symmetry

MDPI

Article

A Novel Classification Framework for Hyperspectral Image Data by Improved Multilayer Perceptron Combined with Residual Network

Aili Wang, Meixin Li and Haibin Wu *

Heilongjiang Province Key Laboratory of Laser Spectroscopy Technology and Application, Harbin University of Science and Technology, Harbin 150080, China; aili925@hrbust.edu.cn (A.W.); 2020600027@stu.hrbust.edu.cn (M.L.)
* Correspondence: woo@hrbust.edu.cn

Abstract: Convolutional neural networks (CNNs) have attracted extensive attention in the field of modern remote sensing image processing and show outstanding performance in hyperspectral image (HSI) classification. Nevertheless, some hyperspectral images have fixed position priors and parameter sharing between different positions, so the common convolution layer may ignore some important fine and useful information and cannot guarantee to effectively capture the optimal image features. This paper proposes an improved multilayer perceptron (IMLP) and IMLP combined with ResNet (IMLP-ResNet) two models for HSI classification. Combined with the characteristics of hyperspectral data, we design IMLP based on three improvements. Specifically, a depthwise over-parameterized convolutional layer is introduced to increase learnable parameters of the model in IMLP, which speeds up the convergence of the model without increasing the computational complexity. Secondly, a Focal Loss function is used to suppress the useless ones in the classification task and enhance the critical spectral–spatial features, which allow the IMLP network to learn more useful hyperspectral image information. Furthermore, to enhance the convergence speed of the network, cosine annealing is introduced to further improve the training performance of IMLP. Furthermore, the IMLP module is combined with a residual network (IMLP-ResNet) to construct a symmetric structure, which extracts more advanced semantic information from hyperspectral images. The proposed IMLP and IMLP-ResNet are tested on the two public HSI datasets (i.e., Indian Pines and Pavia University) and a real hyperspectral dataset (Xuzhou). Experimental results demonstrate the superiority of the proposed IMLP-ResNet method over several state-of-the-art methods with the highest OA, which outperforms CNN by 8.19%, 6.28%, 5.59% and outperforms ResNet by 3.52%, 3.54%, 2.67% on Indian Pines, Pavia University and Xuzhou datasets, respectively, and demonstrates that the well-designed MLPs can also obtain remarkable classification performance of HSI.

Keywords: remote sensing; hyperspectral image classification; convolutional neural network; multilayer perceptron; residual network

Citation: Wang, A.; Li, M.; Wu, H. A Novel Classification Framework for Hyperspectral Image Data by Improved Multilayer Perceptron Combined with Residual Network. *Symmetry* **2022**, *14*, 611. https://doi.org/10.3390/sym14030611

Academic Editor: Dumitru Baleanu

Received: 22 February 2022
Accepted: 16 March 2022
Published: 18 March 2022

1. Introduction

Hyperspectral images (HSI) generally consist of tens to hundreds of continuous spectral bands [1], and can provide rich spatial and spectral information simultaneously, which offers great potential for the subsequent information extraction and practical applications in people's lives [2]. Therefore, HSI is becoming a valuable tool for monitoring the Earth's surface, and is used in a wide range of applications, such as environmental monitoring [3], precision agriculture [4], military investigation [5], and so on.

Hyperspectral image classification (HSIC) is one of the hot issues in hyperspectral research. Taking advantage of rich spectral information, numerous classification methods have been developed. Support vector machine (SVM) [6] has good robustness to high-dimensional hyperspectral data. K-nearest neighbor (KNN) [7] is one of the simplest

classifiers for HSI classification. Random forest (RF) [8] is an ensemble learning method that operates by constructing multiple decision trees in the training process. In addition to these, decision trees [9], extreme learning machines [10], sparse representation-based classifiers [11] and many other methods have been further adopted to improve the performance of hyperspectral image classification. Nevertheless, it is difficult to accurately distinguish different land-cover categories using the spectral information [12]. Zhan et al. [13] used factor analysis to learn effective spectral and spatial features, and applied a Large-margin Distribution Machine (LDM) to hyperspectral remote sensing image classification. Meanwhile, morphological profile-based methods [14] have been proposed to effectively combine spatial and spectral information.

However, the conventional methods are based on handcrafted spectral–spatial features [15], which heavily depend on professional expertise and are quite empirical. Deep learning-based methods can automatically extract spectral features, spatial features, or spectral–spatial features of HSIs for classification application. Chen et al. [16] proposed a stacked autoencoder (SAE) to extract the joint spectral–spatial features for completing accurate HSI classification. Li et al. [17] used a single restricted Boltzmann machine (RBM) and a multilayer DBN to extract spectral–spatial features and obtained superior classification performance compared to the SVM-based method. Makantasis et al. [18] introduced a 2-D CNN to HSI classification, which achieved satisfactory performance with encoded spectral–spatial information with CNN and conducted classification with a multilayer perceptron. Chen et al. [19] used 3-D CNN to simultaneously extract spectral–spatial features and achieved a better result for HSI classification. Nonetheless, due to the loss of information caused by the vanishing gradient problem, training deep CNNs is still a little difficult. Recently, He et al. [20] proposed the residual network (ResNet) to solve this problem well, which defines a residual block as infrastructure elements to facilitate learning of deeper networks and enabling them to be substantially deeper. Zhong et al. [21] designed a spectral–spatial residual network (SSRN), which uses spectral residual blocks and spatial residual blocks consecutively to learn deep discriminative features from abundant spectral features and spatial contexts of HSI and achieved the most advanced HSI classification accuracy on agricultural, urban–rural and urban datasets. Moreover, a deep pyramidal residual network (PyResNet) [22] was developed to learn more robust spectral–spatial representations from the HSI cubes and provided competitive advantages (in terms of both classification accuracy and computational time) over the most advanced HSI classification methods.

Although CNN-based models have achieved good performance for HSI classification, the intrinsic complexity of remote sensing hyperspectral images still limits the performance of many models based on CNN. Firstly, the parameters of CNN increase exponentially with the convolution layer, and the size becomes lager with the increase in computing power. In addition, due to the long-running multiplication and addition time, the consumption of calculation has become the bottleneck of practical application. Finally, the translation invariance and local connectivity of CNN will affect the HSI classification effect. MLP, as a neural network with less constraints, can eliminate the negative effects of translation invariance and local connectivity and has been proven to be a promising machine learning technology. The present MLP-Mixer [23] is known as a pioneering MLP model. Furthermore, Liu et al. [24] proposed gMLP, which is based on MLPs combined with gating, and showed that it can perform as well as transformers in vision applications and key language. H. Touvron et al. [25] proposed ResMLP network structure built entirely upon multi-layer perceptron and attained surprisingly good accuracy/complexity tradeoffs on ImageNet. In addition, RaftMLP [26] aims to achieve cost-effectiveness and ease of application to downstream tasks with fewer resources in developing a global MLP-based model.

MLP solves translation invariance and local connectivity problems; residual networks can prevent model degradation and facilitate rapid convergence due to the retention of original information. Therefore, we proposed two MLP-based classification framework: an

improved MLP (IMLP) model, and IMLP combined with ResNet (IMLP-ResNet) to achieve superior HSI classification performance in this paper.

As a summary, the following are the main contributions of this study.

1. MLP, as a less constrained network, can eliminate the negative effects of translation invariance and local connectivity. Therefore, this paper introduces MLP into HSI classification to fully obtain the spectral–spatial features of each sample and improve the classification performance of HSI.
2. Based on the characteristics of hyperspectral images, we designed IMLP by introducing depthwise over-parameterized convolution, a Focal Loss function and a cosine annealing algorithm. Firstly, in order to improve network performance without increasing reasoning computation, depthwise over-parameterized convolutional layer replaced the ordinary convolution, which can speed up training with more parameters. Secondly, a Focal Loss function is used to enhance the important spectral–spatial features and prevent useless ones in the classification task, which allows the network to learn more useful hyperspectral image information. Finally, a cosine annealing algorithm is introduced to avoid oscillation and accelerate the convergence rate of the proposed model.
3. This paper inserts IMLP between two 3×3 convolutional layers in the ordinary residual block, called as IMLP-ResNet, which has a stronger ability to extract deeper features for HSI. Firstly, the residual structure can retain the original characteristics of the HSI data, and avoid the issues of gradient explosion and gradient disappearance during the training process. In addition, the residual structure can improve the modeling ability of the model. Moreover, IMLP can improve the feature extraction ability of residual network, so that the model strengthens the key features on the basis of retaining the original features of hyperspectral data.

The rest of this article is organized as follows. Section 2 describes our proposed classification approach. Section 3 reports the experimental results and appraises the performance of the proposed method part. Section 4 gives the discussion and analyzes how to choose experimental parameters in the IMLP-ResNet classification model. Section 5 gives the final conclusions and discusses research directions in the future.

2. The Proposed MLP-Based Methods for HSI Classification

Considering that the deepening of network layers in deep learning will cause the phenomenon of gradient disappearance and gradient explosion, the classification model adopts residual network as the basic framework. Figure 1 shows the overview flowchart of the improved MLP combined with ResNet (IMLP-ResNet) for HSI classification.

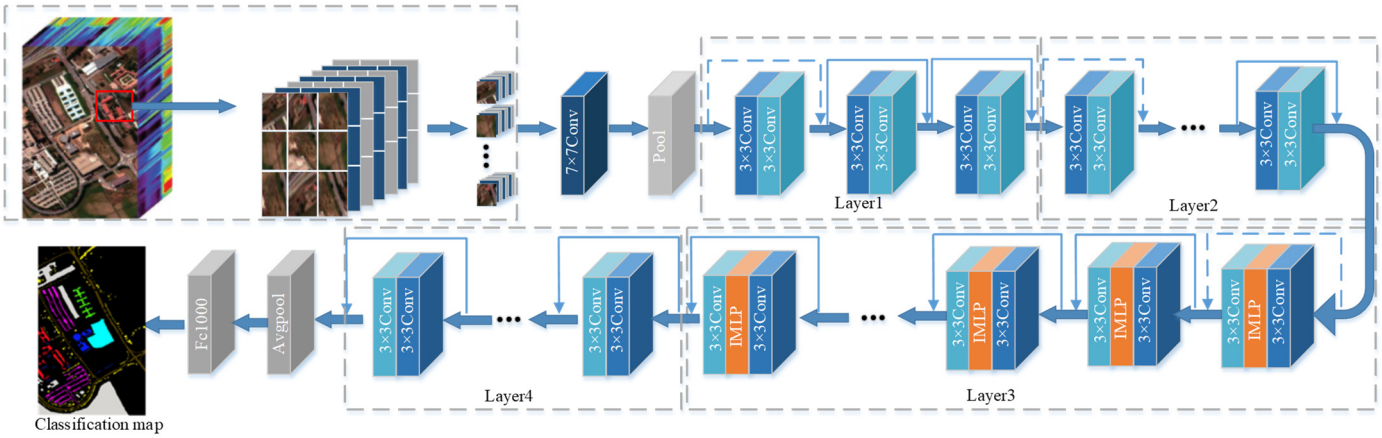


Figure 1. The framework of IMLP-ResNet for HSI classification.

First of all, the improved MLP (IMLP) model for HSI classification is described in detail.

2.1. The Proposed Improved MLP (IMLP) for HSI Classification

Figure 2 gives the overall architecture of the proposed IMLP for HSI classification, which consists of two stages: a training stage and a testing stage. In the training stage, the network consists of a Global Perceptron module, Partition Perceptron module and Local Perceptron module. The structural reparameterization means that the training-time model has a set of parameters and the inference-time model has another set [27], and parameterizes the latter with the former's parameters. The detailed description is explained as follows. It is assumed that the HSI dataset is the size of $H \times W \times nBand$, where H and W represent spatial height and width, and $nBand$ is the frequency band number. First, each pixel of the hyperspectral image is processed with a fixed window size $y \times x$, and a single sample with a shape of $y \times x \times nBand$ is generated. Subsequently, with the shape of each patch, it becomes $R \times R \times nBand$. In this paper, the patch size is set to 4×4.

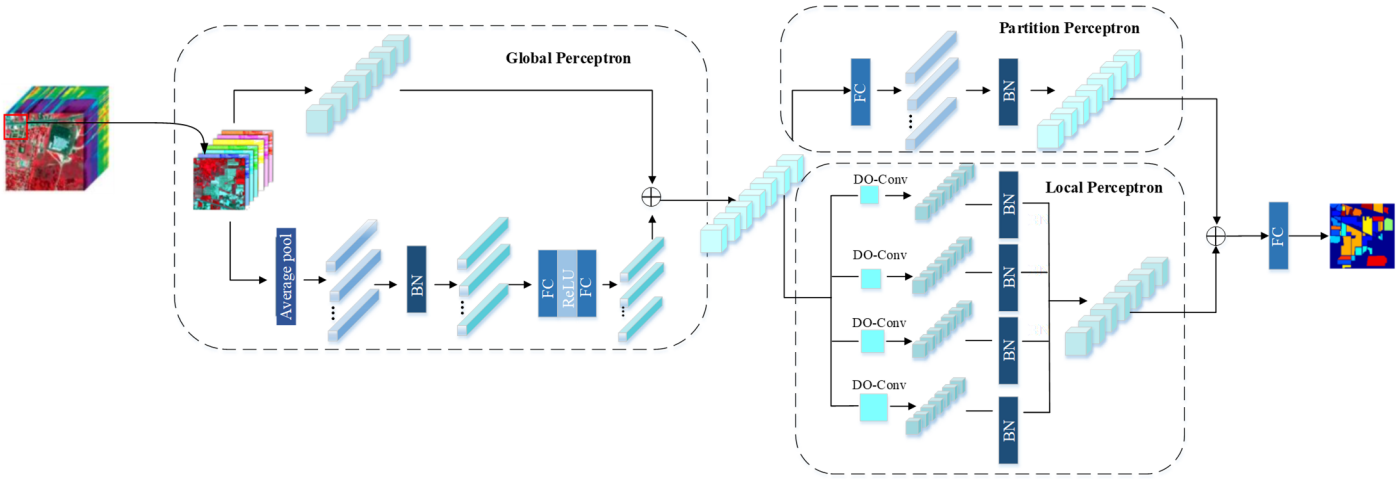


Figure 2. The structure of IMLP for HSI classification.

The global perceptron module block consists of two branches. The first branch of the global perceptron module splits up the input hyperspectral feature image. The hyperspectral feature map changes from (H, W, C) to (h_1, w_1, O) In the second branch, the original feature map (H, W, C) is evenly pooled, and the size of the hyperspectral feature map becomes (h, w, O). H, W, C indicate the height, width and number of input channels of the input hyperspectral feature map, respectively. h_1, w_1, O respectively represent the height, width and number of output channels of the split hyperspectral feature image. Finally, h, w indicate the height and width of the hyperspectral feature image after average pooling as follows:

$$h_1 = \frac{H}{h}, \; w_1 = \frac{W}{w} \tag{1}$$

The second branch uses average pooling to achieve a pixel for each hyperspectral feature image, and then feeds it though BN and a two-layer MLP. The hyperspectral feature map (h, w, O) is sent to the BN layer and two fully connected layers. The ReLU function is introduced between the two fully connected layers to effectively avoid gradient explosion and gradient disappearance. For the fully connected layer, $X^{(in)}$ *and* $X^{(out)}$ represent input and output, and the kernel $W \in R^{Q \times P}$ is the matrix multiplication (MMUL) defined as follows:

$$X^{(out)} = MMUL\left(X^{(in)}, W\right) = X^{(in)} \cdot W^T \tag{2}$$

The hyperspectral vector is transformed into $(1, 1, C)$ by BN layer and two fully connected layers, after which the hyperspectral feature images are obtained after all branches

are added. Then, the hyperspectral features are input to partition perceptron and local perceptron without dividing.

The Partition Perceptron module block contains a BN layer and a group convolution. The input of the partition perceptron is (h, w, O). Then access the group convolution of groups = 4 and BN layer. After BN layer and a group convolution processing, it becomes the original hyperspectral feature input (H, W, C). $Y^{(out)} \in R^{C \times H \times W}$ indicates the output hyperspectral feature. p is the number of pixels filled, while $F \in R^{C/g \times K \times K}$ is the convolution kernel. g indicates the number of convolution groups. The calculation formula of $Y^{(out)}$ is shown in Equation (3).

$$Y^{(out)} = g\left(Y^{(in)}, F, g, p\right),\ F \in R^{C/g \times K \times K} \tag{3}$$

The Local Perceptron module contains a depthwise over-parameterized convolutional layer (DO-Conv) [28] and a BN layer. First, the local perceptron module sends the segmented hyperspectral feature image (h, w, O) simultaneously to the deep hyperparametric convolution layer. Then the feature graph is fed into BN layer, and the output of all convolution branches is added with the output of the partition perceptron as the final output. In the test phase, reparameterization is carried out to fuse the two parts of the local perceptron module and the partitioned perceptron module into a fully connected layer. The FC kernel of a DO-Conv kernel is the result of convolution on an identity matrix with proper reshaping operation. Formula (4) shows exactly how to build $W^{(F,p)}$ from *F and p*.

$$W^{(F,p)} = DOCONV(Y, F, p), (Chw, Ohw)^T \tag{4}$$

In order to increase the learnable parameters of the proposed model, a deeply over-parameterized convolutional layer is introduced to replace the ordinary convolutional layer to construct IMLP. In addition, IMLP introduced Focal Loss for the purpose of solving the problem of data imbalance in hyperspectral image classification and the cosine annealing algorithm to improve the training performance of IMLP, which makes the convergence speed of the network faster. The three modifications are described in the following parts.

2.1.1. DO-Conv

In order to improve the training speed of the model, DO-Conv is introduced to replace the traditional convolution layer in the local perceptron module. The architecture of DO-Conv is shown as Figure 3. There are two components in DO-Conv, including a feature component and a convolution kernel component. The model is more efficient after adding the convolution kernel component, so this paper uses the convolution kernel component to train the network. The DO-Conv is composed of a conventional convolution $W \in R^{C_{out} \times D_{mul} \times C_{in}}$ and a deep convolution. $D \in R^{(M \times N) \times D_{mul} \times C_{in}}$. In conventional convolution, the convolution layer slides the input data, and each element of the output feature is obtained by the horizontal slice of the convolution kernel and P dot product of the image block. In the deep convolution layer, the convolution kernel is convolved with each input channel during the training phase.

At the end of the training phase, the multi-layer composite linear operation used for over-parameterization is folded into a compact single-layer representation. Then, only one layer is used for reasoning, reducing the calculation to full equivalence with the regular layer. M and N are spatial dimensions of $\mathbb{P}$, C_{in} is the number of input feature graphs, C_{out} is the number of D_{mul} output feature graphs, $D^T \in R^{D_{mul} \times (M \times N) \times C_{in}}$ is the transposition of $D \in R^{D_{mul} \times (M \times N) \times C_{in}}$ and the convolution kernel of DO-Conv is W'. First, the deep convolution kernel D^T and the convolution kernel of ordinary convolution W are combined into W', $W' = D^T \circ W$. The convolution output feature O is then generated as $O = W' * \mathbb{P}$, where $*$ is convolution, $\circ$ is the dot product, and # is the defined operator.

$$O = (D, W)\#P = (D\hat{}T \circ W) * P \tag{5}$$

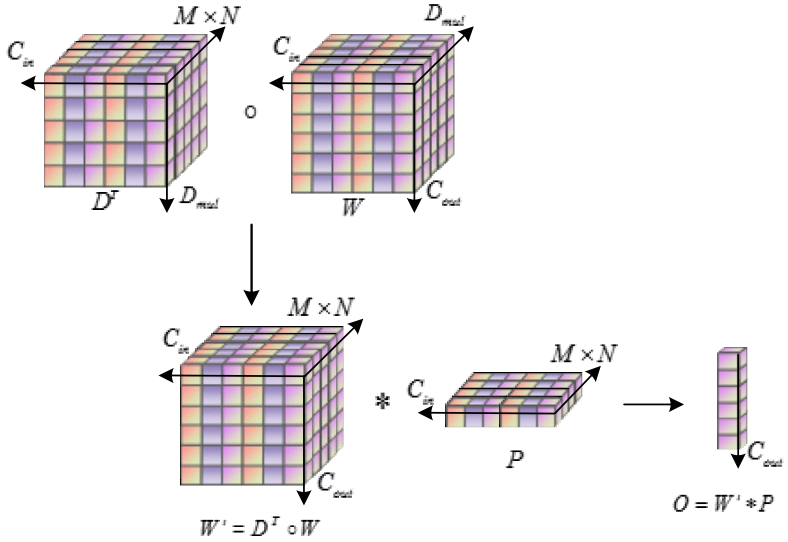


Figure 3. The architecture of DO-Conv.

2.1.2. Focal Loss

Data imbalance is common in hyperspectral remote sensing images. Because there are various objects with different sizes in a hyperspectral scene, it is very difficult to mark samples in practice. Therefore, there is usually a serious imbalance between various samples of hyperspectral data [29]. Thus, this paper introduced focal loss function instead of cross entropy loss (CE) function. CE is written as follows:

$$CE(p, y) = \begin{cases} -\log(p) \; if \; y = 1 \\ -\log(1-p) \; otherwise \end{cases} \tag{6}$$

where $y \in \{\pm 1\}$ specifies the ground-truth class and $P \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$, and p_t is defined as follows:

$$p_t = \begin{cases} p \; if y = 1 \\ 1 - p \; otherwise \end{cases} \tag{7}$$

Focal Loss is calculated as follows:

$$FL(p_t) = -(1 - p_t)^{\gamma} log(p_t) \tag{8}$$

Focusing parameter γ can adjust the weight of positive and negative samples as well as control the weight of difficult and easy samples. When some samples are misclassified and p_t is very small, the regulatory factor $(1 - p_t)^{\gamma}$ is close to 1, which has little influence on loss function. However, as p_t tends to 1, this factor will gradually tend to 0, and losses for well-classified samples will also decrease, so as to achieve the effect of reducing weight. γ will smoothly adjust the proportion of reduced weights for easily classified samples. Increased γ can enhance the influence of the regulatory factor, which reduces the loss contribution of easily classified samples and broadens the range of low loss received by samples.

2.1.3. Cosine Annealing Algorithm

The batch gradient descent (BGD) and stochastic gradient descent (SGD) are mainly used to update parameter values in deep learning. BGD needs to update each parameter with all the data sets. If the sample size is too large, the training speed will be too slow, which will increase the computational cost. However, SGD has a characteristic fast training speed, because it uses part of the information of the data and easily falls into a local optimal solution [30]. Therefore, this article introduces the cosine annealing algorithm to update

the parameter values under the premise of comprehensive training sample speed and computational cost, and the learning rate can be reduced by the cosine function. We decay the learning rate with a cosine annealing for each batch as follows:

$$\eta_t = \eta_{min}^i + 1/2\left(\eta_{max}^i - \eta_{min}^i\right)\left(1 + cos\left(\frac{T_c}{T_i}\pi\right)\right) \quad (9)$$

where η_{min}^i and η_{max}^i are ranges for the learning rate, T_i is the total number of epochs, and T_c is the current epoch. When $T_c = T_i$, η_t reaches the minimum training batch.

When the gradient descent algorithm is used to optimize the objective function, the learning rate should become smaller to get closer to the global minimum value of the loss function and make the model as close as possible to this point. The cosine annealing algorithm can reduce the learning rate by cosine function. The cosine goes down slowly as x increases, then it accelerates and goes down slowly again.

2.2. The Proposed IMLP-ResNet Model for HSI Classification

The main idea of an IMLP-ResNet model refers to the insertion of IMLP between two 3×3 convolutional layers in the ordinary residual block; that is to say, the IMLP module inserted into the third layer of ResNet has a stronger ability to extract deeper features for HSI. First of all, ResNet34 can retain the original characteristics of the HSI data. It can solve gradient explosion and gradient disappearance in the training process. In the meantime, ResNet34 can improve the modeling ability of the model. IMLP can improve the feature extraction ability of residual network and strengthen the key features on the basis of retaining the original features. ResNet34 compared with other CNN models can help overcome the over-fitting phenomenon. The ResNet family includes ResNet18, ResNet34, ResNet50, ResNet152, etc. In order to improve the classification efficiency, ResNet34 with fewer parameters was used in this paper.

2.2.1. The Structure of ResNet34

The classification performance of the deep learning model decreases with the increase in depth [31]. Inspired by deep residual learning framework, this aggravating problem can be solved by adding quick connections and propagating eigenvalues between each layer.

The core of the deep residual network lies in the residual learning module, which can save part of the original input information during the training of the deep CNN model [32,33]. In this way, the learning target is transferred to avoid the saturation of classification accuracy caused by the depth of the network. As shown in Figure 4, x represents the input, $H(x)$ represents the output, and $F(x)$ represents the residual function. The output of the residual unit is shown in Equation (10).

$$H(x) = F(x) + x \quad (10)$$

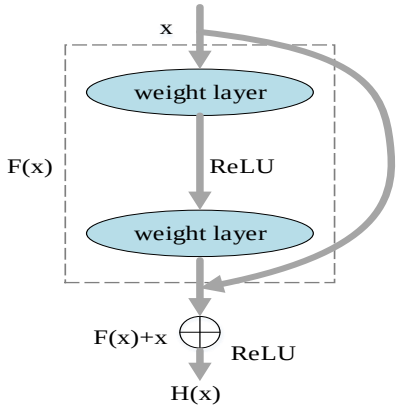


Figure 4. The architecture of the residual block.

The residual module calculates the residual error when the span is not interrupted. $\{W_i\}$ is used to show the residual module block, the residual module actually calculates the output result as shown in Equation (11).

$$y = F(x, \{W_i\}) + x \tag{11}$$

$F(x, \{W_i\})$ is residual mapping and can be obtained by back propagation (BP). For the case of two weight layers, the calculation process is shown in Equation (12) when the bias is ignored.

$$F(x, \{W_i\}) = W_2\sigma(W_1)x = W_2ReLU(W_1)x \tag{12}$$

The calculation of residual module requires that $F(x, \{W_i\})$ and x have the same dimension. A linear projection W_s is proposed by the shortcut connections to match the dimensions:

$$y = F(x, \{W_i\}) + W_s x \tag{13}$$

Figure 5 is the overall architecture of ResNet34, which adds shortcut connections between each two layers, and can directly sample the input image with the convolution of stride of 2. It can be seen that there are four layers in the structure of ResNet34 and each layer has 3, 4, 6, and 3 residual blocks, respectively. The convolutional layers mostly have 3×3 filters for the same output feature map size. In order to maintain the time complexity of each layer, the number of filters was doubled if the feature map size is halved. The size of the feature map is halved and the number of feature maps is doubled to maintain the complexity of the network.

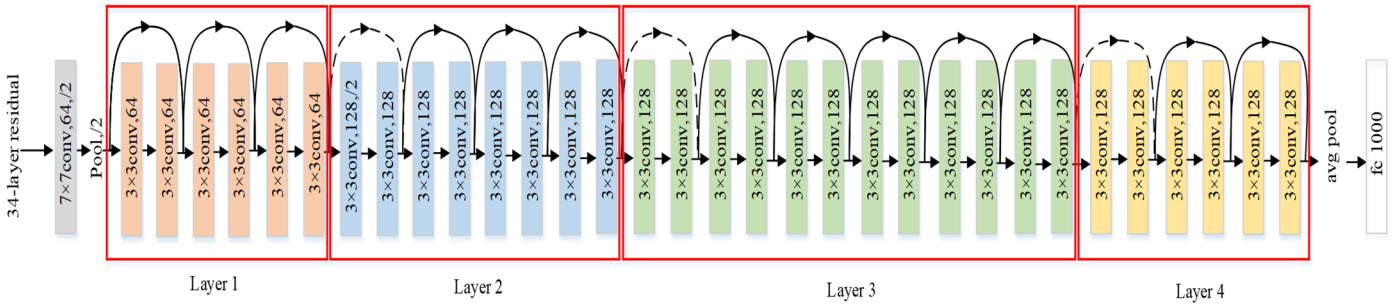


Figure 5. The overall architecture of ResNet34.

2.2.2. IMLP-ResNet Model

Figure 6a shows the structure of the ordinary residual block, which contains two 3×3 convolutional layers and a shortcut connection. BN is applied after the convolutional layer and before the activation function to accelerate the convergence of the module. The shortcut connection enables the gradient to propagate directly from the later to earlier layers, thus mitigating the gradient vanishing. The stacking multiple residual blocks can develop a deeper network to alleviate overfitting of the network.

As shown in Figure 6b, this paper inserts IMLP between two 3×3 convolutional layers in the ordinary residual block to constitute a symmetric structure. Traditional convolutional layers obtain long-range dependencies by the large receptive fields formed by deep stacks of convolutional layers. However, repetition of local operations requires a lot of computation and may cause optimization difficulties. At the same time, some images have intrinsic positional prior, which cannot be fully utilized by a convolutional layer because it shares parameters among different positions. IMLP runs faster than CNN with the same number of parameters and has global capacity and positional perception. Therefore, our proposed IMLP-ResNet can perform fine-feature extraction at different network levels and learn more comprehensive feature representations for HSI classification.

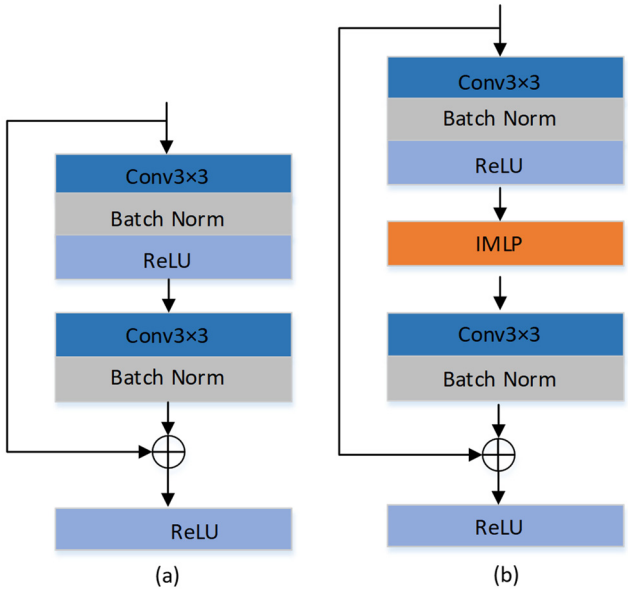


Figure 6. Architectures of ResNet and IMLP-ResNet block. (**a**) residual block; (**b**) IMLP-ResNet.

3. Results

3.1. Dataset Description

In order to verify the classification performance efficiently, a number of experiments were performed on two standard hyperspectral datasets (Indian Pines and Pavia University), and the Xuzhou dataset. The Indian Pines dataset was acquired in 1992 by an Airborne Visible-Infrared Imaging Spectrometer (AVIRIS) sensor at the Indian Pines Test Site in northwestern Indiana with a size of 145 × 145 pixels, 224 spectral bands and 16 types of land cover. The number of bands was reduced to 200 by removing the bands covering the water-absorbing area (bands 104–108, 150–163, 220). The Pavia University dataset was picked up by ROSIS sensors flying over Pavia in northern Italy. The number of spectral bands is 103, and the size is 610 × 610 pixels with nine categories. Figures 7 and 8 show the false-color composite image and ground truth map, and Tables 1 and 2 report the detailed number of pixels available in each class for the two datasets respectively.

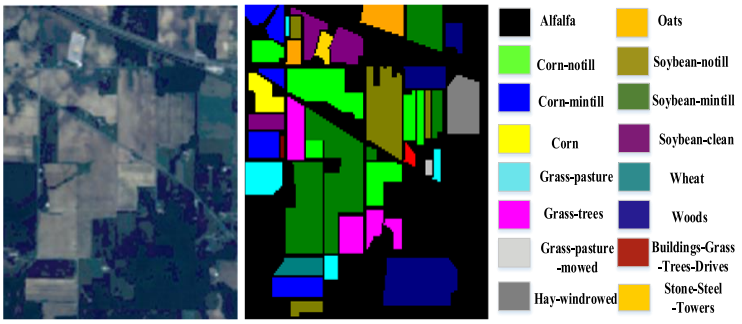


Figure 7. Indian Pines dataset. (**a**) False color map; (**b**) ground truth map.

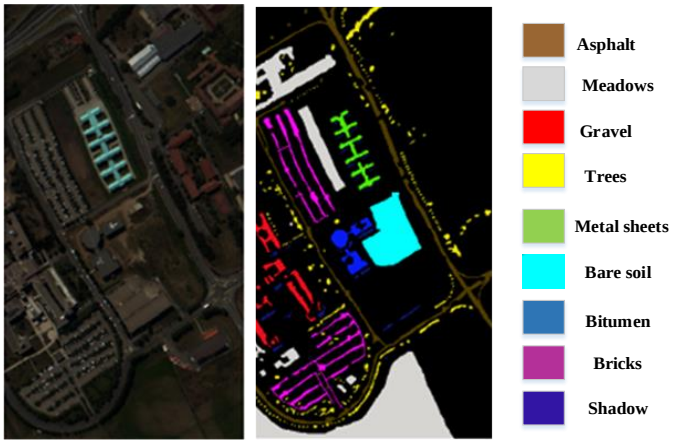


Figure 8. Pavia dataset. (**a**) False color map; (**b**) ground truth map.

Table 1. Indian pines labeled sample counts.

Class Code	Name	Sample Numbers
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	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. Pavia University labeled sample counts.

Class Code	Name	Sample Numbers
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

The Xuzhou dataset was obtained via a HySpex SWIR-384 and HySpex VNIR-1600 imaging spectroradiometer in Xuzhou in November 2014, with a size of 500 × 260 pixels and 436 bands. Based on the field survey, nine feature types were identified. Figure 9 shows the false-color composite image and the ground truth graph. Table 3 reports the detailed number of pixels available in each class.

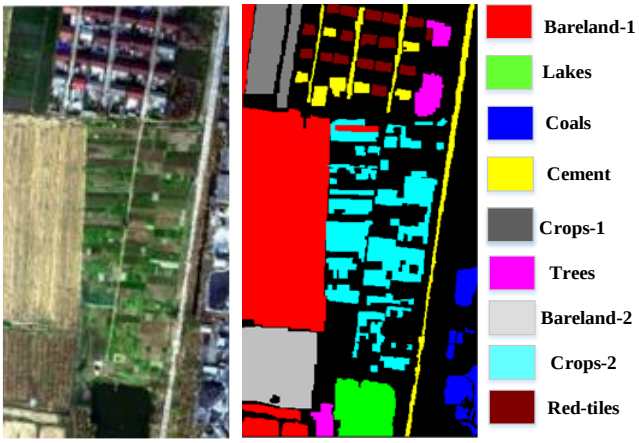


Figure 9. Xuzhou dataset. (**a**) False color map; (**b**) ground truth map.

Table 3. Xuzhou labeled sample counts.

Class Code	Name	Sample Numbers
1	Bareland-1	26,396
2	Lakes	4027
3	Coals	2783
4	Cement	5214
5	Crops-1	13,184
6	Trees	2436
7	Bareland-2	6990
8	Crops-2	4777
9	Red-tiles	3070
Total		68,877

3.2. Experimental Parameters Setting

All experiments were performed on an Intel(R) Xeon(R) 4208 CPU @ 2.10 GHz processor and Nvidia GeForce RTX 2080Ti graphics card. In order to reduce experimental errors, the model randomly selected a limited number of samples from the training set for training. The epoch was set to 200, and the batch size was set to 32. All experimental results were averaged from 10 experiments. Overall accuracy (OA), average accuracy (AA) and Kappa coefficient (K) were used as evaluation indexes to measure the performance of each method. This model uses Adam optimizer to learn the weight of three-dimensional spectral space filter, and adopts cosine annealing to adjust the learning rate, taking cosine function as the period, and resetting the learning rate at the maximum value of each period. The initial learning rate of this method was 0.001, with a cycle of 15 epochs. After 15 epochs, the learning rate was automatically increased and the local optimum was skipped.

3.3. Evaluation Metrics

The evaluation index is the standard to evaluate the quality of the algorithm model, which guides us to better improve the algorithm's classification performance. In this experiment, the Confusion Matrix is used to count the classification results, and the Overall Accuracy (OA), Average Accuracy (AA) and Kappa coefficient (K) are used to evaluate the classification results.

Confusion Matrix is a kind of evaluation matrix commonly used in classification problems. Each row of the matrix represents the number vector of a category divided into all classes, and each column represents the number vector of all categories divided into all classes. As shown in Formula (14), the diagonal elements of the matrix are the number of correctly classified categories of a certain category, where C is the number of categories

of classification problems, and m_{ij} represents the ith class samples misclassified into the jth class.

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1j} & \dots & m_{1C} \\ m_{21} & m_{22} & \dots & m_{2j} & \dots & m_{2C} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ m_{i1} & m_{i2} & \dots & m_{ij} & \dots & m_{iC} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ m_{C1} & m_{C2} & \dots & m_{Cj} & \dots & m_{CC} \end{bmatrix} \quad (14)$$

OA computes to the ratio between the number of correctly classified samples and that of the total samples to be tested. This index is a common evaluation standard for classification problems and reflects the probability of consistency between classification results and real reference values, as written in Formula (15).

$$OA = \frac{trace(M)}{N} \quad (15)$$

where $trace(M)$ is the trace of the matrix, that is, the sum of all elements on the main diagonal of matrix M, and N is the total number of all test samples.

AA represents the average classification accuracy of each category, which reflects the average performance of all categories. m_{i+} represents the sum of all elements in row i, and C represents the total number of categories.

$$AA = \Sigma_{i=1}^{N} \frac{m_{ii}}{m_{i+}} / C \quad (16)$$

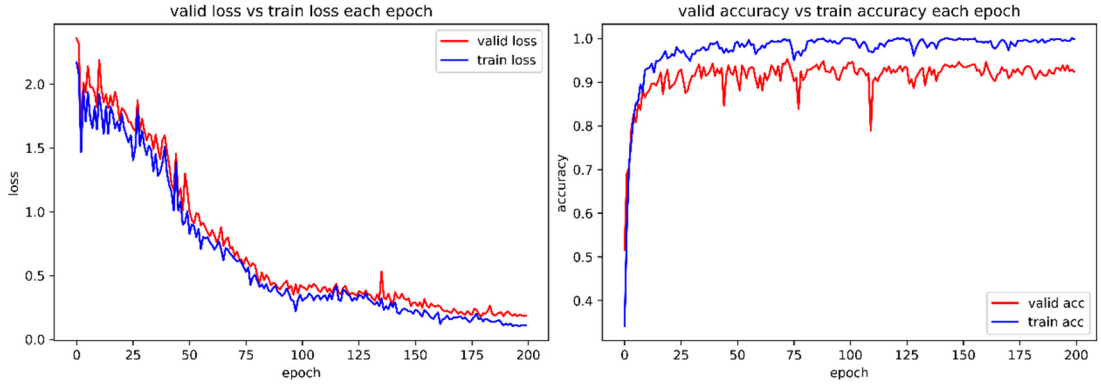
K is an index to measure the classification accuracy, which can evaluate the classification performance more comprehensively by integrating the overall classification accuracy and average classification accuracy.

$$K = \frac{N\Sigma_{i=1}^{C} m_{ii} - \Sigma_{i=1}^{C} m_{i+} m_{+i}}{N^2 - \Sigma_{i=1}^{C} m_{i+} m_{+i}} \quad (17)$$

where m_{i+} represents the ith row of the confusion matrix, and m_{+i} represents the ith column of the Confusion Matrix.

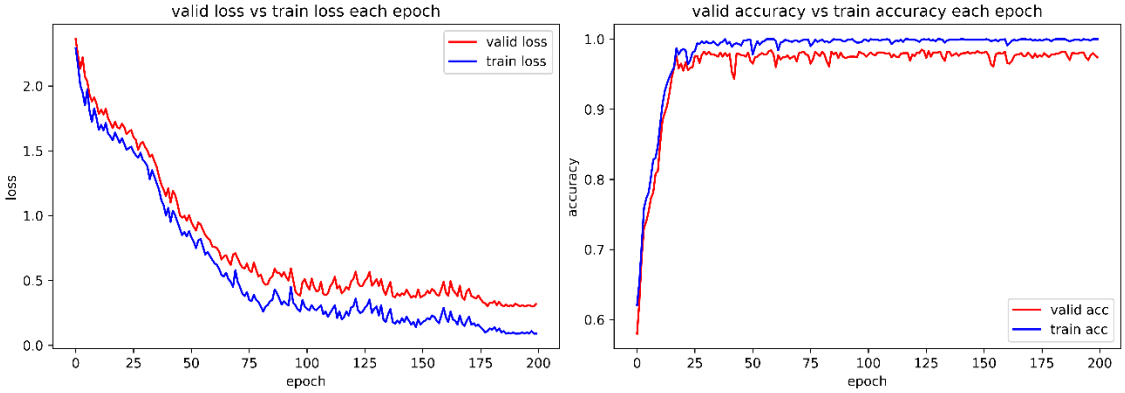
3.4. Comparison of the Proposed Methods with the State-of-the-Art Methods

The curves of the loss and accuracy of the training and testing of all datasets classified by the proposed IMLP-ResNet over 200 epochs are shown in Figures 10–12. It can be observed from Figures 10a, 11a and 12a that, with the increase in the number of epochs in the Indian Pines, Pavia University and Xuzhou datasets, the losses in training sets and validation sets decreased continuously. In Figures 10b, 11b and 12b, classification accuracy keeps improving. The Indian Pines dataset and Pavia University dataset converged around epoch 180, while the Xuzhou dataset converged around the epoch of 190. Among them, the Xuzhou data set converges slowly compared with the other two data sets, because the number of training samples in this dataset is higher than the other two datasets. However, the accuracy of the training set and validation set of the three datasets is still improved after the model converges. The main reason is that, with the continuous optimization of parameters, the gradual fitting of curves verifies the good generalization ability of our proposed model and the convergence of this model.



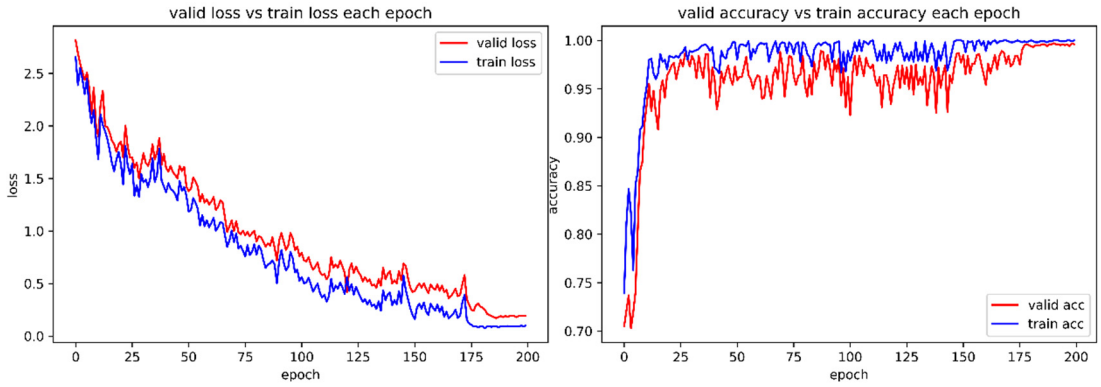
(**a**) (**b**)

Figure 10. Comparison of loss and accuracy in the search process on the Indian Pines dataset. (**a**) Loss; (**b**) Accuracy.



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

Figure 11. Comparison of loss and accuracy in the search process on the Pavia dataset. (**a**) Loss; (**b**) Accuracy.



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

Figure 12. Comparison of loss and accuracy in the search process on the Xuzhou dataset. (**a**) Loss; (**b**) Accuracy.

The experiment mainly compared the proposed algorithm with the Radial Basis Function (RBF) Support Vector Machine algorithm (RBF-SVM) [34] and Extended Morphological Profile (EMP) Support Vector Machine Calculation Methods (EMP-SVM) [35], Deep Convolutional Neural Network (DCNN) [36], Spectral–Spatial Residual Network (SSRN) [21],

Residual Network (ResNet) [37], Pyramid Residual network (PyResNet) [22], RepMLP [38], IMLP classification performance for hyperspectral dataset. Ten percent of the total sample number was used as the training sample number for hyperspectral classification as shown in Tables 4–6. Compared with other methods, the IMLP-ResNet proposed in this paper has the highest classification accuracy for the three datasets. For example, compared with RBF-SVM in the Indian Pines dataset, IMLP-ResNet increased OA, AA and K by 12.85%, 12.87% and 10.55% and improved by 0.54%, 0.70% and 0.58% compared with RepMLP, respectively. Taking the Xuzhou dataset as an example, OA reached 98.15%, compared with RBF-SVM, EMP-SVM, DCNN, SSRN, ResNet, PyResNet, RepMLP and IMLP, in which OA increased by 15.89%, 10.72%, 5.59%, 3.98%, 2.67%, 1.80%, 1.37% and 1.01% respectively. The Indian Pines dataset and Pavia University dataset have similar classification results. All the experimental results show that the proposed IMLP-ResNet is superior to other methods.

Table 4. Classification results on the Indian Pines dataset by different classification methods.

Class Code	RBF-SVM	EMP-SVM	DCNN	SSRN	ResNet	PyResNet	RepMLP	IMLP	IMLP-ResNet
1	78.25 ± 2.24	79.62 ± 3.61	78.34 ± 1.89	80.91 ± 1.96	82.67 ± 1.51	88.89 ± 1.57	87.05 ± 2.06	89.25 ± 0.87	91.33 ± 1.28
2	79.22 ± 3.02	84.26 ± 3.49	88.15 ± 2.36	90.04 ± 1.49	91.26 ± 6.29	92.12 ± 1.48	93.38 ± 0.74	94.02 ± 1.23	96.97 ± 2.29
3	80.27 ± 0.98	79.15 ± 2.81	82.37 ± 3.63	84.65 ± 1.91	83.95 ± 1.29	88.39 ± 2.36	90.65 ± 0.28	90.37 ± 3.68	92.16 ± 1.61
4	82.02 ± 1.78	85.34 ± 2.69	89.06 ± 1.67	88.65 ± 5.66	90.38 ± 3.99	91.93 ± 8.23	90.54 ± 1.08	90.37 ± 1.06	92.24 ± 2.98
5	88.22 ± 1.56	87.37 ± 1.63	90.38 ± 1.06	91.98 ± 3.26	93.93 ± 4.06	92.21 ± 1.89	92.48 ± 1.73	93.07 ± 1.36	95.09 ± 2.11
6	82.52 ± 3.02	86.39 ± 2.57	91.38 ± 4.39	92.32 ± 2.32	93.72 ± 3.15	94.26 ± 1.07	94.22 ± 0.27	93.18 ± 3.03	96.51 ± 0.88
7	82.22 ± 2.27	84.38 ± 0.31	85.31 ± 0.98	86.70 ± 3.82	84.31 ± 13.96	90.64 ± 8.96	89.46 ± 3.79	90.37 ± 0.46	95.17 ± 4.44
8	85.02 ± 1.02	86.20 ± 1.58	90.27 ± 3.18	93.08 ± 6.67	92.08 ± 4.37	93.10 ± 2.70	92.22 ± 0.17	93.56 ± 2.30	94.44 ± 1.85
9	83.20 ± 0.52	81.27 ± 2.94	85.09 ± 0.67	86.73 ± 5.95	81.90 ± 1.65	87.84 ± 11.12	89.34 ± 1.29	88.06 ± 3.75	93.60 ± 3.22
10	79.22 ± 1.02	84.66 ± 3.10	88.09 ± 2.16	90.33 ± 5.14	90.98 ± 7.37	91.12 ± 2.50	91.05 ± 2.44	92.34 ± 2.88	94.46 ± 2.56
11	82.27 ± 2.98	86.27 ± 1.06	89.37 ± 1.06	90.36 ± 0.96	91.72 ± 0.68	93.71 ± 2.82	92.54 ± 3.08	93.09 ± 2.85	95.68 ± 2.60
12	85.02 ± 2.27	88.34 ± 0.43	90.76 ± 0.41	92.17 ± 0.61	95.01 ± 0.61	90.70 ± 7.52	91.09 ± 2.06	91.45 ± 1.14	96.02 ± 3.03
13	82.22 ± 0.53	85.61 ± 0.39	89.05 ± 3.28	95.39 ± 1.22	94.91 ± 2.78	95.89 ± 2.93	92.16 ± 3.08	93.07 ± 0.39	94.88 ± 1.07
14	80.52 ± 2.02	85.17 ± 2.09	90.36 ± 1.02	92.03 ± 2.36	91.55 ± 1.89	95.95 ± 1.70	93.81 ± 0.46	94.03 ± 2.69	95.36 ± 2.09
15	81.22 ± 2.27	86.20 ± 1.43	91.06 ± 2.47	93.84 ± 1.45	92.75 ± 3.26	94.65 ± 2.19	94.89 ± 2.04	95.30 ± 0.88	96.33 ± 2.76
16	85.63 ± 1.20	88.69 ± 3.07	90.67 ± 4.09	92.87 ± 2.93	93.65 ± 2.79	95.05 ± 3.12	94.73 ± 3.17	95.37 ± 0.63	96.03 ± 1.58
OA(%)	81.55 ± 1.43	83.64 ± 0.47	86.21 ± 1.43	88.66 ± 0.60	90.88 ± 1.90	92.21 ± 0.98	93.05 ± 3.27	93.59 ± 0.69	94.40 ± 1.62
AA(%)	79.37 ± 0.58	81.76 ± 2.14	83.65 ± 0.48	85.83 ± 3.37	87.76 ± 2.81	90.27 ± 4.12	90.96 ± 0.25	91.66 ± 2.23	92.24 ± 1.73
100 K	82.33 ± 1.86	84.59 ± 0.35	86.93 ± 1.28	88.34 ± 0.69	89.61 ± 1.89	90.78 ± 1.08	91.34 ± 4.87	91.92 ± 0.27	92.88 ± 1.83

Table 5. Classification results on the Pavia dataset by different classification methods.

Class Code	RBF-SVM	EMP-SVM	DCNN	SSRN	ResNet	PyResNet	RepMLP	IMLP	IMLP-ResNet
1	76.56 ± 1.28	86.24 ± 0.43	90.07 ± 1.95	92.29 ± 1.82	92.11 ± 3.35	93.05 ± 1.37	93.08 ± 3.05	93.25 ± 0.22	94.58 ± 4.76
2	81.23 ± 3.54	87.36 ± 1.94	92.48 ± 0.67	93.27 ± 1.79	95.03 ± 2.76	95.88 ± 4.62	94.66 ± 1.31	96.17 ± 2.47	97.55 ± 0.29
3	80.34 ± 0.89	85.57 ± 3.29	90.36 ± 1.65	91.51 ± 2.93	92.58 ± 2.96	92.97 ± 3.13	93.15 ± 2.67	93.20 ± 1.58	94.80 ± 3.75
4	82.01 ± 2.68	85.15 ± 2.36	91.43 ± 3.21	92.22 ± 1.59	94.73 ± 1.25	95.45 ± 1.07	95.89 ± 2.16	96.22 ± 0.34	97.64 ± 0.45
5	80.15 ± 1.34	86.20 ± 2.48	91.86 ± 2.37	93.08 ± 3.07	95.37 ± 2.15	96.87 ± 1.25	96.90 ± 4.79	97.06 ± 3.28	98.57 ± 0.76
6	79.60 ± 2.36	85.71 ± 1.99	92.10 ± 3.08	93.46 ± 2.54	94.78 ± 4.61	95.33 ± 2.46	95.24 ± 1.02	96.37 ± 0.61	98.83 ± 0.40
7	75.36 ± 2.88	84.01 ± 3.49	90.22 ± 0.44	91.03 ± 0.75	93.76 ± 1.91	94.59 ± 2.66	93.57 ± 3.09	94.38 ± 1.57	96.51 ± 2.07
8	73.47 ± 4.16	82.28 ± 1.75	86.25 ± 3.19	88.03 ± 0.43	90.27 ± 0.39	91.36 ± 3.36	91.16 ± 2.14	92.59 ± 2.60	93.26 ± 1.59
9	84.02 ± 4.39	85.13 ± 2.16	90.24 ± 0.82	93.76 ± 1.60	95.33 ± 0.54	96.55 ± 1.82	96.98 ± 1.56	97.03 ± 1.44	98.25 ± 1.85
OA(%)	83.12 ± 2.72	86.01 ± 1.03	91.78 ± 2.52	93.03 ± 1.36	94.52 ± 2.93	95.68 ± 0.18	96.31 ± 3.28	96.89 ± 0.77	98.06 ± 0.64
AA(%)	80.31 ± 3.64	85.24 ± 1.37	90.36 ± 1.04	91.28 ± 2.61	93.49 ± 1.95	94.05 ± 0.32	94.36 ± 0.23	94.87 ± 1.23	95.59 ± 0.69
100 K	78.54 ± 0.19	83.54 ± 2.68	89.02 ± 0.86	90.87 ± 0.18	92.01 ± 2.95	93.87 ± 3.08	94.52 ± 4.17	95.03 ± 1.09	96.88 ± 1.87

Table 6. Classification results on the Xuzhou dataset by different classification methods.

Class Code	RBF-SVM	EMP-SVM	DCNN	SSRN	ResNet	PyResNet	RepMLP	IMLP	IMLP-ResNet
1	81.34 ± 0.25	86.25 ± 3.41	91.25 ± 3.08	93.24 ± 0.37	94.09 ± 1.67	94.14 ± 3.94	95.18 ± 4.96	95.54 ± 0.16	96.98 ± 4.57
2	81.23 ± 2.13	87.16 ± 4.39	91.28 ± 2.14	94.12 ± 4.06	95.02 ± 0.68	96.19 ± 2.17	96.25 ± 0.83	97.73 ± 3.64	98.86 ± 1.56
3	79.28 ± 3.46	86.52 ± 0.63	90.27 ± 0.93	93.36 ± 2.45	94.68 ± 2.17	94.36 ± 2.35	95.94 ± 4.36	96.19 ± 4.72	98.63 ± 0.25
4	80.49 ± 4.10	85.07 ± 1.69	88.21 ± 1.07	90.47 ± 3.88	91.26 ± 3.24	92.10 ± 2.91	92.76 ± 3.41	93.21 ± 1.55	95.16 ± 0.73
5	82.74 ± 0.43	86.06 ± 3.81	90.38 ± 2.46	93.67 ± 2.53	94.06 ± 0.46	95.33 ± 0.97	95.84 ± 3.25	96.58 ± 1.61	98.71 ± 0.52
6	81.09 ± 1.51	84.68 ± 1.42	89.07 ± 3.86	91.03 ± 3.67	93.47 ± 1.23	94.67 ± 4.26	95.22 ± 2.03	96.34 ± 2.57	98.70 ± 3.46
7	80.98 ± 2.29	85.34 ± 3.06	88.22 ± 0.58	91.09 ± 0.18	92.20 ± 0.65	93.84 ± 2.91	93.97 ± 1.78	95.20 ± 4.09	96.91 ± 1.97
8	82.63 ± 4.41	87.03 ± 4.19	89.17 ± 2.02	92.97 ± 2.56	93.67 ± 3.68	94.29 ± 3.07	95.56 ± 2.26	96.77 ± 3.67	98.26 ± 3.49
9	81.06 ± 1.94	86.05 ± 3.43	88.06 ± 1.24	90.38 ± 2.69	91.18 ± 0.39	92.45 ± 0.37	93.71 ± 0.13	94.05 ± 2.14	96.21 ± 3.16
OA(%)	82.26 ± 0.19	87.43 ± 3.74	92.56 ± 2.37	94.17 ± 3.25	95.48 ± 1.82	96.25 ± 3.24	96.78 ± 0.34	97.14 ± 3.65	98.15 ± 0.28
AA(%)	84.09 ± 1.07	86.02 ± 2.75	91.66 ± 3.10	93.26 ± 0.28	93.07 ± 1.44	95.23 ± 0.21	95.64 ± 1.36	96.08 ± 2.17	97.49 ± 0.98
100 K	80.37 ± 3.26	85.49 ± 4.12	90.21 ± 4.32	93.67 ± 1.49	94.18 ± 0.98	95.98 ± 3.76	96.02 ± 2.37	97.54 ± 3.68	98.44 ± 0.65

Figures 13–15 show the classification diagram of different methods for all datasets of 10% training samples. Compared with the classical EMP-SVM method and deep learning-based DCNN, SSRN, ResNet and other methods, the proposed classification model in this paper has more accurate classification results. Taking Pavia University dataset as an

example, the traditional RBF-SVM and EMP-SVM methods have many noise points in classification results.

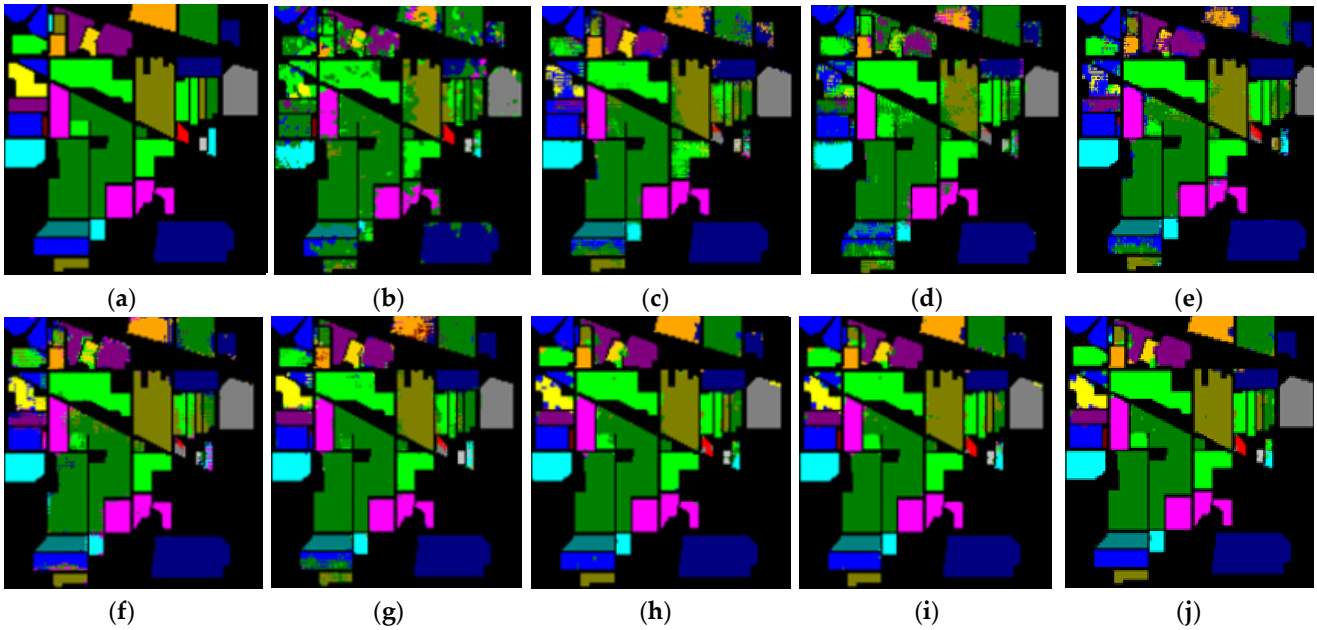


Figure 13. The classification results of Indian pines dataset. (**a**) Ground truth; (**b**) RBF-SVM; (**c**) EMP-SVM; (**d**) DCNN; (**e**) SSRN; (**f**) ResNet; (**g**) PyResNet; (**h**) RepMLP; (**i**) IMLP; (**j**) IMLP-ResNet.

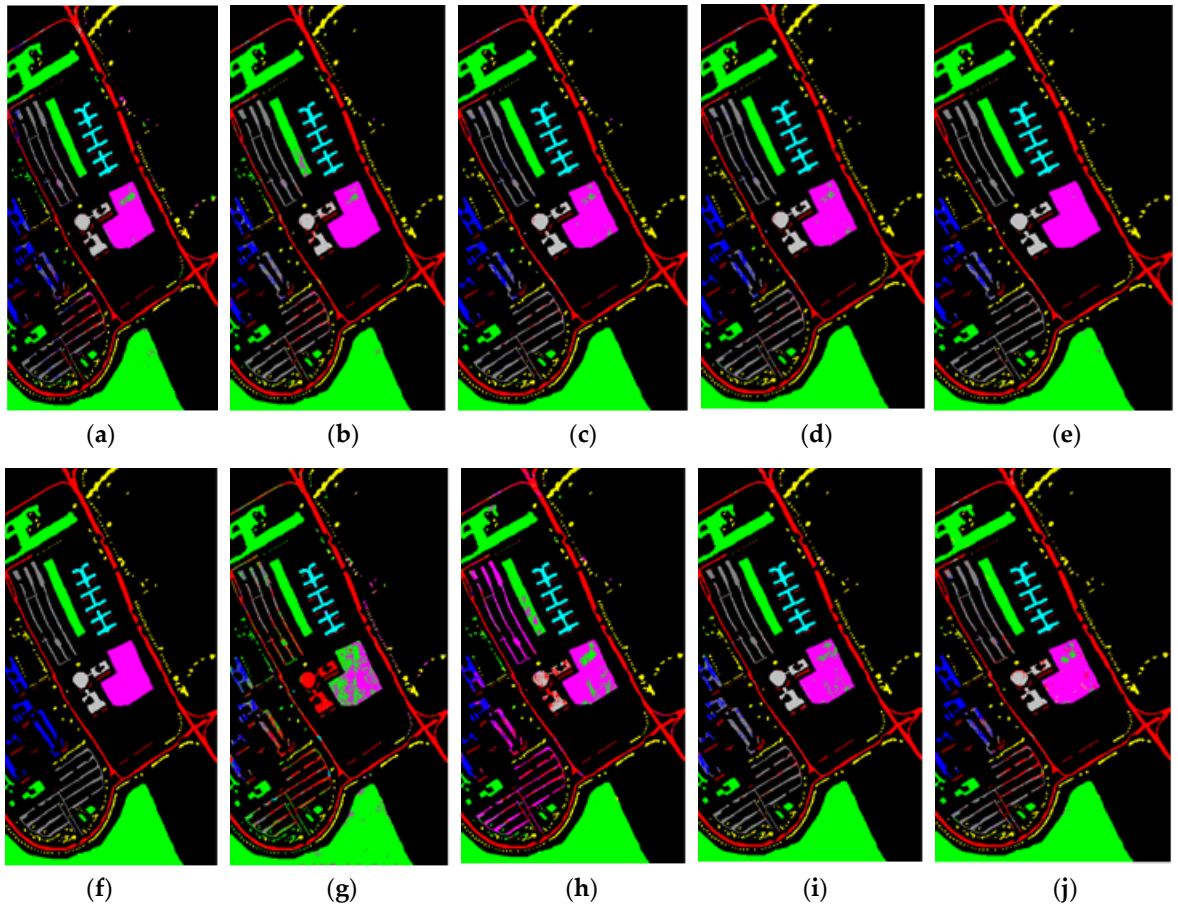


Figure 14. The classification results of Pavia University dataset. (**a**) Ground truth; (**b**) RBF-SVM; (**c**) EMP-SVM; (**d**) DCNN; (**e**) SSRN; (**f**) ResNet; (**g**) PyResNet; (**h**) RepMLP; (**i**) IMLP; (**j**) IMLP-ResNet.

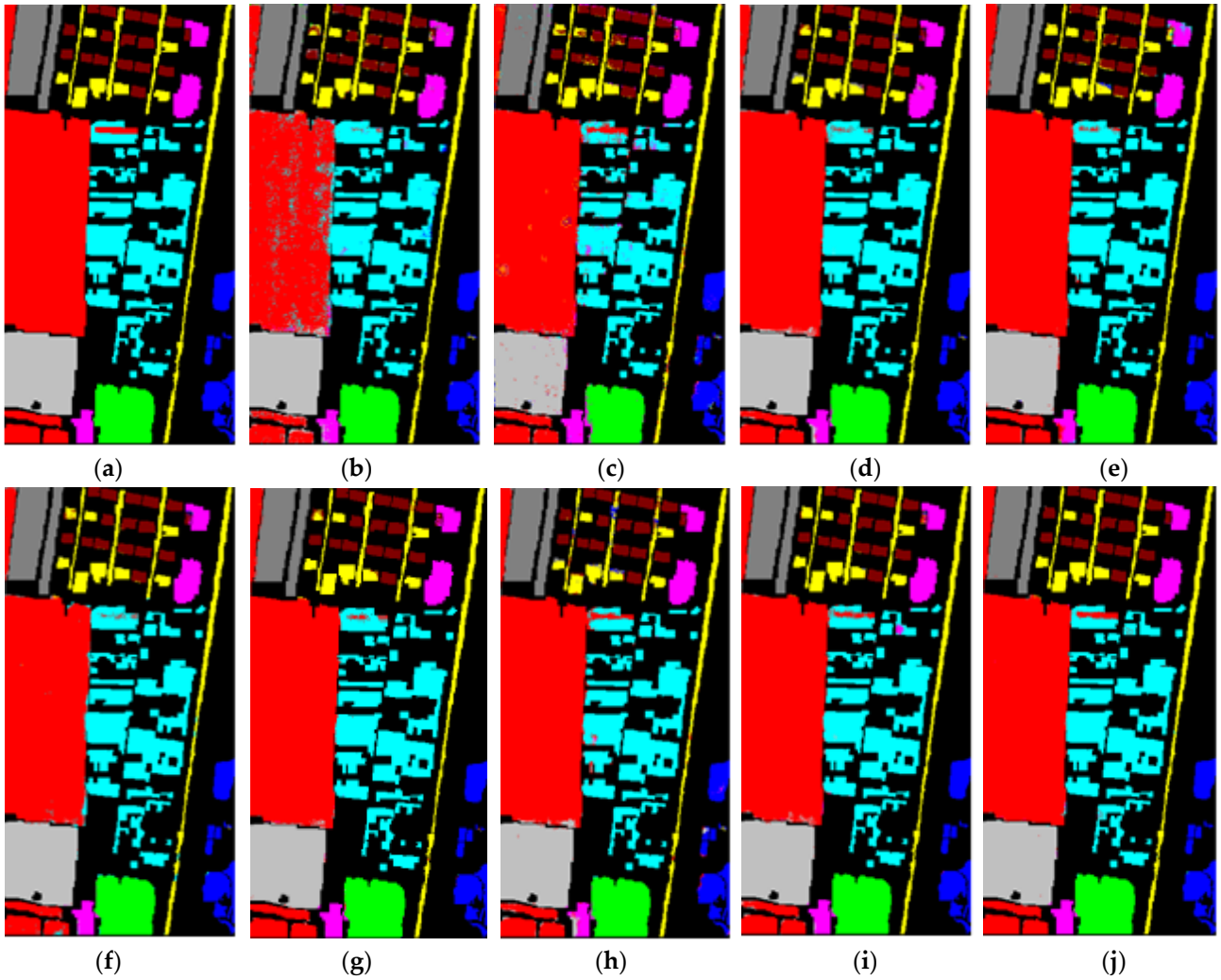


Figure 15. The classification results of Xuzhou dataset. (**a**) Ground truth; (**b**) RBF-SVM; (**c**) EMP-SVM; (**d**) DCNN; (**e**) SSRN; (**f**) ResNet; (**g**) PyResNet; (**h**) RepMLP; (**i**) IMLP; (**j**) IMLP-ResNet.

As shown in Figure 14, parts of the traffic land are mistakenly classified as grassland, and the classification accuracy of ground objects is relatively low. Compared with SVM, DCNN and SSRN classification methods, the classification effect of ResNet and PyResNet is improved, but there are still some misclassification phenomena. However, the IMLP-ResNet model can make full use of each convolutional layer and feature map, and the classification effect is greatly improved. It also eliminates block misclassification and protects edge information. Experiments show that IMLP–ResNet can effectively extract more refined features from three kinds of data sets and cross-dimensional information interaction focuses on more important features, thus improving the classification accuracy.

Figure 15 shows the classification results of the Xuzhou dataset. Xuzhou is an important coal-producing area in China, and coal mining areas may lead to surface subsidence and soil quality degradation, which threatens the safety of residential areas and crop planting. At the same time, it may induce secondary geological disasters. Figure 15 can reflect the land-use situation of the mining area. According to the current classification results, there is still a large area of cultivated land around the tailings pond. By classifying all kinds of ground objects in the test area, we can understand the distribution of the tailings pond, which is helpful to the later mining area.

4. Discussions

In order to find the optimal architecture, it is necessary to do experiments with different main parameters, which plays a crucial role in the size of the model and the complexity of the proposed IMLP–ResNet. By comparing the overall accuracy of different parameters, the influence of these parameters on the model can be analyzed. In the Indian Pines dataset, Pavia University dataset and Xuzhou dataset, the improvement effects of different parameter changes on the model are shown in Figures 16–19.

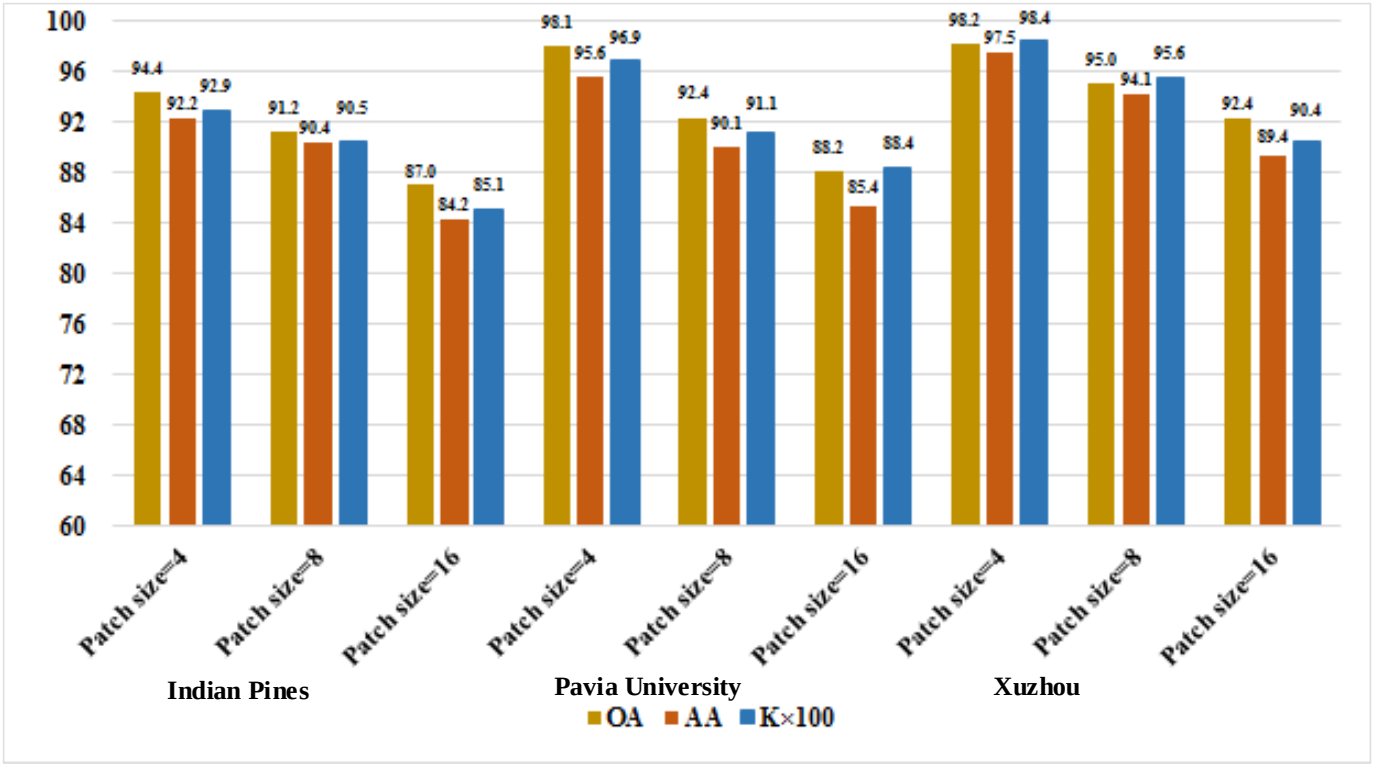


Figure 16. Classification results comparison of IMLP-ResNet with different patch sizes.

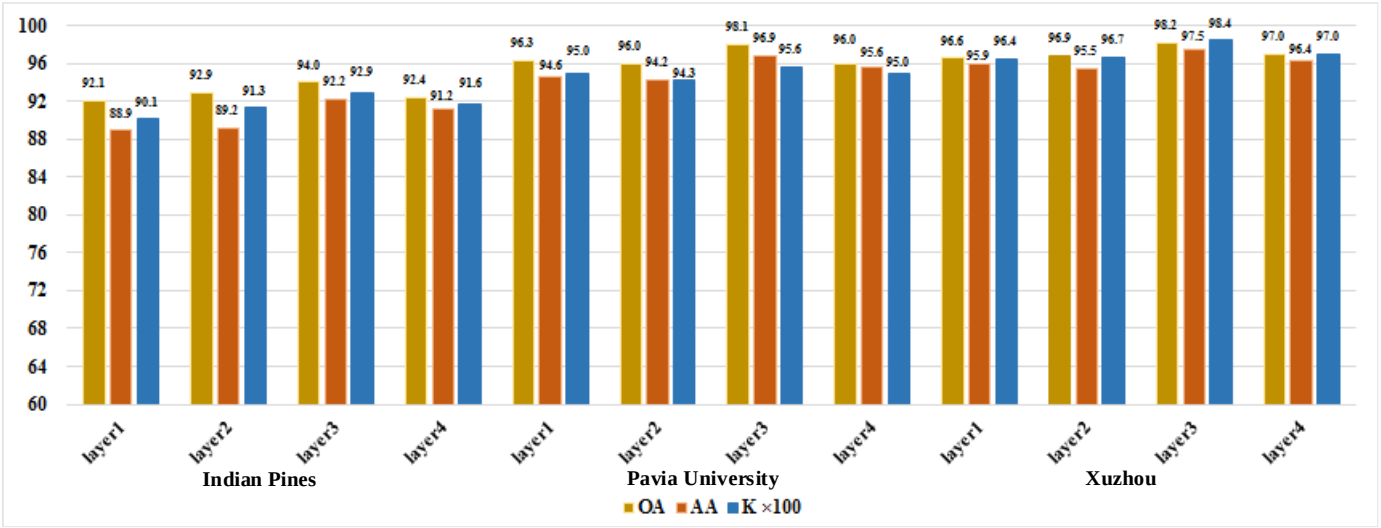


Figure 17. Classification results comparison of IMLP inserted ResNet in different layers.

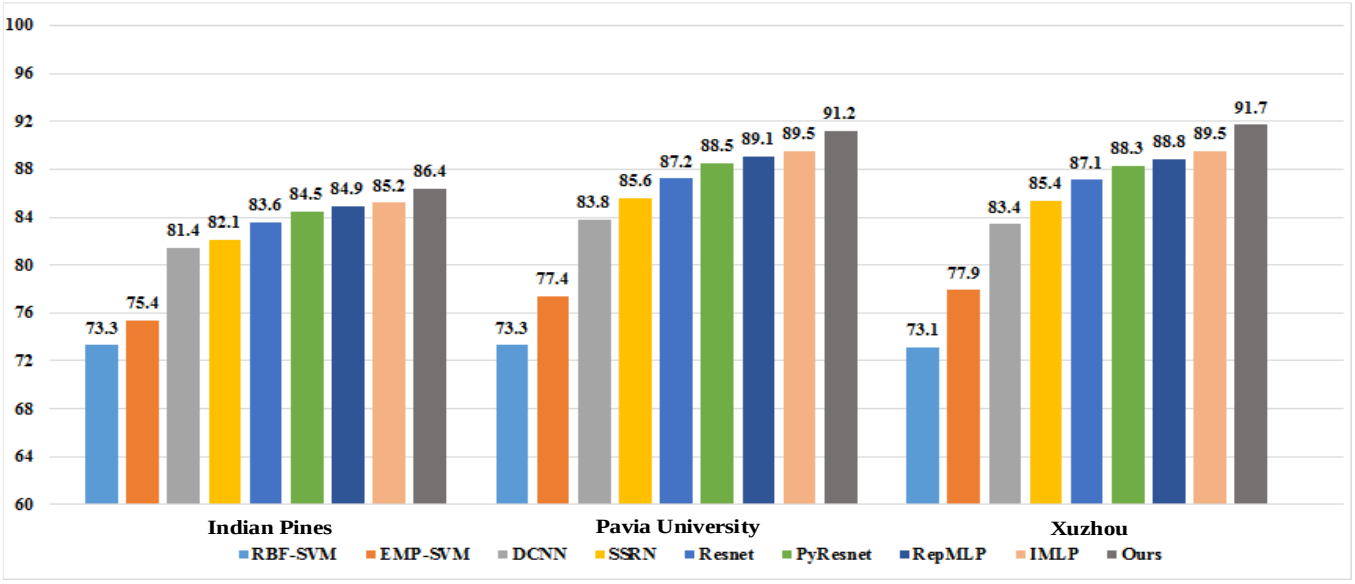


Figure 18. Test accuracy (%) comparisons under different methods on the three datasets with 5% training samples.

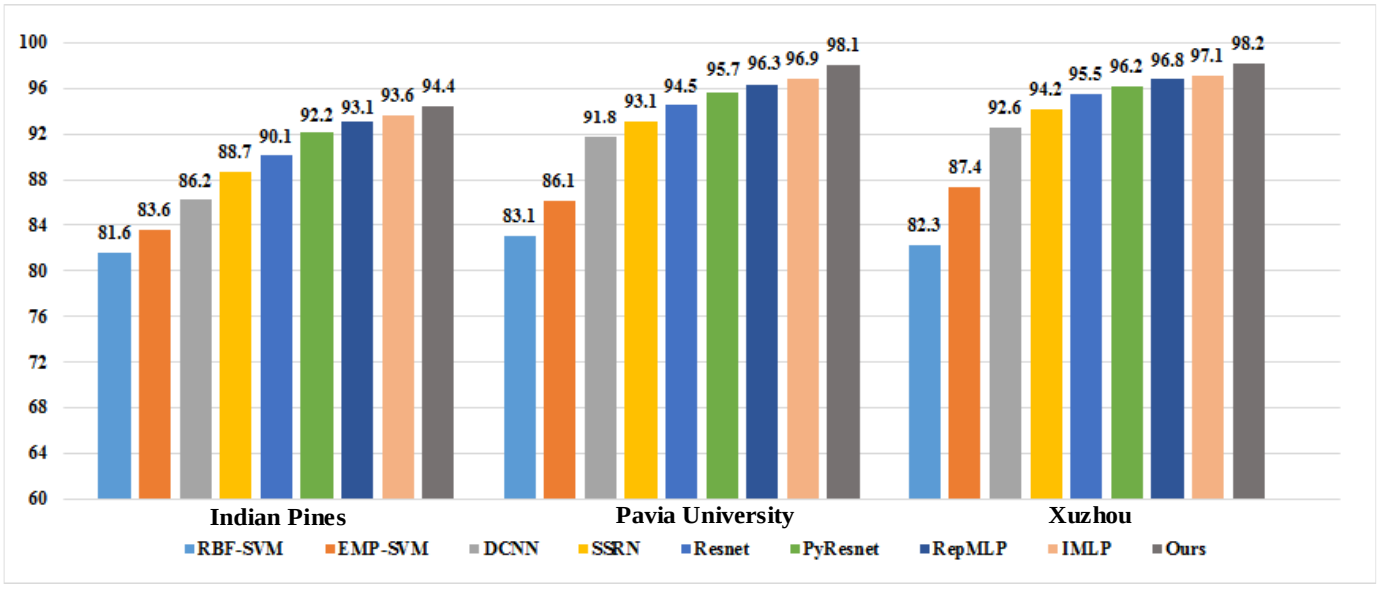


Figure 19. Test accuracy (%) comparisons under different methods on the three datasets with 10% training samples.

The first parameter was experimentally verified in a different patch size. The hyperspectral images were first divided into fixed-size patches to input IMLP-ResNet, and patch sizes were set as 4×4, 8×8, 16×16 respectively. The corresponding input dataset is divided into $4 \times 4 \times nBand$, $8 \times 8 \times nBand$, and $16 \times 16 \times nBand$. As shown in Figure 16, for the three datasets, OA, AA and Kappa coefficients all showed a decreasing trend with the increase in patch size. When patch size = 4, the IMLP-ResNet model proposed achieves the best classification accuracy, because the correlation between the internal information of image patches weakens with the increase in patch size.

The second parameter is to choose the layer of ResNet into which the proposed IMLP module should be inserted to get the best classification results. As shown in Figure 17, we can conclude that the IMLP module inserted into the third layer of ResNet has the highest accuracy in the three datasets. This is because the number of residual blocks in ResNet34 is [3,4,6], that is, the number of residual blocks in the third layer is more than that in the

other three layers. The IMLP module inserted in the third layer of ResNet has a deeper network than the other three layers, which has a stronger ability to extract deeper features of hyperspectral images, so the classification accuracy is higher.

The third parameter is the proportion of training samples to the total samples. The patch size is set to 4 and IMLP module is the third layer of ResNet; 5% and 10% of training samples are taken from the three data sets, respectively, as shown in Figures 18 and 19.

It can be seen from Figures 18 and 19 that, when the number of training samples accounts for 10% of the total samples, the OA is higher than when the number of training samples accounts for 5% of the total samples. This is because the more training samples exist, the more accurately the model can estimate the data distribution, thus the better the generalization performance in the validation set, which leads to higher accuracy. The above results show that when the patch size is 4, the IMLP module is inserted into the third layer of ResNet, and the number of training samples accounts for 10% of the total number of samples, the three datasets can achieve the best classification performance with our proposed IMLP-ResNet.

5. Conclusions

In this paper, two HSI classification frameworks based on MLP are proposed: the IMLP model and IMLP–ResNet. Firstly, according to the characteristics of HSI, three improvements were made to the original model and the IMLP was designed. Secondly, in order to improve the network performance without increasing the amount of inference computation, we introduced a deep over-parameterized convolution layer instead of ordinary convolution. Thirdly, in order to enable the network to learn more useful hyperspectral image information and suppress useless features, we used a Focal Loss function to enhance the key spectral spatial features in the classification task. Finally, in order to avoid oscillation, a cosine annealing algorithm is introduced to accelerate the convergence of the model. The residual structure can retain the original characteristics of this data, avoid the problems of gradient explosion and gradient disappearance in the training process, and improve the modeling ability of the model. In addition, IMLP can improve the feature extraction capability of ResNet, so that the model can enhance the key features while preserving the original features of hyperspectral data. Therefore, in this paper, we proposed IMLP–ResNet, which can extract 3D spectral–spatial features at different levels of the network and learn more comprehensive feature representation for HSI classification.

The proposed IMLP and IMLP–ResNet were tested on two public datasets (Indian Pine and Pavia) and a real HSI dataset (Xuzhou). Compared with the classic methods and deep learning-based methods, the proposed IMLP and IMLP-ResNet show obvious improvements. The results show that the proposed IMLP algorithm and IMLP–ResNet algorithm are meaningful and can obtain better classification results in HSI classification.

However, in the task of hyperspectral image classification, the available marker samples are usually very limited. When analyzing the classification effect of the number of training samples, the IMLP–ResNet proposed in this paper finds that the effect of 10% of the number of samples is better than 5%. Therefore, in the next step, we will consider data expansion, active learning, transfer learning, meta learning and other technologies to realize the construction and design of a network model combined with MLP under small samples. In addition, the means of using unlabeled samples more effectively for semi-supervised hyperspectral classification based on MLP is also worthy of further research.

Author Contributions: Conceptualization, A.W. and H.W.; methodology, software, validation, M.L.; writing—review and editing, H.W. and A.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China under Grant NSFC-61671190.

Data Availability Statement: The data are available at http://www.ehu.eus/ccwintco/index.php?%20title=Hyperspectral-Remote-Sensing-Scenes (accessed on 21 February 2022).

Acknowledgments: We thank Kaiyuan Jiang for his valuable comments and discussion.

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

References

1. Hong, D.; Wu, X.; Ghamisi, P.; Chanussot, J.; Yokoya, N.; Zhu, X.X. Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3791–3808. [CrossRef]
2. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
3. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
4. Zhang, X.; Sun, Y.; Shang, K.; Zhang, L.; Wang, S. Crop classification based on feature band set construction and object-oriented approach using hyperspectral images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4117–4128. [CrossRef]
5. Shimoni, M.; Haelterman, R.; Perneel, C. Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 101–117. [CrossRef]
6. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790.
7. Ma, L.; Crawford, M.M.; Tian, J. Local manifold learning-based-nearest-neighbor for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4099–4109. [CrossRef]
8. Ham, J.; Chen, Y.; Crawford, M.M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
9. Delalieux, S.; Somers, B.; Haest, B.; Spanhove, T.; Borre, J.V.; Mücher, C.A. Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers. *Remote Sens. Environ.* **2012**, *126*, 222–231. [CrossRef]
10. Li, W.; Chen, C.; Su, H.; Du, Q. Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3681–3693. [CrossRef]
11. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]
12. He, L.; Li, J.; Liu, C.; Li, S. Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1579–1597. [CrossRef]
13. Zhan, K.; Wang, H.; Huang, H.; Xie, Y. Large margin distribution machine for hyperspectral image classification. *J. Electron. Imaging* **2016**, *25*, 63024. [CrossRef]
14. Song, B.; Li, J.; Dalla Mura, M.; Li, P.; Plaza, A.; Bioucas-Dias, J.E.M.; Benediktsson, J.A.; Chanussot, J. Remotely sensed image classification using sparse representations of morphological attribute profiles. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 5122–5136. [CrossRef]
15. Xue, F.; Tan, F.; Ye, Z.; Chen, J.; Wei, Y. Spectral-spatial classification of hyperspectral image using improved functional principal component analysis. *IEEE Trans. Geosci. Remote Sens.* **2021**, *19*, 1–5. [CrossRef]
16. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
17. Li, T.; Zhang, J.; Zhang, Y. Classification of hyperspectral image based on deep belief networks. In Proceedings of the 2014 IEEE International Conference on Image Processing, Paris, France, 27 October 2014.
18. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium, Milan, Italy, 26–31 July 2015; pp. 4959–4962.
19. 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]
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* **2016**, arXiv:1512.03385.
21. 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]
22. Paoletti, M.E.; Haut, J.M.; Fern; Ez-Beltran, R.; Plaza, J.; Plaza, A.J.; Pla, F. Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 740–754. [CrossRef]
23. Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv* **2021**, arXiv:2105.01601.
24. Liu, H.; Dai, Z.; So, D.; Le, Q. Pay attention to MLPs. *Adv. Neural Inf. Processing Syst. arXiv* **2021**, arXiv:2105.08050.
25. Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Jégou, H. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv* **2021**, arXiv:2105.03404.
26. Tatsunami, Y.; Taki, M. Raftmlp: How much can be done without attention and with less spatial locality? *arXiv* **2021**, arXiv:2108.04384.
27. Zhang, M.; Zuo, X.; Chen, Y.; Liu, Y.; Li, M. Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization. *IEEE Trans. Robot.* **2021**, *37*, 1081–1099. [CrossRef]

28. Cao, J.; Li, Y.; Sun, M.; Chen, Y.; Lischinski, D.; Cohen-Or, D.; Chen, B.; Tu, C. DO-Conv: Depthwise over-parameterized convolutional layer. *arXiv* **2020**, arXiv:2006.12030.
29. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef]
30. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
31. Yuan, Y.; Wang, C.; Jiang, Z. Proxy-based deep learning framework for spectral-spatial hyperspectral image classification: Efficient and robust. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [CrossRef]
32. Chen, W.; Zheng, X.; Lu, X. Hyperspectral image super-resolution with self-supervised spectral-spatial residual network. *Remote Sens.* **2021**, *13*, 1260. [CrossRef]
33. Feng, J.; Wu, X.; Shang, R.; Sui, C.; Zhang, X. Attention multibranch convolutional neural network for hyperspectral image classification based on adaptive region search. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5054–5070. [CrossRef]
34. Melgani, F.; Bruzzone, L. Support vector machines for classification of hyperspectral remote-sensing images. In Proceedings of the 2002 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2002), Toronto, ON, Canada, 24–28 June 2002.
35. Gu, Y.; Liu, T.; Jia, X.; Benediktsson, J.O.N.A.; Chanussot, J. Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3235–3247. [CrossRef]
36. Yue, J.; Zhao, W.; Mao, S.; Liu, H. Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* **2015**, *6*, 468–477. [CrossRef]
37. Zhong, Z.; Li, J.; Ma, L.; Han, J.; He, Z. Deep residual networks for hyperspectral image classification. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017.
38. Ding, X.; Xia, C.; Zhang, X.; Chu, X.; Han, J.; Ding, G. Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv* **2021**, arXiv:2105.01883.