Journal of Computers Vol. 31 No. 4, 2020, pp. 174-186
doi:10.3966/199115992020083104013

Spatial Feature Extraction for Hyperspectral Image Classification Based on Multi-scale CNN

Haifeng Song, Weiwei Yang*

School of Electronics and Information Engineering (School of Big Data Science), Taizhou University, Taizhou 318000, China
yww_1680@163.com

Received 10 March 2020; Revised 6 April 2020; Accepted 6 May 2020

Abstract. In recent years, convolutional neural network has been widely used in the field of computer vision and achieved good results in image classification. In the field of remote sensing image processing, the demand for hyperspectral image (HSI) classification is also increasing. However, as the training of CNN requires large amounts of labeled hyperspectral images, convolutional neural networks are difficult to apply to this domain. Considering this problem, we proposed a hyperspectral image classification model with multi-scale convolutional neural networks. The input of the model is the original hyperspectral image and the output is the final classification result. The characteristics of this model are as follows. First, the model can extract spatial features of the input data from multi-scales automatically, rather than requiring handcrafted features. Second, multi-scales feature extraction is adopted, and more samples are involved in the classification, solving the problem of obtaining large amounts of labeled hyperspectral data. Third, the model proposed in this paper consists of a multi-scale convolutional spatial feature extraction layer, a feature fusion layer, a normalization layer, a dropout layer and an activation layer, which are more suitable for hyperspectral image classification. Finally, the experimental results for the Indian Pines dataset show that the classification model proposed in this paper is better than other state-of-the-art classification models in terms of the overall accuracy, average accuracy and the Kappa coefficient.

Keywords: convolutional neural network, feature extraction, multi-scale, hyperspectral image classification, spatial feature

1 Introduction

Recently, convolutional neural networks (CNNs) have been widely applied in the field of computer vision [1], for example, moving object detection [2], image feature extraction [3], image classification [4], etc. The classification process of hyperspectral remote sensing image is similar to that of traditional image. The difference between them is that traditional images contain only 3-4 channels, while hyperspectral images contain at least 200 channels. In practical applications, a large number of labeled data sets can be used to classify traditional images, such as ImageNet [5]. However, the target of hyperspectral image classification is to classify each pixel in the image, and it is time-consuming and laborious to label hyperspectral remote sensing image, so the labeled samples is very small. Therefore, the problem of small sample set (3S) is an urgent problem for hyperspectral remote sensing image classification.

A common solution to this problem is to reduce the dimensionality of the data, which can also address the influence of the Hughes effect on classification accuracy [6-7]. Dimensionality reduction is the projection of high-dimensional data to a lower dimensional space. The common projection based dimensionality reduction algorithms include principal component analysis [8], independent component analysis [9], manifold learning [10], etc. Another way to reduce the dimensionality is to extract the most

* Corresponding Author

representative band images from the HSI data, e.g., via a sorting algorithm [11], clustering algorithm [12], etc. However, one of the problems with dimensionality reduction is that some important information will be lost. Also, after dimensionality reduction, the features extracted from the spectral domain are often not enough to represent a class of real objects. Therefore, it is necessary to find more representative data features.

But some important information may be lost during the dimensionality reduction process. More severely, the features obtained through dimensionality reduction in the spectral domain cannot fully characterize the properties of the materials; hence more discriminative features should been extracted.

The spatial information contained in HSI data can be used to determine the spatial characteristics of adjacent pixels. Therefore, spatial features make up for the shortage of spectral domain features. These spatial features can further improve the accuracy of HSI classification. In addition to geometric morphometric [13], there are many other spatial filters that can be used to extract HSI spatial features. However, in order to train a well model, most of the models need a large number of labeled samples; it is difficult to meet this requirement in practical application. In order to extract the spectral and spatial features further, researchers in the area have proposed the use of Gabor filters to extract them simultaneously, which has produced good results in small scale HSI classification [14]. Recent researches have shown that the convolutional neural network can automatically learn the parameters of the convolutional filter, while the Gabor filter with different directions and scales is actually a convolutional filter. [15]. This conclusion further proves that it is feasible to extract spatial features by using multi-scale convolutional neural network in this paper. A typical spatial feature extraction algorithm is geometric morphometric, which mainly transforms open or closed areas, thus extracting the spatial structure features [16].

A typical convolutional neural network is usually composed of several convolutional layers and pooling layers. Generally, after the convolution layer, the activation function is applied to extract the image features. The commonly used activation functions are Relu, Sigmoid and Tanh functions. The pooling layer combines adjacent pixels to extract local features. The role of the pooling layer is to improve robustness regarding slight distortions of the image. CNNs have been widely applied in the computer vision field. LeCun uses backward propagation and a gradient descent algorithm to train a CNN, and applies the trained network model to handwritten digit recognition [17]. Krizhevsk has applied a convolutional neural network to classify the ILRSRC2012 datasets containing 1000 different classes of images. The classification accuracy surpassed that of the previous approaches for handwritten feature classification [5]. Since 2012, CNNs have been widely applied in the fields of image classification, semantic segmentation, target recognition, video analysis, etc. Literature [18] proposed a convolutional neural network to classify HSIs. It takes the original HSI data as input and outputs the corresponding pixel category. There are 200 training samples for each category.

There are some deep learning related works on HSI classification in the literature. Such as in [19], deep stacked auto encoders are employed to extract features. The auto encoder is a kind of unsupervised method. The proposed method in [19] combines principle component analysis (PCA), auto encoders and logistic regression, and it is not an end-to-end deep method. An end-to-end deep CNN method is proposed in [20]. The method takes the raw data as the input and outputs the predicted class labels. The number of training samples of each class is 200, and it is relative large number.

This paper first analyzes the basic architecture, principles and training methods for convolutional neural networks. In addition, it presents a classification model for HSI based on multi-scale convolutional neural networks (Multi-CNNs). Finally, the experiments verify that the accuracy of this model is superior to state-of-the-art algorithms in the classification of HSI. It is demonstrated that the hyperspectral image classification model based on Multi-CNN can be widely used in HSI classification.

The remainder of the paper is organized as follows: In Section 2 the basic principles of cnns is discussed. In Section 3 the proposed HSI classification model based on multi-cnn is described in detail. The experimental results and analysis are presented in Section 4, followed by the conclusion in Section 5.

2 The Basic Principles of CNN

As shown in Fig. 1, a typical convolutional neural network mainly consists of input layer, convolution layer, pooling layer, fully connected layer and output layer.

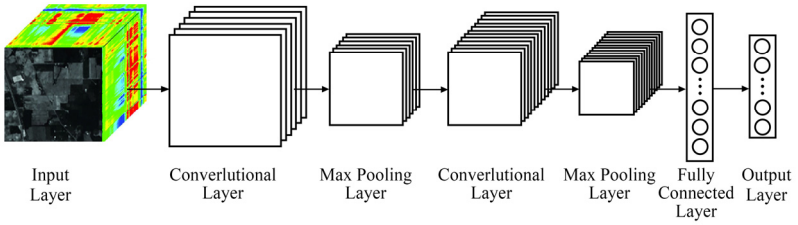

Fig. 1. The typical convolutional neural network architecture

Normally, the input layer of the convolutional network is the original image X. In this paper, H_i is the i^{th} layer of the convolutional neural network of the feature map ($H_0 = X$). H_i can be obtained as formula (1):

$$H_i = f(H_{i-1} \otimes W_i + b_i) \tag{1}$$

where W_i is the weight vector of the convolutional kernel of the ith selected layer. The operation symbol $\otimes$ is the convolution operation, which represents the convolution operation between the image of the ith level and the $(i-1)^{th}$ for the feature map. Then, the output of the convolutional layer and bias vector b_i is add, and finally the result is applied to the nonlinear activation function to obtain the feature map of the i-th layer

The convolutional layer is followed by the pooling layer, which is used to down sample the feature map according to certain rules. The pooling layer performs two tasks : (1) dimensionality reduction of the feature map; (2) keep the scale invariance of the feature graph. Assuming H_i is the pooling layer, there are formula (2):

$$NLL(W,b) = -\sum_{i=1}^{|Y|} \log Y_i \tag{2}$$

After alternately calculating the multiple convolutional and pooling layers, the extracted features are classified by the fully connected layer of the CNN, and the probability distribution Y is obtained based on the input (l_i represents the i^{th} tag category). As shown in formula (3), the CNN makes the original matrix (H_0) transform the data or reduce the dimensionality through multiple layers essentially, and maps it to a new mathematical model of feature expression.

$$H_i = subsampling(H_{i-1}) \tag{3}$$

The ultimate goal of training the convolutional neural network is to minimize the loss function $L(W,b)$. We calculate the forward propagation of the input H_0 to obtain the value of the loss function, and use the value of the loss function to calculate the difference between the input H_0 and the expected value. The difference is called the "residual". The common loss functions include: the Mean Squared Error (MSE) function as shown in formula (4), the Negative Log Likelihood (NLL) function as shown in formula (5), etc.

$$Y_i = P(L = l_i \mid H_0 : (W,b)) \tag{4}$$

$$MSE(W,b) = \frac{1}{|Y|}\sum_{i=1}^{|Y|}(Y_i - \hat{Y}_i)^2 \tag{5}$$

In order to alleviate the problem of overfitting, the L2 norm is used to control the overfitting of weights, and the intensity is controlled by parameter λ (weight decay), as shown in formula (6).

$$E(W,b) = L(W,b) + \frac{\lambda}{2}W^T W \tag{6}$$

In the training process, the gradient descent algorithm is usually used to optimize CNNs. In the process of back propagation, gradient descent method is usually used to update the trainable parameters (w,b) layer by layer. The parameter learning rate η is used to control the strength of back propagation. As shown in formula (7) and (8)

$$W_i = W_i - \eta \frac{\partial E(W,b)}{\partial W_i} \tag{7}$$

$$b_i = b_i - \eta \frac{\partial E(W,b)}{\partial b_i} \tag{8}$$

3 The HSI Classification Model Based on Multi-CNN

In this paper, we proposed a convolutional neural network model based on multi-scale feature extraction, according to the basic principle of neural network. The model is divided into two parts: the first part is multi-scale feature extraction and fusion, the second part is HSI classification, as shown in Fig. 2 and Fig. 3.

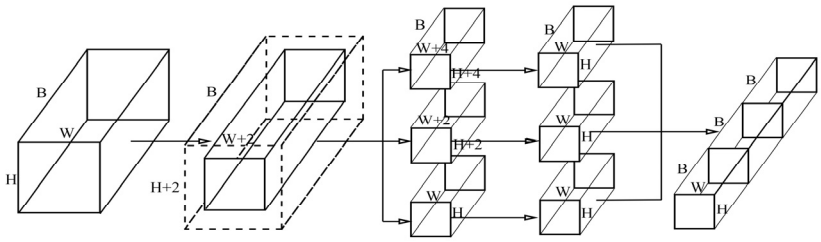

Fig. 2. Multi-scale feature extraction and fusion model

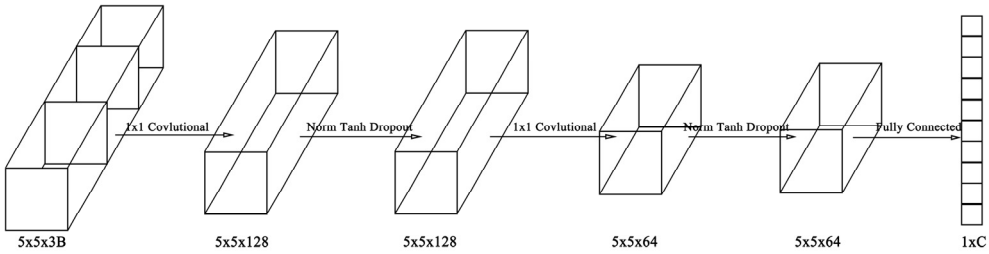

Fig. 3. HSI classification model based on Multi-CNN

3.1 Multi-scale Feature Extraction and Fusion Model

The architecture of the multi-scale feature extraction and fusion model proposed in this paper is shown in Fig. 2. It first uses the original HSI as input, and then calculates the convolution of the original images using a convolutional kernel of $1\times1\times B$, $3\times3\times B$, $5\times5\times B$ (B is the number of HSI bands). The convolved image can be used to extract the spatial features of HSI. The outputs of the convolutional layer are three sets of convolutional feature maps. Finally, we apply the fully connected technology of feature fusion to three sets of convolutional feature maps, to obtain a feature map with size $W\times H\times 3B$.

As can be seen from Fig. 2, the size of the convolutional maps calculated by three convolutional kernels of different sizes is not the same. Therefore, these three convolutional kernels need to be adjusted to be the same size. First, two pixels are added around the original image and the pixel value is set to 0. Thus, the expanded images can output a feature map of size $(H+4,W+4),(H+2,W+2),(H,W)$ (H is the height and W is the width of the original image) through convolution. The convolutional kernels are $1\times1,3\times3,5\times5$ and the step size is 1. Then, the convolutional graph is calculated using max pooling of

size $(5\times5),(3\times3)$. Ultimately, the size of all the three convolutional graphs are adjusted to $H\times W$.

3.2 The HSI Classification Model Based on Multi-CNN

As shown in Fig. 3, the multi-scale HSI classification model proposed in this paper is mainly composed of the following parts: three convolutional layers, two normalized layers, two dropout layers, two activation layers and a full connection layer. The input data is an image of size $5\times5\times N$ (N is the number of HSI bands) centered on the HSI pixel point to be classified. The first convolutional layer in this paper uses a 1×1 convolutional kernel. The total number of convolutional kernels is 128, so the size of the feature map output from the first convolutional layer is $5\times5\times128$. There are the normalization layer, the dropout layer and the activation layer after the first convolutional layer. The second convolutional layer uses a $1\times1\times64$ convolutional kernel, so the size of the feature map output from the second convolutional layer is $5\times5\times64$. There is already a normalization layer, dropout layer and activation layer after the second convolutional layer. The third convolutional layer uses a $1\times1\times C$ convolutional kernel (C is the number of categories to be classified), so the size of the feature map output from the third convolutional layer is $5\times5\times C$. The last layer is the pooling layer for global evaluation whose input is a feature map of size $5\times5\times C$ and the output is the eigenvector of size $1\times1\times C$, where the i^{th} maximum value of the output represents the category of the pixel.

3.3 The Learning Process of the HSI Classification Model Based on Multi-CNN

The learning process of the multi-CNN HSI classification model is shown in Fig. 4. The steps for HSI classification are as follows:

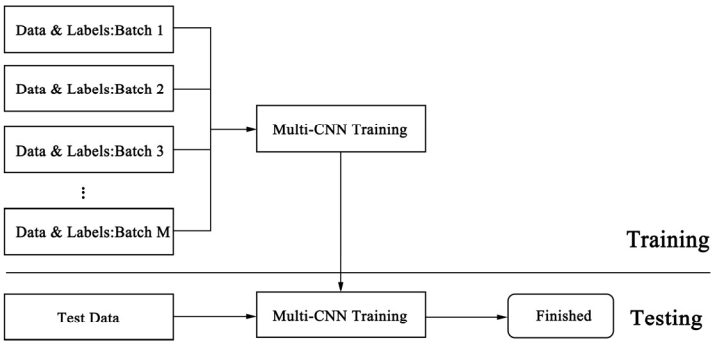

Fig. 4. The learning process of the HSI classification model based on Multi-CNN

Step 1. The original training data set is randomly sampled to generate M subsets containing the same amount of data. In this paper, M=16.
Step 2. The 16 subsets are used for training separately by using stochastic gradient descent. Only one subset is iterated at a time. Repeat the process until the maximum value of iterations is reached. Step 3: In the test phase, the test sample is input into the trained Multi-CNN model, and the i^{th} maximum value of the output vector is the category of the sample.

4 Experiments and Result Analysis

The Indian Pines dataset is used in our testing. First, the main components of the HSI model based on Multi-CNN are tested, including the multi-scale convolutional kernel, convolutional kernel size and the impact of the activation function on classification accuracy. Then, the proposed classification model is compared with mainstream approaches in terms of the overall accuracy, average accuracy and kappa coefficient. The hardware and software environment used in the experimentation is shown in Table 1.

Table 1. Experimental environment: hardware and software

Category	Item	Parameter
Hardware environment	CPU	Intel Celeron CPU E3400 @2600GHZ
	GPU	NVIDIA GeForce GTX 950
	Memory	6 GB
	Hard Disk	500 GB
Software environment	Operating System	Ubuntu 16.04
	Deep Learning Tools	TensorFlow 1.2.0
	Compute Unified Device Architecture	CUDA Toolkit 8
	CUDA Deep Neural Network	cuDNN 5.1
	Development language	Python 3.5

4.1 Brief Introduction to The Experimental Dataset

The dataset (Indian Pines) was collected by AVRIS on 12 June, 1992. The area of the hyperspectral image is a farm at Purdue University in northwest Indiana. The reason for selecting this dataset is that it is a standard hyperspectral reference dataset, and it contains accurate labels to cover the truth value. This is beneficial to repeat experiments and algorithm reappearance. The dataset is widely used in hyperspectral image classification research.

The data band covers a range of 400-2500 nm. The original data contains 220 spectral channels. The spatial resolution is 20 m. Each band contains 145×145 pixels. Fig. 5 is the grayscale image of the hyperspectral data with the band 50, 27 and 17. Fig. 6 is the ground truth image of the Indian Pines. The dataset has 16 land cover categories, and the sample size is shown in Table 2.

Fig. 5. The grayscale image of the Indian Pines

Fig. 6. The ground truth image

Table 2. Indian Pines data sample distribution

No.	Class	Num of Samples
1	Alfalfa	54
2	Corn-notill	1434
3	Corn-mintill	834
4	Corn	234
5	Grass/pasture	497
6	Grass/tree	747
7	Grass/pasture/mowed	26
8	Hay/Windrowed	489
9	Oats	20
10	Soybean-notill	968
11	Soybean-mintill	2468
12	Soybean-clean	614
13	Wheat	212
14	Woods	1294
15	Buildings/grass/trees/drives	95
16	Stone/steel/towers	380
	Total	10366

4.2 Effect of the Multi-scale Convolutional Kernel on Classification Accuracy

This experiment verified the influence of the size of the multi-scale convolutional kernel on the classification results based on spatial feature extraction. First, the basic structure of the multi-scale convolutional kernel was investigated, as shown in Fig. 7. Second, the effects of the convolutional kernel with different sizes $(1\times1, 3\times3, 5\times5, 7\times7)$ were investigated, as shown in Table 3. Finally, the classification accuracy when applying the multi-scale convolutional kernel was found to be better than that of using the convolutional kernel alone. The experimental results illustrate that the classification accuracy for the multi-scale convolutional kernel is 7.84% higher than that obtained when using the 1×1 convolutional kernel alone.

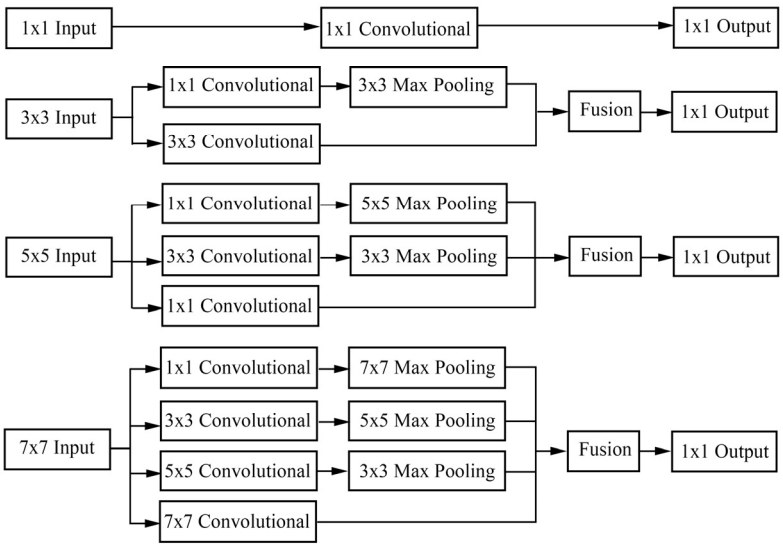

Fig. 7. The basic structure of the multi-scale kernel

Table 3. The effect of different convolutional kernels on the classification results

Dataset	1x1	1x1-3x3	1x1-5x5	1x1-7x7
Indian Pines	84.47%	87.69%	92.31%	85.36%

There are two main reasons for the improvement of the classification accuracy: (1) The multi-scale convolutional kernel can extract original HSI features at different scales and generate many different features. (2) The computation of the multi-scale convolutional kernel is equivalent to increasing the number of samples of the original dataset, which makes the training process more effective. At the same time, it is found that the classification accuracy does not improve when increasing the multi-scale convolutional kernel. When sizes of $1\times1-7\times7$ multi-scale convolutional kernels are applied, the accuracy decreases because of overfitting. Therefore, the three sizes of convolutional cores $(1\times1,3\times3,5\times5)$ are selected in this paper.

As can be seen from Fig. 8, with the increase of convolutional kernel size involved in classification, the classification accuracy improves, and the maximum of the overall accuracy is at 1×1-5×5 pixels. However, the overall accuracy will decrease if the convolutional kernel size is further increased. This is because when the convolutional kernel size involved in classification is relatively small, there is too little neighborhood information to be extracted, so the overall accuracy is lower. When the convolutional kernel size involved in classification is increased, the extracted neighborhood information increases and the overall accuracy are improved. But when the convolutional kernel size is increased further, the overall accuracy decreases. As the classification kernel size increases there are fewer broken sections with increased continuity, and the accuracy improves. When it is too large, more broken sections appear and the accuracy is reduced.

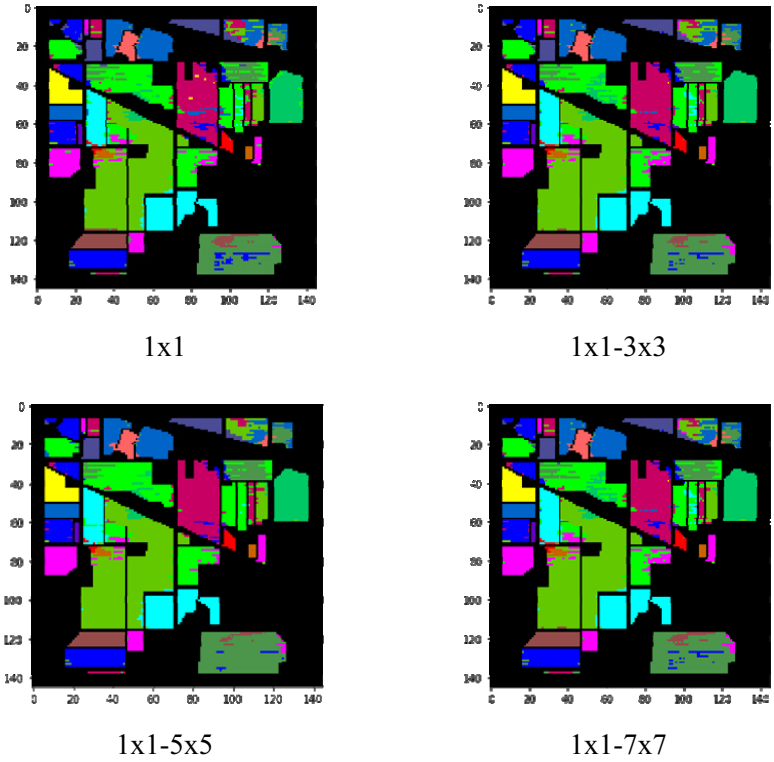

Fig. 8. The classification results of the different convolutional kernel sizes

4.3 Effect of Convolutional Kernel Size on Classification Accuracy

This experiment tests the influence of convolutional kernels with different sizes on the training and testing process. A trained CNN model should exhibit close to zero loss on the training dataset, and also on the test dataset. If the loss value of the model is close to zero on the training dataset and very large on the test dataset, it shows that the CNN has overfitted, and the generalization ability of such a network will be very poor.

The HSI model based on Multi-CNN (Fig. 3) is trained and tested with a convolutional kernel of size 1×1 and 3×3. The change of loss value during the training and testing process are shown in Fig. 9. It can be seen from this figure that when a 1×1 convolutional kernel is used in training and testing, the

value of the loss function converges to 0 as the number of iterations increases. When a 3×3 convolutional kernel is used in training, the value of the loss function converges slowly to 0 with increasing iterations. However, in the process of testing, the value of the loss function does not converge with increasing iterations. The test shows that when a 3×3 convolutional kernel is used then overfitting occurs. From these experiments we concluded that a CNN with good generalization ability can be obtained using a convolutional kernel of size 1×1.

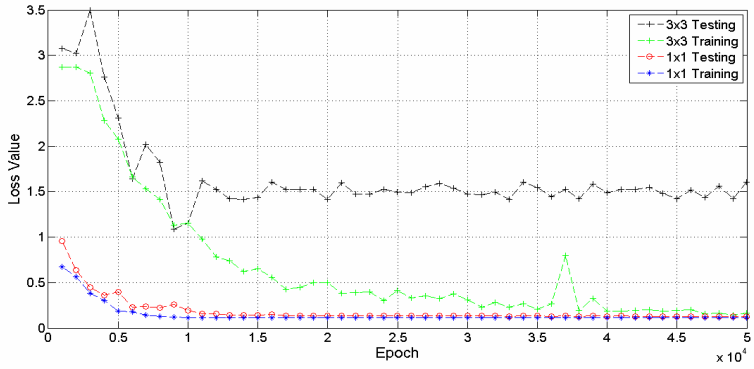

Fig. 9. The change of loss value using different convolutional kernel sizes: training and testing

4.4 Effect of Activation Function on Classification Accuracy

This experiment verified the influence of the choice of activation function on the classification results. In the experiment, the parameters of the Multi-CNN spatial feature extraction model were fixed. The activation function is set to ReLU, Sigmoid, or Tanh. The experimental results were recorded separately. Fig. 10 shows the variation of the overall accuracy corresponding to the three activation functions.

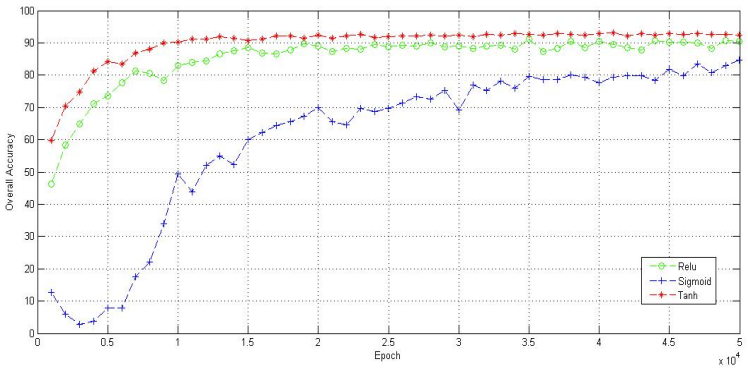

Fig. 10. The variation of the overall accuracy corresponding to the three activation functions

We can see from Fig. 10 that as the iterations increased, the overall accuracy of the three activation functions gradually improved. However, the overall accuracy of the Sigmoid function was significantly lower than that of ReLU and Tanh. The overall accuracy of Tanh and ReLU is essentially the same, but Tanh converges faster. Therefore, Tanh is selected as the activation function in this paper.

4.5 Comparison of Overall Classification Accuracy

In this section, we compared the classification model based on Multi-CNN spatial feature extraction with other HSI classification algorithms (namely, SVM, MLRsub, SVM-GC, and MLRsubMLL). The classification results of each algorithm are compared with respect to the overall accuracy (OA), average accuracy (AA) and Kappa coefficient (Kappa).

Overall accuracy: This is the probability that the result of the classification is consistent with the test data for each random sample. It is equal to the number of pixels in the correct classification divided by the total number of pixels. The calculation is shown in formula (9).

$$OA = \frac{\sum_{i=1}^{K} C(i,i)}{M} \tag{1}$$

Average accuracy: This is calculated by averaging the total accuracy of each category. The calculation is shown in formula (10).

$$AA = \frac{\sum_{i=1}^{K} OA_i}{K} \tag{2}$$

In (9) and (10): $C(i,i)$ is the correct classification for the ith category, M is the total number of pixels, K is the number of categories, OA is the overall accuracy, and AA is the average accuracy

The Kappa coefficient is another method for comparing classification performance. The value of the Kappa coefficient is between -1 and 1, but usually ranges between 0 and 1. The Kappa coefficient considers the influence of uncertainty on the classification results when estimating the recognition accuracy. The calculation is shown in formula (11).

$$Kappa = \frac{M\sum_{i=1}^{K} C(i,i) - \sum_{i=1}^{K} (C(i,+)C(+,i))}{M^2 - \sum_{i=1}^{K} (C(i,+)C(+,i))} \tag{3}$$

In formula (11), M is the total number of pixels, $C(i,i)$ is the correct classification for the ith category, $C(i,+)$ is the number of pixels which are classified to the ith category, $C(+,i)$ is the number of ground truth samples for the ith category.

The experimental results are shown in Table 4. As shown in Table 4, the classification models proposed in this paper are superior to other classification algorithms with respect to the OA, AA and Kappa. Fig.11 represents the different classification algorithms results. The advantage of the classification model proposed in this paper is mainly due to the fact that the model uses CNNs to extract multi-scale spatial features of hyperspectral images. The experiment further confirms that the model proposed in this paper is better than other state-of-the-art classification models. The hyperspectral image classification model based on Multi-CNN spatial feature extraction can be applied to other areas, such as forestry, agriculture, the environment, etc.

Table 4. Results for the different classification algorithms

Class	Classification Algorithm				
	SVM [21]	*MLRsub* [21]	*SVM-GC* [22]	*MLRsubMLL* [23]	*Multi-CNN*
1	73.17	46.34	95.12	95.12	98.53
2	62.65	40.93	68.48	50.04	85.71
3	52.88	26.34	56.49	13.12	83.33
4	32.39	17.37	77.00	15.02	83.24
5	91.24	70.97	94.47	73.04	96.72
6	92.09	94.37	97.72	98.93	92.49
7	36.00	18.18	34.42	37.25	48.35
8	95.58	96.51	100	100	90.00
9	0	22.22	0	0.	81.82
10	61.44	25.06	75.06	19.68	88.49
11	86.92	78.23	95.47	88.82	98.32
12	76.36	16.51	99.44	16.51	33.33
13	91.85	93.48	98.37	99.46	100.00
14	97.01	99.38	97.45	99.91	100.00
15	48.13	4.32	76.66	60.52	100.00
16	91.57	77.11	97.80	83.13	100.00
OA	77.02	63.12	85.92	70.45	88.93
AA	68.08	50.57	76.91	56.82	84.59
Kappa	73.49	53.13	83.78	65.43	87.19

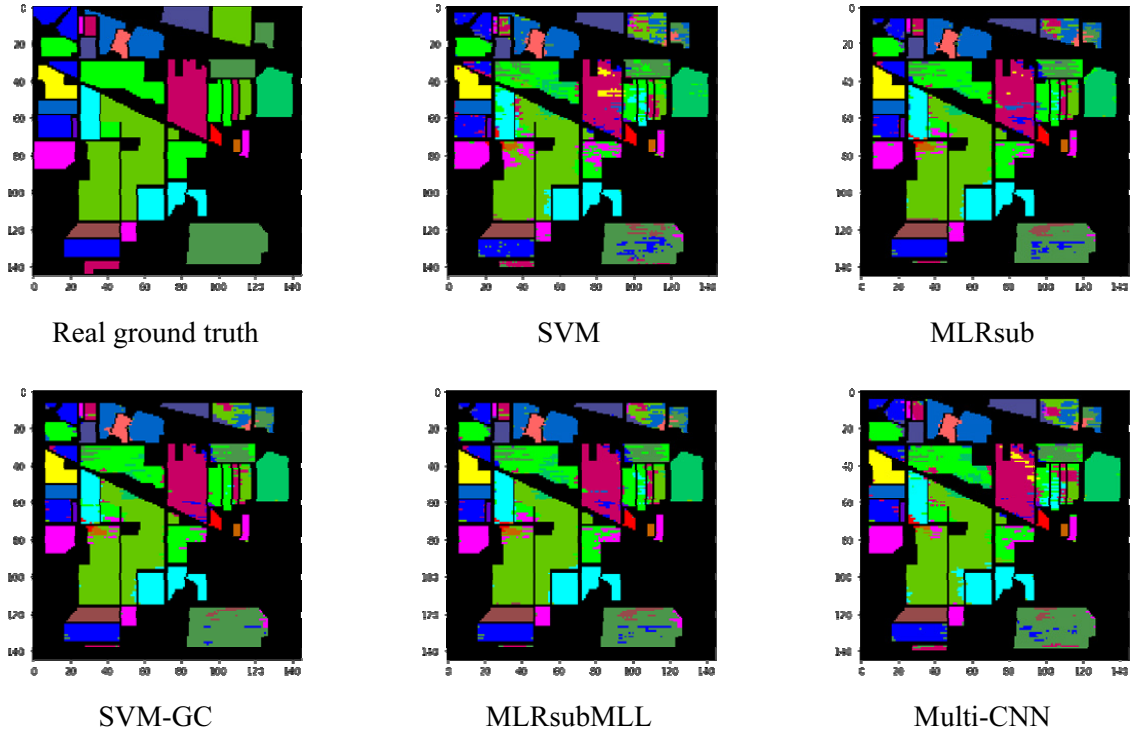

Fig. 11. The represent of the different classification algorithms results

5 Conclusion

In the proposed work, we presented a hyperspectral image classification model with a multi-scale convolutional neural network to address the problem of obtaining enough hyperspectral images with labels for effective training. Compared with SVM (Supported Vector Machine) classifiers and MLR (Multiple Linear Regression) classifiers, by using the Indian pines experimental data set, this method can obtain higher classification accuracy in the case of a small number of training samples.

In this paper, our work is to use multi-scales CNNs for HSI classification exploration, and has a very well performance. In the future, we will consider the comprehensive extraction of spectral features and spatial features of the hyperspectral images. The spectral features and spatial features have been proved to be robust in the case of a small number of training samples for each category. At the same time, we will also use some techniques to alleviate overfitting problems caused by limited training samples. In addition, recent research on deep learning suggests that unsupervised learning can be used to train CNNs. Unsupervised learning does not require a large number of labeled training samples Deep learning, especially deep convolutional neural network, has solid theoretical basis and application value in future HSI classification. We believe that the technology based on the extraction of spectral and spatial features can further improve the classification accuracy of HSI.

Acknowledgments

The work described in this paper is supported by the Natural Science Foundation of China (61672179, 61370083, 61402126), the Specialized Research Foundation for Doctoral Discipline of Higher Education (No. 20122304110012), the Science Foundation for Youths in Heilongjiang Province (QC2016083), the Postdoctoral Foundation in Heilongjiang province (LBH-Z14071), Application Technology Research and Development Projects of Harbin city in 2015 (No. 2015RQQXJ024), the Entrepreneurship Practice for Students in Heilongjiang Province (201611802059), and the Natural Science Foundation of Heilongjiang Province (China) (LH2019A030), the Cultivating Science Foundation of Taizhou University (2019PY014, 2019PY015).

References

[1] R. Gandikota, D. Mishra, How You see me: understanding convolutional neural networks, in: Proc. 2019 IEEE Region 10 Conference (TENCON), 2019.

[2] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39(6)(2017) 1137-1149.

[3] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE Trans. Geosci. Remote Sens. 54(10)(2016) 6232-6251.

[4] S. Roychowdhury, J. Ren, Non-deep CNN for multi-modal image classification and feature learning: an Azure-based model, in: Proc. 2016 IEEE International Conference on Big Data, 2016.

[5] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25(2)(2012) 1-9.

[6] S. Haifeng, C. Guangsheng, W. Hairong, Y. Weiwei, The improved (2D) 2PCA algorithm and its parallel implementation based on image block, Microprocess. Microsyst. 47(2016) 170-177.

[7] B.M. Shahshahani, D.A. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, IEEE Trans. Geosci. Remote Sens. 32(5)(1994) 1087-1095.

[8] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdisciplinary Reviews: Computational Statistics 2(4)(2010) 433-459.

[9] L. De Lathauwer, B. De Moor, J. Vandewalle, An introduction to independent component analysis, J. Chemom. 14(3)(2000) 123-149.

[10] K. Sun, S. Marchand-Maillet, An information geometry of statistical manifold learning, J. Mach. Learn. Res. 32 (2014) 1-9.

[11] P. Brazdil, C. Soares, A comparison of ranking methods for classification algorithm selection, Mach. Learn. ECML 2000(1810)(2000) 63-75.

[12] I. Frades, R. Matthiesen, Overview on techniques in cluster analysis, Methods Mol. Biol. 593(2010) 81-107.

[13] J.H. Arbour, C.M. Brown, Incomplete specimens in geometric morphometric analyses, Methods Ecol. Evol. 5(1)(2014) 16-26.

[14] L. Shen, S. Jia, Three-dimensional gabor wavelets for pixel-based hyperspectral imagery classification, IEEE Trans. Geosci. Remote Sens. 49(12)(2011) 5039-5046.

[15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11)(2018) 2278-2323.

[16] Y. Huang, L. Zhang, P. Li, Y. Zhong, High-resolution hyper-spectral image classification with parts-based feature and morphology profile in urban area, Geo-Spatial Inf. Sci. 13(2)(2010) 111-122.

[17] M. Bronstein, X. Bresson, Y. Lecun, A. Szlam, J. Bruna, Geometric deep learning on graphs and manifolds using mixture model CNNs, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[18] S. Yu, S. Jia, C. Xu, Convolutional neural networks for hyperspectral image classification, Neurocomputing 219(2017) 88-98.

[19] Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7(6)(2014) 2094-2107.

[20] W. Hu, Y. Huang, L. Wei, F. Zhang, H. Li, Deep Convolutional neural networks for hyperspectral image classification, Journal of Sensors 2015(2015) 1-12.

[21] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, J. Paisley, Hyperspectral image classification with markov random fields and a convolutional neural network, IEEE Trans. Image Process. 27(5)(2018) 2354-2367.

[22] Y. Tarabalka, M. Fauvel, J. Chanussot, J.A. Benediktsson, SVM- and MRF-based method for accurate classification of hyperspectral images, IEEE Geosci. Remote Sens. Lett. 7(4)(2010) 736-740.

[23] J. Li, J. Bioucas-Dias, A. Plaza, Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields, Geoscience & Remote Sensing IEEE Transactions on 50(3)(2012) 809-823.