achieved a better performance on all four data sets. The residual learning modules can extract more discriminative features for classification and can help the network maintain a higher accuracy with deeper layers. The experimental results show the superiority of the proposed method with small labeled samples on the four data sets over other state-of-the-art methods. In future work, we can research a better fusion strategy and fuse different data sets from other sensors to improve the classification accuracy.

Author Contributions: Conceptualization, Z.L.; Methodology, Z.L., L.H. and J.H.; Writing – original draft, Z.L., L.H. and J.H.

Funding: This research was funded by the Natural Science Foundation of Liaoning, grant number 20180550337, the China Postdoctoral Science Foundation, grant number No. 2018M633585, and the Natural Science Basic Research Plan in Shaanxi Province of China, grant number No.2018JQ6060.

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

References

1. Laurin, G.V.; Chan, J.C.W.; Chen, Q.; Lindsell, J.A.; Coomes, D.A.; Guerriero, L.; Del Frate, F.; Miglietta, F.; Valentini, R. Biodiversity mapping in a tropical West African forest with airborne hyperspectral data. *PLoS ONE* **2014**, *9*, e97910.
2. Van der Meer, F.D.; Van der Werff, H.M.; Van Ruitenbeek, F.J.; Hecker, C.A.; Bakker, W.H.; Noomen, M.F.; van der Meijde, M.; Carranza, E.J.M.; de Smeth, J.B.; Woldai, T. Multi-and hyperspectral geologic remote sensing: A review. *Int. J. Appl. Earth Observ. Geoinf.* **2012**, *14*, 112–128. [CrossRef]
3. 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]

4. Lu, X.; Li, X.; Mou, L. Semi-Supervised Multitask Learning for Scene Recognition. *IEEE Trans. Cybern.* **2015**, *45*, 1967–1976. [PubMed]
5. Zhong, P.; Zhang, P.; Wang, R. Dynamic learning of SMLR for feature selection and classification of hyperspectral data. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 280–284. [CrossRef]
6. Khodr, J.; Younes, R. Dimensionality reduction on HSIs: A comparative review based on artificial datas. In Proceedings of the 4th 2011 International Congress on Image and Signal Processing, Shanghai, China, 15–17 October 2011; pp. 1875–1883.
7. Plaza, A.; Martinez, P.; Plaza, J.; Perez, R. Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 466–479. [CrossRef]
8. Lu, X.; Zheng, X.; Yuan, Y. Remote Sensing Scene Classification by Unsupervised Representation Learning. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5148–5157. [CrossRef]
9. Serpico, B.S.; Bruzzone, L. A new search algorithm for feature selection in hyperspectral remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *39*, 1360–1367. [CrossRef]
10. Tao, C.; Pan, H.; Li, Y.; Zou, Z. Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2438–2442.
11. 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 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–12.
12. Persello, C.; Bruzzone, L. Kernel-Based Domain-Invariant Feature Selection in Hyperspectral Images for Transfer Learning. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 2615–2626. [CrossRef]
13. Imbiriba, T.; Bermudez, J.C.; Richard, C. Band selection for nonlinear unmixing of hyperspectral images as a maximal clique problem. *IEEE Trans. Image Process.* **2017**, *99*, 2179–2191. [CrossRef] [PubMed]
14. Yuan, Y.; Zheng, X.; Lu, X. Discovering Diverse Subset for Unsupervised Hyperspectral Band Selection. *IEEE Trans. Image Process.* **2016**, *26*, 51–64. [CrossRef]
15. Damodaran, B.B.; Courty, N.; Lefèvre, S. Sparse Hilbert Schmidt Independence Criterion and Surrogate-Kernel-Based Feature Selection for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2385–2398. [CrossRef]
16. Yuan, Y.; Lin, J.; Wang, Q. Dual-Clustering-Based Hyperspectral Band Selection by Contextual Analysis. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1431–1445. [CrossRef]
17. Jia, S.; Ji, Z.; Qian, Y.; Shen, L. Unsupervised band selection for hyperspectral imagery classification without manual band removal. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2012**, *5*, 531–543. [CrossRef]
18. Chang, C.I.; Wang, S. Constrained band selection for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 1575–1585. [CrossRef]
19. Sun, K.; Geng, X.; Ji, L. A new sparsity-based band selection method for target detection of hyperspectral image. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 329–333.
20. Jimenez-Rodrguez, L.O.; Arzuaga-Cruz, E.; Velez-Reyes, M. Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 469–483. [CrossRef]
21. Zhong, Z.; Fan, B.; Duan, J.; Wang, L.; Ding, K.; Xiang, S.; Pan, C. Discriminant Tensor Spectral–Spatial Feature Extraction for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *12*, 1028–1032. [CrossRef]
22. Wong, W.K.; Lai, Z.; Wen, J. Low rank embedding for robust image feature extraction. *IEEE Trans. Image Process.* **2017**, *26*, 2905–2917. [CrossRef]
23. 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]
24. Lu, X.; Yuan, Y.; Zheng, X. Joint Dictionary Learning for Multispectral Change Detection. *IEEE Trans. Cybern.* **2017**, *47*, 884–897. [CrossRef]
25. Xu, L.; Wong, A.; Li, F.; Clausi, D.A. Intrinsic Representation of Hyperspectral Imagery for Unsupervised Feature Extraction. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1118–1130. [CrossRef]
26. Zhao, W.; Du, S. Spectral–Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]

27. Guo, X.; Huang, X.; Zhang, L.; Zhang, L.; Plaza, A.; Benediktsson, J.A. Support tensor machines for classification of hyperspectral remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3248–3264. [CrossRef]
28. Yuan, S.; Xia, M.; Chen, L. Multilinear Spatial Discriminant Analysis for Dimensionality Reduction. *IEEE Trans. Image Process.* **2017**, *26*, 2669–2681. [CrossRef]
29. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 844–853. [CrossRef]
30. Li, Z.; Huang, L.; Zhang, D.; Liu, C.; Wang, Y.; Shi, X. A Deep Network Based on Multiscale Spectral-Spatial Fusion for Hyperspectral Classification. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Changchun, China, 17–19 August 2018; pp. 283–290.
31. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
32. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
33. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
34. Yang, J.; Zhao, Y.Q.; Chan, C.W. Learning and Transferring Deep Joint Spectral–Spatial Features for Hyperspectral Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4729–4742. [CrossRef]
35. 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]
36. Zhao, W.; Du, S. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 155–165. [CrossRef]
37. Song, W.; Li, S.; Fang, L.; Lu, T. Hyperspectral Image Classification with Deep Feature Fusion Network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *99*, 3173–3184. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 770–778.
39. Cristianini, N.; Shawetaylor, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. *Kybernetes* **2001**, *32*, 1–28.