Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(4): 3450–3477.
DOI: 10.3934/mbe.2020195
Received: 23 January 2020
Accepted: 22 April 2020
Published: 02 May 2020

Research article

Using dual-channel CNN to classify hyperspectral image based on spatial-spectral information

Haifeng Song[1], Weiwei Yang[1,*], Songsong Dai[1], Lei Du[1] and Yongchen Sun[2]

[1] School of Electronics and Information Engineering (School of Big Data Science), Taizhou University, Taizhou, China

[2] The Transportation Monitoring and Emergency Response Center of Shandong Province, Jinan, China

* **Correspondence:** Email: yww_1680@163.com; Tel: +86-15957669112.

Abstract: In the field of remote sensing image processing, the classification of hyperspectral image (HSI) is a hot topic. There are two main problems lead to the classification accuracy unsatisfactory. One problem is that the recent research on HSI classification is based on spectral features, the relationship between different pixels has been ignored; the other is that the HSI data does not contain or only contain a small amount of labeled data, so it is impossible to build a well classification model. To solve these problems, a dual-channel CNN model has been proposed to boost its discriminative capability for HSI classification. The proposed dual-channel CNN model has several distinct advantages. Firstly, the model consists of spectral feature extraction channel and spatial feature extraction channel; the 1-D CNN and 3-D CNN are used to extract the spectral and spatial features, respectively. Secondly, the dual-channel CNN have been used for fusing the spatial-spectral features, the fusion feature is input into the classifier, which effectively improves the classification accuracy. Finally, due to considering the spatial and spectral features, the model can effectively solve the problem of lack of training samples. The experiments on benchmark data sets have demonstrated that the proposed dual-channel CNN model considerably outperforms other state-of-the-art method.

Keywords: hyperspecral image; spatial-spectral information; dual-channel; convolutional neural network; classification

1. Introduction

Recent years, a lot of scholars have proposed many methods to extract features from hyperspectral image. These methods can be divided into three categories: spectral domain analysis, spatial domain analysis and spatial-spectral analysis.

Spectral domain analysis refers to use of spectral information in the classification of hyperspectral image [1–3]. There are two kinds of spectral domain analysis. One kind of spectral domain analysis does not reduce the dimensionality of hyperspectral image, hyperspectral image are classified by using the original spectral information directly [4–7]. The other kind of spectral domain analysis method is to first reduce the dimension of hyperspectral image, and then classify the hyperspectral image. The commonly methods used for dimensionality reduction of hyperspectral image include: PCA [2,8], ICA [3] and LDA [1]. However, the disadvantage of these methods is that they only use the spectral features of hyperspectral image, ignoring the relationship between different pixels in hyperspectral image, their classification results may contain noise, like salt-and-pepper [9]. Therefore, the classification accuracy of spectral domain analysis method is not ideal.

Spatial domain analysis refers to use of spatial information in the classification of hyperspectral image. These spatial informations include color, contour and texture. Numerous research results have shown that the spatial features are helpful to improve the representation and classification accuracy of HSI data. In order to extract the spatial features of HSI, it is necessary to define a spatial filter, the common spatial filters include: gray level co-occurrence matrix, wavelet sign, geometric features, texture features and so on. But these spatial features are usually designed for specific data sets, with weak generalization ability, and cannot be widely used. Meanwhile, the variability of spatial features is also very large, which makes it impossible to set classification parameters of HSI by using empirical values. In recent years, deep learning technology has been widely used in hyperspectral image processing. Compared with the traditional artificial design method of spatial feature parameters, deep learning method can automatically extract spatial features, which have strong robustness in classification tasks. Y. Chen et al [9] input spatial features into the automatic coding machine directly, the classification of hyperspectral image had been implemented. However, this method converts the original two-dimensional image data into one-dimensional data when data are input, which causes a great loss of spatial information X. Chen et al [10] proposed a convolutional neural network model, which can be used to extract two-dimensional spatial features and implement the vehicle recognition. However, there are two problems with the above spatial domain analysis methods. First, in hyperspectral image, different objects often have different sizes, so the fixed size detection window can't meet the detection requirements of different size objects. Second, the spatial domain analysis method ignores the spectral features of the original hyperspectral image.

Spatial-spectral analysis methods refers to consider both spectral and spatial information together. Spatial-spectral methods have attracted great interests and improved the HSI classification accuracy significantly [11–17]. Camps-Valls et al. [18] proposed a Composite Kernel (CK) that easily combines spatial and spectral information to enhance the classification accuracy of HSI. Li et al. [19] extended CK to a generalized framework, which exhibits the great flexibility of combining the spectral and spatial information of HSIs. Li et al. [20] proposed the Maximized of the Posterior Marginal by Loopy Belief Propagation (MPM-LBP). It exploits the marginal probability distribution from both the spectral and spatial information. Zhong et al. [21] developed a discriminate tensor spectral-spatial feature extraction method for HSI classification. Kang et al. [22] proposed a spectral-spatial classification framework based on Edge-Preserving Filtering (EPF), where the filtering operation achieves a local optimization of the probabilities. Feng et al. [11] defined discriminate spectral-spatial margins (DSSMs) to reveal the local information of hyperspectral pixels and explore the global structures of both labeled and unlabeled data via low-rank representation. Zhou et al. [23] proposed a spatial and

spectral regularized local discriminant embedding (SSRLDE) method for DR of HSIs. However, most of these extract spectral-spatial features using a shallow architecture and yield limited complexity and non-linearity.

Although the above methods have made some achievements in different areas, but the problem is most of these methods are based on the features for manual design. These methods must be used under the condition of establishing the classification strategy first and these methods design classification strategies directly on data without using classification label information. They are not an end-to-end approach method. Therefore, these methods highly dependent on prior knowledge of specific fields and are usually not the optimal solution [24]. Generally, HSI classification aims at classifying each pixel to its correct class. However, pixels in smooth homogeneous regions usually have high within-class spectral variations. Consequently, it is crucial to exploit the nonlinear characteristics of HSI and to reduce interclass variations. In recent years, the advantages of deep learning in these aspects have gradually emerged, and there have been successful cases of hyperspectral image classification using deep learning. Such as stacked auto encoder [9] and deep belief network [25] in unsupervised feature learning method. Although theses unsupervised learning methods can extract deep features, they need to expand the three-dimensional data into a one-dimensional form to meet the requirements for input data. Therefore, these methods lose the spatial information [26]. The other methods are based on supervised auto-encoder methods [27], which makes use of classification label information in the learning process. These works demonstrate that deep learning opens a new window for future research, showcasing the deep learning-based methods' huge potential. However, how to design a proper deep net is still an open area in the machine learning community [28, 29].

As mentioned above, compared with the traditional spectral domain HSI classification method, the deep learning method can directly learn the data dependency from the original data and make hierarchical representation of the data. Although the above methods of deep learning achieved well results, they did not make full use of spectral information and spatial information for classification. Therefore, it is necessary to synthesize spatial-spectral feature information to further improve the classification accuracy of hyperspectral image. To solve these problems, in this paper a dual-channel CNN model has been proposed to boost its discriminative capability for HSI classification. The proposed dual-channel CNN model has several distinct advantages. Firstly, the model consists of spectral feature extraction channel and spatial feature extraction channel; each channel can extract the spectral and spatial features of the original HSI separately. Secondly, the spectral and spatial features have been fused by using full-connection layer; the fusion feature is input into the classifier, which effectively improves the classification accuracy. Finally, due to considering the spectral and spatial features, the model can effectively solve the problem of lack of training samples. The experiments on benchmark data sets have demonstrated that the proposed dual-channel CNN model considerably outperforms other state-of-the-art method.

An important contribution to the success of the dual-channel CNN to classify hyperspectral image based on spatial-spectral information can be summarized as follows:

(1) A novel end-to-end neural network architecture has been proposed that performs for superior modeling of hyperspectral image. The architecture has fewer independent connection weights and thus requires lesser number of training data. The method is found to outperform the highest reported accuracies on popular hyperspectral image dataset.

(2) Compared with hand-crafted feature extraction, the proposed deep model can adaptively learn

spectral-spatial joint feature, which contains semantic and discriminative information from both spectral and spatial domains.

(3) The design is aimed at efficient spectral-spatial joint feature learning keeping the number of parameters low. So considerable improvement in training time is observed when compared to other popular architectures.

2. Related works

In recent years, the convolutional neural network has made great achievements in the field of computer vision [30]. Many researches have shown that the method based on CNN can significantly improve the accuracy of hyperspectral image classification. For example, LI et al. proposed a feature extractor based on convolutional neural network, which can learn the feature representation of hyperspectral image [31].

2.1. CNN

As shown in Figure 1, the typical convolutional neural network is mainly composed of input layer, convolutional layer, pooling layer, fully connected layer and output layer.

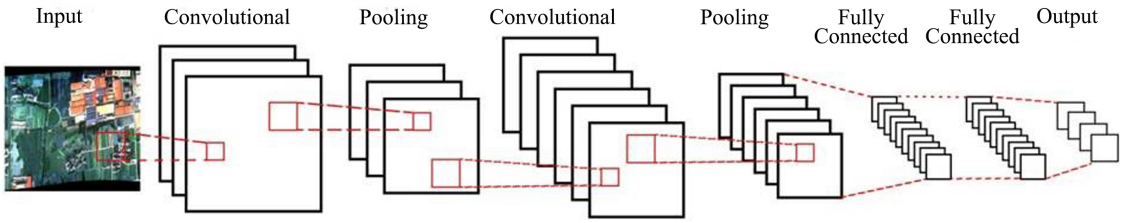

Figure 1. The architecture of convolutional neural network.

Normally, the input of convolutional neural network is the original image X . In this paper, H_i is used to represent the feature map of the *ith* layer of convolutional neural network. Eq. (2.1) is used to calculate the H_i

$$\mathrm{H}_i = f(H_i \otimes W_i + b) \tag{2.1}$$

W_i represents the weight vector of the convolution kernel at the *ith* layer. The symbol $\otimes$ represents the convolutional operation of the *ith* layer and $(i-1)th$ layer with the image or feature map. The output of the convolution is added to the bias b at the *ith* layer. Finally, the feature map W_i of the *ith* layer is obtained through the nonlinear activation function.

Pooling layer is under the convolutional layer, the pooling layer samples the feature map according to certain rules. There are two main rules of pooling layer: (1) Reduce the dimension of the feature map. (2) Keep the scale-invariant properties of the features. Suppose H_i is the pooling layer, H_i can be calculated as Eq. (2.2):

$$\mathrm{H}_i = subsampling(H_{i-1}) \tag{2.2}$$

After completing the calculation of the convolutional layer and pooling layer alternatively. Convolutional neural network classifies extracted features by the values of fully connected network. Obtained the probability distribution Y_{l_i} of the input data (l_i is the *ith* label). As shown in Eq. (2.3), convolutional

neural network is a mathematical model that maps the original matrix (H_0) to a new feature expression Y through multiple layers of data transformation or dimension reduction.

$$\mathrm{Y}_i = P(L = l_i|H_0 : (W, b)) \tag{2.3}$$

The goal of convolutional neural networks for training process is to minimize the loss function $L(W, b)$ of the network. After the input (H_0) passes through the forward conduction, the difference between the predicted value and real value is calculated through the loss function. The typical loss function includes Mean Squared Error, Negative Log, as shown in Eq. (2.4) and Eq. (2.5).

$$\mathrm{MS}\,E(W, b) = \frac{1}{|Y|}\sum_{i=1}^{|Y|}(Y_i - \hat{Y}_i)^2 \tag{2.4}$$

$$\mathrm{NLL}(W, b) = -\sum_{i=1}^{|Y|} logY_i \tag{2.5}$$

In order to alleviate the problem of over-fitting, the final loss function is usually obtained by adding the L2 norm, and λ is the parameter for controlling the strength of over-fitting, as shown in Eq. (2.6).

$$E(W, b) = L(W, b) + \frac{\lambda}{2}W^T W \tag{2.6}$$

In the training process, gradient descent is the common optimization method of convolutional neural network. Loss values are back propagated through gradient descent, the training parameters (W, b) of each layer in convolutional neural network are updated layer by layer. The learning rate η is used to control the intensity of back propagation for Loss value. The updating methods of W and b are shown in Eq. (2.7) and Eq. (2.8)

$$\mathrm{W}_i = W_i - \eta\frac{\partial E(W, b)}{\partial W_i} \tag{2.7}$$

$$\mathrm{b}_i = b_i - \eta\frac{\partial E(W, b)}{\partial b_i} \tag{2.8}$$

Zhao et al. extracted the spatial features which combined with spectral information by using convolutional neural network, and combined with local discrimination embedded for hyperspectral image classification [32]. However, after dimensionality reduction this method only takes the three principal components of the original hyperspectral image as the input, so some information is still lost in the process of spatial feature extraction.

2.2. 3D-CNN

To solve the above problems of convolutional neural network model, Chen et al. extracted spectral-spatial features from the original hyperspectral image by using 3D convolutional neural network, and the results performed better than the aforementioned method on the same data set [33]. Li et al. further researched the 3D convolutional neural network for spatial-spectral joint features by changing the size of the hyperspectral image input cube [31].

The architecture of 3D convolutional neural network (3D-CNN) is similar to that of 2D convolutional neural network (2D-CNN). They are all composed of convolution layer and pooling layer. Unlike

the 2D-CNN, 3D-CNN implements the convolution operation by using 3D convolution kernel, which is one of the key differences between the two kinds of convolution operation. 3D-CNN is shown as Figure 2.

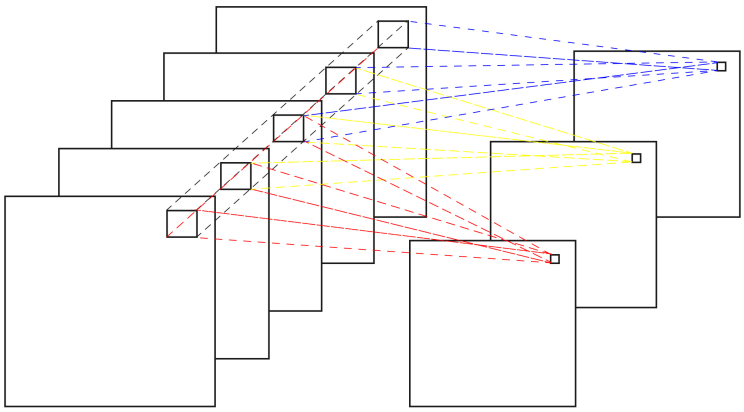

Figure 2. The architecture of 3D convolutional neural network.

The value v at the position (x, y, z) of the jth feature map in the layer l is calculated as Eq. (2.9):

$$v_{lj}^{xyz} = f\left\{b_{ij} + \sum_{m}\sum_{p=0}^{P_l-1}\sum_{q=0}^{Q_l-1}\sum_{r=0}^{R_l-1} w_{ljm}^{pqr} v_{(l-1)m}^{(x+p)(y+q)(z+r)}\right\} \tag{2.9}$$

P_l and Q_l represents the length and width of the three dimensional convolution kernel, R_l is the size of the 3D convolution kernel in spectral dimension, m represents the number of feature maps connected to the current feature graph in the $l-1$ layer, w_{ljm}^{pqr} represents the weight of the mth feature map connected to the $l-1$ layer, $v_{(l-1)m}^{(x+p)(y+q)(z+r)}$ represents the value of the mth feature map at the position $(x+p, y+q, z+r)$ in the $l-1$ layer. b_{ij} is the bias of the jth feature map in the l layer.

Compared with the previous methods, Li et al. And Chen et al. Provided a more concise idea. The model can directly process the original hyperspectral image to obtain feature maps. However, with the expansion of data scale, the classification performance of the model will decrease when the network is deepened.

2.3. Multi-level Convolutional Neural Networks for Scene Understanding

Although CNN and 3D-CNN have gained significant popularity as methods for learning image representations and helped improve the performance of many important computer vision tasks, the transformation of the learned knowledge from the known domain to a new domain such as scene parsing is uncovered yet.

In order to solve the problem, Tam V. Nguyen exploited generic multi-level convolutional neural networks for scene understanding or image parsing task [34]. The input of the proposed model is an image, first, a set of similar images from the training set are retrieved based on global-level CNN feature matching similarities. Then, the input test image and the similar images are oversegmented into superpixels. Next, the class of each test image's superpixel is initialized by the majority vote of the

k-nearest-neighbor superpixels based on regional-level CNN features and hand-crafted features matching. The initial superpixel parsing is later combined with per-exemplar sliding windows to roughly form the pixel labels. Eventually, the final labels are further refined by the contextual smoothing. This is a simple yet effective approach to scene understanding or image parsing that can take advantage of generic convolutional neural network for feature extraction at both image and superpixel levels. Extensive experiments on different challenging datasets demonstrate the multi-level convolutional neural networks can extract the discriminate features which can actually improve the performance significantly.

Inspired by the ideas of this paper, we proposed the dual-channel model to extract the spectral feature and spatial feature by using the 1D and 3D-CNN. It is hoped that this method can improve the classification accuracy of hyperspectral image.

3. Architecture and training of dual-channel CNN

In order to extract features for hyperspectral image, the information in both spectral and spatial domain should be learned jointly. In this section, we proposed a dual-channel deep convolutional neural network for joint spatial-spectral feature learning. Firstly, the spectral and spatial features are extracted, respectively. For the spectral channel, 1-D convolutional neural network is used to extract the spectral features. For the spatial channel, 3-D convolutional neural network is used to extract the spatial features. Then, the spectral-spatial features can be obtained by using the fully connected layers. Finally, the spectral-spatial features are inputted into a classifier, and classification results can be achieved.

3.1. Spatial feature extraction with 3-D CNN

In this section, the HSI spatial feature extraction model based on 3-D convolutional neural network is proposed. The model consists of one input layer, two convolutional layers, two pooling layers, two full connection layers and one output layer. This model can automatically extract the spatial information features of hyperspectral image. The model is shown as Figure 3.

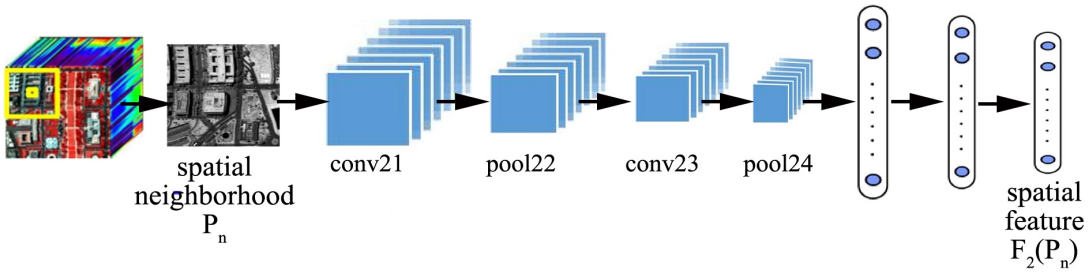

Figure 3. The HSI spatial feature extraction model based on 3-D convolutional neural network.

Assume the hyperspectral image is $H \in R^{h\times w\times d}$, h and w represent the height and width of hyperspectral image, respectively, d is the number of bands, the category of each pixel in hyperspectral image is defined as $K = 1, 2, 3, \cdots, k$, k is the number of categories in hyperspectral image. A sample

set of $p \times p$ size is extracted at the center of each pixel in hyperspectral image H. The data set is represented as $X = x_1, x_2, x_3, \cdots, x_n$, $x_i \in R^{p \times p \times d}$, $n = h \times w$ is the total number of the data set. During the extraction process of the data set, the sample points for the boundary are filled with data 0. Each sample point x_i is input into the convolutional neural network model as input data, the output is z_i, and the corresponding category is $y_i \in K$, y_i is the category of the corresponding vector centered on x_i. The data (x_i, y_i) represents the sample with size $p \times p$ centered on pixel point x_i. After convolutional neural network, the category of input vector is predicted to be y_i. For the *kth* category in the training dataset, there is $D_l = D^{(1)}_{l_{(1)}}, D^{(2)}_{l_{(2)}}, D^{(3)}_{l_{(3)}}, \ldots, D^{(k)}_{l_{(k)}}$, $l = \sum l(k) \ll n$.

Instead of converting the input data into one-dimensional data, the HSI spatial feature extraction model based on 3-D convolutional neural network can directly input the original three-dimensional data into the convolutional neural network model. The size of the input layer data is $p \times p \times d$.

Firstly, the sample centered on x_i with size $p \times p \times d$ is input into the first convolutional layer, the kernel size of the first convolutional layer is $5 \times 5 \times d$, and the number of the kernel is 100. After the first convolutional layer operation, 100 feature maps with size $n_1 \times n_1$ ($n_1 = p - 4$) will be obtained. After the first convolutional layer is the second max pooling layer, the size of the pooling kernel is 2×2. After the max pooling operation, the size of the output feature map is $n_2 \times n_2 \times 100$ ($n_2 = \lceil n_1/2 \rceil$).

Secondly, the feature map will be input to the third convolutional layer, the kernel size of the third convolutional layer is $3 \times 3 \times d$, and the number of the kernel is 300. After the third convolutional layer operation, 300 feature maps with size $n_3 \times n_3$ ($n_3 = n_2 - 2$) will be obtained. After the third convolutional layer is the fourth max pooling layer, the size of the pooling kernel is 2×2. After the max pooling operation, the size of the output feature map is $n_4 \times n_4 \times 300$ ($n_4 = \lceil n_3/2 \rceil$).

Finally, the output feature map of the fourth max pooling layer well be converted to a one-dimensional vector $x_{pool2}(1 \times (n_4 \times n4 \times 300))$. The fifth layer, the sixth layer and the seventh layer is the fully connecter layer. The output of the seventh layer is a one-dimensional vector with the size of $1 \times K$. The fully connected operation formula of the fifth, sixth and seventh layers are shown in Eq. (3.1), Eq. (3.2) and Eq. (3.3):

$$\mathrm{f}^{(5)}(x_{pool2}) = \sigma(W^{(5)} x_{pool2} + b^{(5)}) \tag{3.1}$$

$$\mathrm{f}^{(6)}(x_{pool2}) = \sigma(W^{(6)} f^{(5)}(x_{pool2}) + b^{(6)}) \tag{3.2}$$

$$\mathrm{f}^{(7)}(x_{pool2}) = \sigma(W^{(7)} f^{(6)}(x_{pool2}) + b^{(7)}) \tag{3.3}$$

$W^{(5)}$, $W^{(6)}$ and $W^{(7)}$ is the weight vector, $b^{(5)}$, $b^{(6)}$, $b^{(7)}$ is bias. $\sigma(\cdot)$ is the nonlinear activation function. In this paper, the activation function used in two convolutional layers and three fully connected layers is Tanh. In order to simplify the parameters of the model, suppose $W = W^{(1)}, W^{(3)}, W^{(5)}, W^{(6)}, W^{(7)}$,$b = b^{(1)}, b^{(3)}, b^{(5)}, b^{(6)}, b^{(7)}$. $W^{(i)}$ is the weight of the layer. $b^{(i)}$ is the bias of the *ith* layer. The parameters of the model can be represented with (W, b).

The output of model $f^{(7)} \in R^K$ can be input to the Softmax classifier for the classification of hyperspectral image based on spatial features. $y_i = e^{f^{(7)}_{(i)}} / \sum_{i=1}^{K} e^{f^{(7)}_{(i)}}$, $y^{(W,b)}_{ik} = max(y_i)$. The predicted value of the category of the sample with size $p \times p \times d$ centered on x_i can be obtained. Then, the label y_i and predicted value $y^{(W,b)}_{ik}$ of sample points are taken as input values, the cross entropy is calculated by using Eq. (3.4)

$$\mathrm{E}(W, b) = \frac{1}{l} \sum_{i} \sum_{k=1}^{K} y_{ik} log\, y^{(W,b)}_{ik} \tag{3.4}$$

The parameters W and b are optimized by stochastic gradient descent. After the lth iterations, the calculation methods of W and b are shown in Eq. (3.5) and Eq. (3.6)

$$\mathrm{W}_{t+1} = W_t - \alpha \frac{\partial E(W,b)}{\partial W}\bigg|_{W_t} \tag{3.5}$$

$$\mathrm{b}_{t+1} = b_t - \alpha \frac{\partial E(W,b)}{\partial b}\bigg|_{b_t} \tag{3.6}$$

The back propagation algorithm is used to calculate the gradient of parameters W and b. η is the learning rate.

The HSI spatial feature extraction model based on 3-D convolutional neural network proposed in this paper is different from the traditional convolutional neural network classification model. The traditional convolutional neural networks are mostly based on fine-turning technique. In other words, the convolutional neural network is firstly trained with some prepared samples, and then its parameters are fine-tuned. However, the HSI spatial feature extraction model based on 3-D convolutional neural network proposed in this paper does not require training of prepared samples. The parameter W can be initialized by the standard global distribution; b can be initialized to 0. To prevent overfitting, the dropout is applied after the fifth and sixth full connection layers.

Table 1 shows the parameters of all layers in Figure 3.

Table 1. The parameters of all layers in spatial feature extraction model.

Layers	Parameters
input	$H \in R^{h\times w\times d}$, h=145 w=145 d=220
spatial neighborhood	$P_n = P \times P\times$ d, d=220
conv21	kernel_size:$3\times 3\times 5$,100 output:$n_1 \times n_1 \times 100$, $n_1 = P - 4 = 7$
pool22	kernel_size:$2\times 2\times 5$,stride=2 output:$n_2 \times n_2 \times 100$, $n_2 = \lceil n_1/2 = 4 \rceil$
conv23	kernel_size:$3\times 3\times 5$,300 output:$n_3 \times n_3 \times 300$, $n_3 = \lceil n_2/2 = 2 \rceil$
pool24	kernel_size:$2\times 2\times 5$,stride=2 output:$n_4 \times n_4 \times 300$, $n_4 = \lceil n_3/2 = 1 \rceil$
fc25	200
fc26	84
fc27	42

3.2. Spectral feature extraction with 1-D CNN

In this section, the HSI spatial feature extraction model based on 1-D convolutional neural network is proposed. The 1-D convolutional neural network is used to extract spectral features of hyperspectral images. Replacing the traditional 2-D convolutional kernel with a 1-D convolution kernel can effectively extract the spectral features of hyperspectral image. The model consists of one input layer, three convolutional layers, three pooling layers, two full connection layers and one output layer. This model can automatically extract the spectral information features of hyperspectral image. The model is shown as Figure 4.

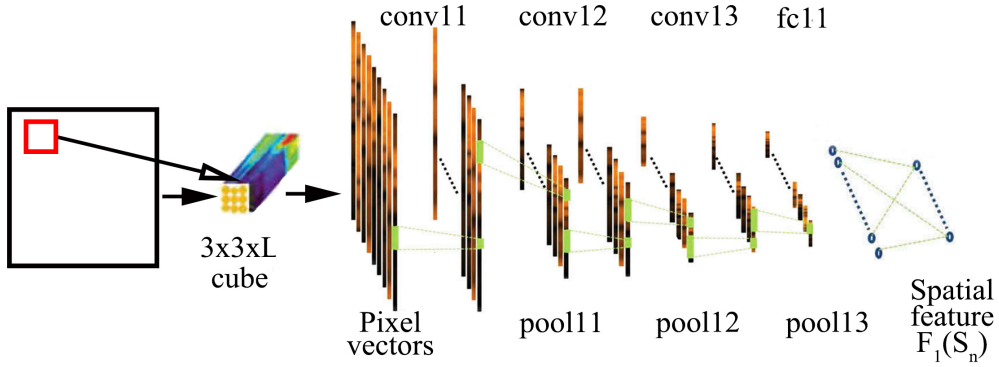

Figure 4. The HSI spectral feature extraction model based on 1-D convolutional neural network.

Table 2 shows the parameters of all layers in Figure 4.

Table 2. The parameters of all layers in spectral feature extraction model.

Layers	Parameters
input	$H \in R^{h\times w\times d}$, h=145 w=145 d=220
3 × 3 × L cube	3 × 3 × 220
pixel vectors	9 × 220
conv11	kernel_size:1 × 1,100 output:9×220,100
pool11	pool_kernel:2×55 stride=2 output:⌈(9-2+1)/2=4,(220-55+1)/2=83⌉,100
conv12	kernel_size:1 × 1,200 output:4×83,200
pool12	pool_kernel:2×40 stride=2 output:⌈(4-2+1)/2=2,(83-40+1)/2=22⌉,200
conv13	kernel_size:1 × 1,300 output:2×22,300
pool13	pool_kernel:2×22 stride=2 output:1 × 1,300
fc11	110
fc12	42

First, the data of its 3 × 3 neighborhood window is collected at a pixel in the original hyperspectral image. L is the band number of the hyperspectral image. Convert the data of $3 \times 3 \times L$ to nine $L \times 1$ 1-D vectors. The value of the *jth* eigenvector of data x in the *lth* layer is shown in Eq. (3.7)

$$v_{l,j}^{x} = f\left(\sum_{m}\sum_{h=0}^{H_l-1} k_{l,j,m}^{h} v_{(l-1),m}^{(x+h)} + b_{l,j}\right) \tag{3.7}$$

l is the number of layer, j is the number of eigenvector, $b_{i,j}$ is the bias of the *ith* eigenvector in the *lth* layer, $f()$ is the activation function, m is the index of the $(l-1)$ layer that connected to current layer, $k_{l,j,m}$ is the *hth* value of the convolution kernel connected to the *mth* eigenvector in the $(l-1)$*th* layer. H_l is the length of the convolutional kernel. In practical applications, we can choose different types of activation functions, such as Sigmoid, ReLU and Tanh. The effect of each activation function will be analyzed through experiments to determine which is the most appropriate activation function.

The pooling layer is usually located after the convolution layer, and the pooling operation can effectively reduce the dimension of the eigenvector. The most commonly used max-pooling operation methods will be adopted in this paper. It is important to note that the input data is a one-dimensional vector, so the convolutional kernel and the pooling kernel are all one-dimensional.

3.3. Spatial-spectral feature extraction with dual-channel CNN

In order to extract the spectral and spatial features of the original hyperspectral image simultaneously. In this section, a HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network is proposed. The model consists of two channels: the first channel is spectral feature extraction channel, and the second channel is spatial feature extraction channel. The architecture of the model is shown as Figure 5.

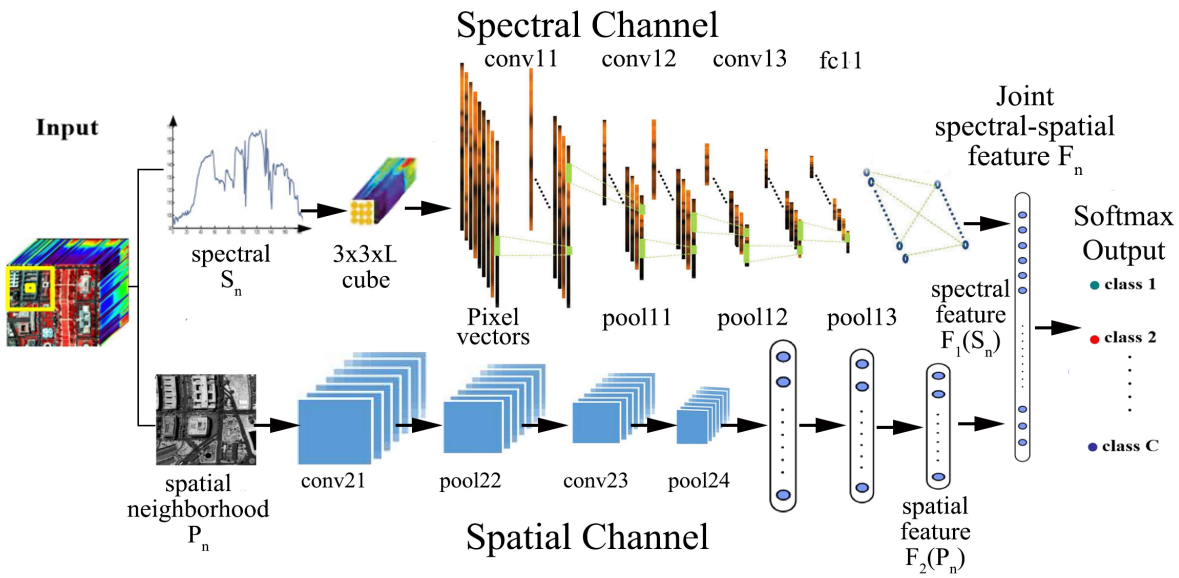

Figure 5. The HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network.

In the spectral feature extraction channel, S_n is used to represent the input data corresponding to the *nth* pixel. After a series of convolution and pooling operations, the output data $F_1(S_n)$ of the spectral feature extraction channel can be obtained. The output data of the channel is the spectral features extracted from the original input data. In the calculation processing of spectral channels, the input data is a $1-D$ vector, so convolution operation and pooling operation are both $1-D$ operation forms.

In the spatial feature extraction channel, P_n is used to represent the $p \times p$ neighborhood window data of the *nth* pixel. This is the input data of spatial feature extraction channel. After a series of convolution and pooling operations, the output data $F_2(P_n)$ of the spatial feature extraction channel can be obtained. The output data of the channel is the spatial features extracted from the original input data. In the calculation processing of spatial channels, the input data is a $3-D$ vector, so convolution operation and pooling operation are both $3-D$ operation forms.

After calculating spectral feature $F_1(S_n)$ and spatial feature $F_2(P_n)$, in order to make comprehensive

use of spectral features and spatial features, feature fusion joint calculation is made for $F_1(S_n)$ and $F_2(P_n)$, as shown in Eq. (3.8):

$$\mathrm{F}_1(S_n) \bullet F_2(P_n) \tag{3.8}$$

• represents the connected operation, this operation corresponds to the method keras.layers.concatenate which was used in the experiment. The input data for this method is a list of concatenated tensors, and the return value is an output tensor concatenated by all the input tensors..

The data after the connected operation is fed to the full connection layer for the operation shown in the Eq. (3.9).

$$\mathrm{F}^{(n)} = f\{W \bullet [F_1(S_n) \bullet F_2(P_n)] + b\} \tag{3.9}$$

W represents the weight vector of the fully connected layer, b represents the bias of the fully connected layer. The output $F^{(n)}$ is calculated by taking spectral and spatial feature as input data, so $F^{(n)}$ can be regarded as the spatial-spectral feature of the *nth* pixel.

Finally, $F^{(n)}$ is input into the Softmax classifier and the probability distribution of the *nth* pixel is calculated, as shown in the Eq. (3.10)

$$Y^{(n)} = \frac{1}{\sum_1^C e^{W_k F^{(n)+b_k}}} \begin{bmatrix} e^{W_1 F^{(n)+b_1}} \\ e^{W_2 F^{(n)+b_2}} \\ \vdots \\ e^{W_c F^{(n)+b_c}} \end{bmatrix} \tag{3.10}$$

C is the number of categories of data to be classified. The maximum value of $Y^{(n)}$ is the corresponding category of the pixel.

It is worth noting that hyperspectral image is inevitably affected by local spatial deformation, shadow, illumination and blur, which greatly affect the classification accuracy. The HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network is proposed in this section which can effectively reduce the influence of local spatial deformation, shadow, light and fuzziness on the classification accuracy rate because of its deep hierarchy architecture.

3.4. The training and optimizing process

The training and optimizing process can be divided into two parts as shown in Figure 6.

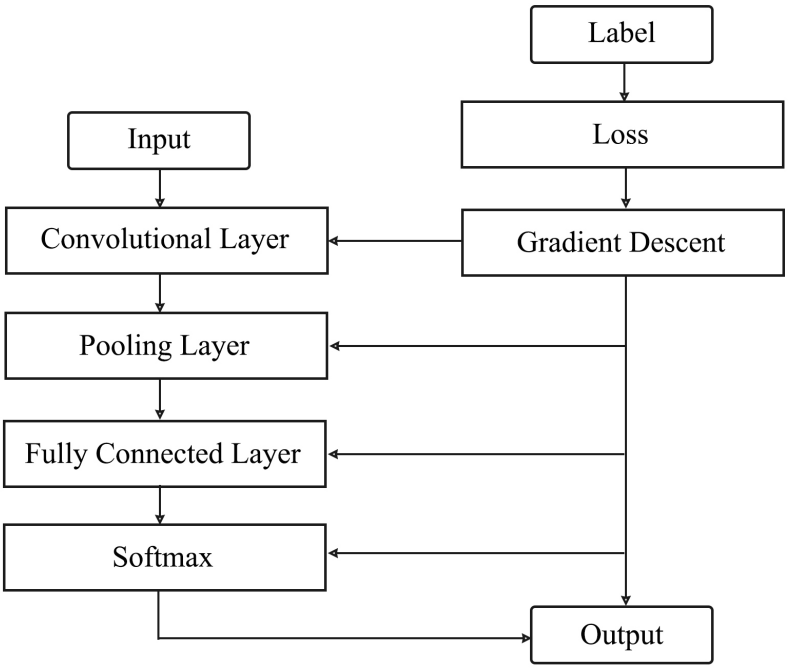

Figure 6. The training and optimizing process of the dual-channel CNN.

The spectral data and spatial data are input into the dual-channel network. After a series of convolution and pooling operations, the data will be input to the fully connected layer. The purpose of the fully connected layer is to map distributed feature representation to the sample label space, and the mapped features can be classified by the Softmax classifier. The predicted value and label are used to calculate the loss value, the gradient descent algorithm and back propagation is used to adjust the network parameters. In the process of training, minimize the loss until the network convergence. The verification process of the network is to cross-verify the trained network model. Parts of the sample data is randomly selected as training data and provided to the network model for identification, calculate the overall accuracy performance of the network by analyzing the performance of the model.

Through training and optimizing process, all parameters in the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network proposed in this paper are learned. The loss function is shown as Eq. (3.11).

$$J(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{C} 1\{k = c^{(n)}\} log P_k^{(n)} \tag{3.11}$$

N is the number of training samples, $c^{(n)}$ is the real category of the *nth* training sample, $p_k^{(n)}$ is the distribution value of the *nth* category corresponding to $p^{(n)}$, $p_k^{(n)}$ is the probability of distribution the *nth* sample to the *kth* category. θ represents the convolutional kernel and the bias. $1\bullet$ is the indicator function, the value is 1 when the parenthesis condition is satisfied, otherwise it is 0.

The random gradient descent algorithm was used to optimize the parameter θ. The parameter θ is initialized with standard deviation of 0.05 and mean value of 0 for random Gaussian distribution. The parameter bias is initialized with 0. The learning rate is initialized with 0.0001. The number of iterations is initialized with 5×10^4.

In order to obtain the model with the best classification accuracy, we divided the experimental dataset into two groups: training set and testing set. The K-fold cross-validation method is adopted in the process of training and testing. As shown in Figure 7, the initial sample is divided into K subsamples, one of the subsamples is retained as testing set for the model, the other k-1 samples were used as the training data set. The cross validation was repeated K times, and each subsample was verified once. The average value of the results of K times was shown in Eq. 3.12. The advantage of this method is that randomly generated subsamples are repeatedly used for training and testing, and the results are verified once each time, it is very useful for the experiment based on one dataset. In my experiment K=10.

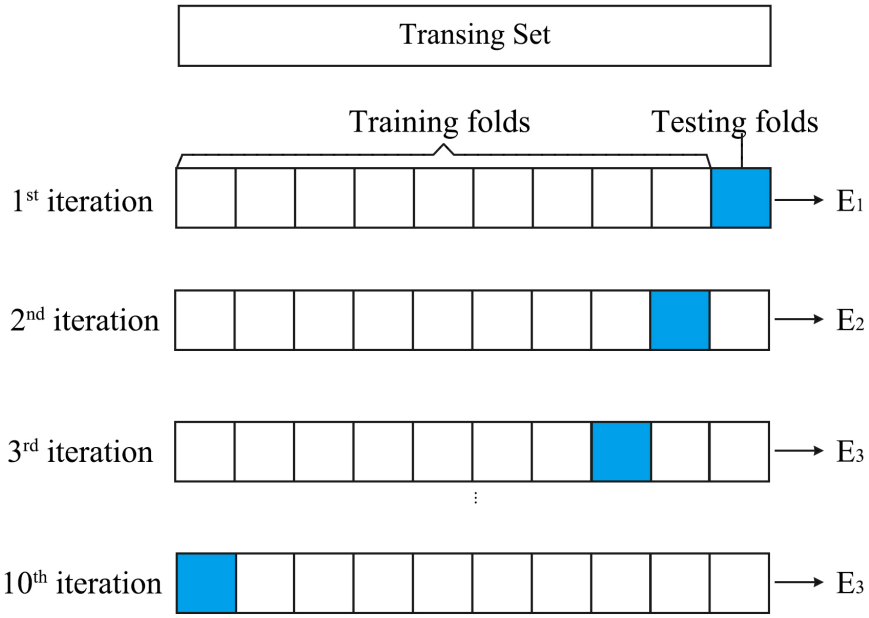

Figure 7. The K-fold cross-validation method.

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \tag{3.12}$$

4. Experimental results and discussions

In this section, the experimental analysis of the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network is conducted. The hardware and software environment used in the experiment is shown in Table 3.

Table 3. The hardware and software environment.

Category	Items	Parameters
Hardware Environment	CPU	Intel E5-1620 v4
	GPU	GTX 1080 Ti
	Memory	32GB
	Hard Disk	1TB
Software Environment	OS	Ubuntu16.04
	IDE Environment	TensorFlow-GPU 1.8.0
	Development Language	Python 3.5

4.1. Description of experimental data sets

The Indiana Pines dataset were collected on 12 June 1992. The collection was taken at Purdue University Farm in Northwest Indiana, USA. The collection equipment is AVIRIS (Airborne Visible Infrared Imaging Spectrometer). Table 4 is a description of the relevant parameters for the dataset.

Table 4. A description of the relevant parameters for Indian Pines.

Num.	Category	Value
1	Band coverage	400–2500nm
2	Spectral channel number	220
3	Spatial resolution	20m
4	Image resolution	145×145

Figure 8 is the gray-scale image corresponding to hyperspectral image, which is composed of band 10.

Figure 8. The gray-scale image corresponding to Indiana Pines.

The ground truth available contains 16 classes and the number of samples in each class distribute unevenly. Table 5 summarizes the categories and image counts for each.

Table 5. The number of samples for each category in Indian Pines.

Num.	Category	Number of samples	Num.	Category	Number of samples
1	Alfalfa	46	9	Oats	20
2	Corn-notill	1428	10	Soybean-notill	972
3	Corn-min	830	11	Soybean-mintill	2455
4	Corn	237	12	Soybean-clean	593
5	Pasture	483	13	Wheat	205
6	Trees	730	14	Woods	1265
7	Pasture-mowed	28	15	Gldg-tree	386
8	Hay-windrowed	478	16	Towers	93
Total				10249	

Figure 9 is the sample distribution of each category.

Figure 9. The sample distribution of each category.

The samples contained in this data set can be divided into four categories: crops, forests, perennials and others. Among them, crops include: Corn-notill[1428], Corn-min[685], Corn[221], Oats[20], Soybean-notill[924], Soybean-mintill[2350], Soybean-clean[561], Wheat[205]; forests include: Trees[730], Woods[1265], Gldg-tree[265]; Forests include: Trees[730], Woods[1265], Gldg-tree[265]; Perennials include: Alfafa[46], Pasture[423], Pasture-mowed[28], Hay-windrowed[478]; Others: Towers[93]. According to the categories of all samples in the data set, crops accounted for 65.77%, woodland accounted for 23.25%, perennial plants accounted for 10.03%, and others accounted for 0.96%.

4.2. Experimental setup for classification of labeled pixels

The framework for all data sets was established as follows. All data sets were randomly divided into the two following groups: a training set, and a testing set. The training sets were used to optimize model parameters. The testing sets were used to test the performance of the model after the training was completed. The batch size was set to 16 and the Adam [29] optimizer was used for stochastic optimization. We used the Xavier normal distribution initialization method [27], also known as the Glorot normal distribution initialization method, for the fully-connected layer. We used a variable learning rate, which was gradually reduced during the optimization process. This was done because the learning rate must be smaller when closer to the valley. The number of training epochs was set to 50000 and the initial learning rate was set to 0.0001. The learning rate was halved when the loss did not decrease after 10 epochs.

The overall accuracy (OA), average accuracy (AA), and the kappa coefficient (K) are adopted to qualitatively evaluate the classification results.

Overall accuracy: refers to the probability that the classified result is consistent with the test data category for each random sample. The overall accuracy is equal to the sum of the pixels that correctly classified divided by the total pixels. The calculation method is shown in Eq. (4.1):

$$OA_i = \frac{C(i,i)}{N_i} \tag{4.1}$$

Average accuracy: refers to the average of classification accuracy of each category. The calculation method is shown in Eq. (4.2):

$$AA = \frac{\sum_{i=1}^{K} OA_i}{K} \tag{4.2}$$

Kappa coefficient is another method to calculate classification accuracy. The Kappa coefficient is between -1 and 1. But usually Kappa coefficient falls between 0 and 1, $Kappa = 1$ indicates complete agreement between the two judgments, $Kappa \geq 0.75$ indicates a satisfactory agreement, $Kappa < 0.4$ indicates less than ideal. It is an ideal index to describe the consistency of diagnosis, so it has been widely used in practical engineering. The calculation method is shown in formula Eq. (4.3):

$$Kappa = \frac{M \sum_{i=1}^{K} C(i,i) - \sum_{i=1}^{K} (C(i,+)C(+,i))}{M^2 - \sum_{i=1}^{K} (C(i,+)C(+,i))} \tag{4.3}$$

In addition to these basic settings, four key factors were used to configure the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network. Namely, (1) The effect of convolutional kernel size; (2) The effect of spatial neighborhood window size; (3) The effect of activation function; (4) The effect of output feature vector dimension on classification results. These four factors are discussed by the OA of IP below.

First, the size of convolution kernel size can affect the OA on classification results. During the experiment, the convolutional kernel size of the first convolutional layer is first fixed, and then the convolutional kernel size of the second convolutional layer is changed to evaluate the effect of the convolutional kernel size on the classification results. Table 6 shows the experimental results. It can be seen from the experimental results that increasing the size of convolution kernel can improve the classification accuracy under certain conditions. However, the accuracy of classification does not

increase linearly with the increment of the convolutional kernel size. In this dataset, with the increment of convolutional kernel size, the accuracy of classification appears to rise first and then fall. The experiments results show that the classification accuracy is the highest when the convolutional kernel size is 3×3. Therefore, the convolution kernel size of the second convolutional layer is set as 3×3. It can also be seen from Table 6 that as the size of convolutional kernel size increases, the computational complexity of the model increases and the classification time increases gradually. Figure 10 shows the curve of classification accuracy during the training process. It can be seen from Figure 10, when the number of iterations is less than 15,000, the classification accuracy increases rapidly with the number of iterations; when the number of iterations is more than 15,000, the classification accuracy increases very slowly with the number of iterations and gradually converges.

Table 6. The effect of convolutional kernel size on classification results.

Kernel Size	1×1	2×2	3×3	4×4	5×5
OA	81.56%	89.25%	89.94%	89.75%	85.25%
Time(s)	5896.1	7419.5	8408.5	9984.8	10178.2

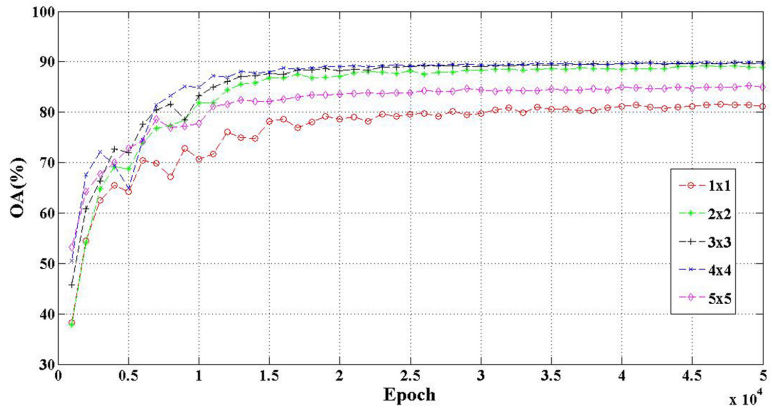

Figure 10. The curve of classification accuracy during the training process.

Second, the window size of spatial neighborhood can effect on classification. This experiment analyzes the effect of convolution kernel size on classification results. During the experiment, the convolutional kernel size of the first convolutional layer is first fixed, and then the convolutional kernel size of the second convolutional layer is changed to evaluate the effect of the convolutional kernel size on the classification results. 5 different neighborhood window sizes of 7×7 pixels, 9×9 pixels, 11×11 pixels, 13×13 pixels and 15×15 pixels were selected to analyze the classification results. Figure 11 is the comparison of classification results.

It can be seen from Figure 11 that the overall classification accuracy does not increase with the increase of spatial neighborhood window size, the overall classification accuracy appears to be increased first and then decreased, reached the best implement at 11×11 pixels. This is because: when the size of the spatial neighborhood window is small, it contains few spatial features that reflect the relationship between adjacent pixel points and cannot describe the spatial features between pixel points very

well, so the overall accuracy is low. When spatial neighborhood window size increases gradually, it contains more and more spatial features that reflect the relationship between adjacent pixels, but it also brings a lot of redundant information or noise data, the redundant information or noise data will affect the classification accuracy, so when the spatial neighborhood window increases to a certain value, the overall classification accuracy declines continue to increase the window size.

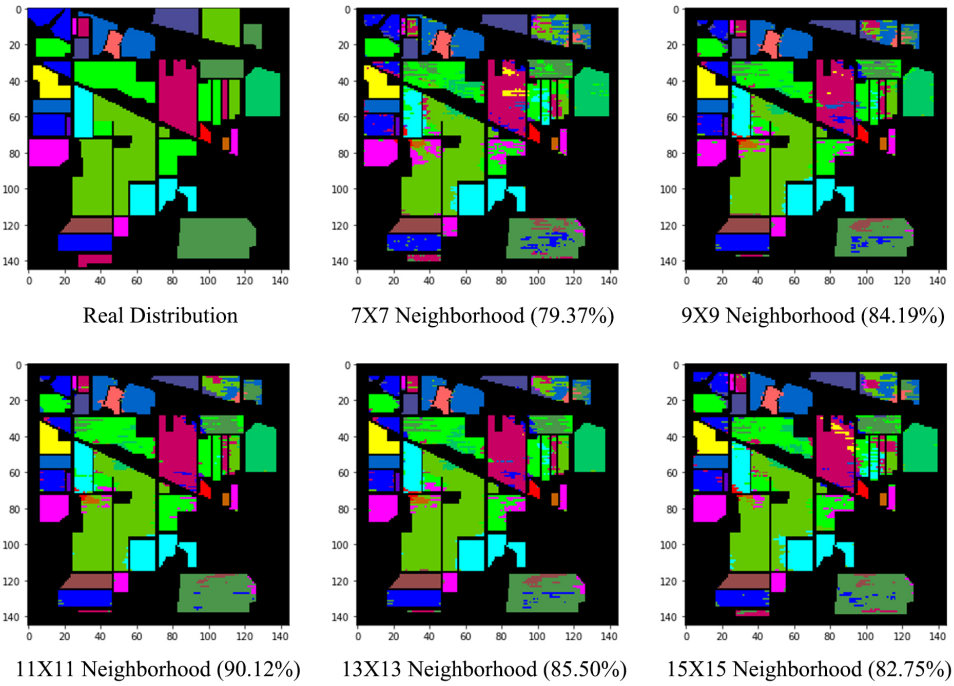

Figure 11. The effect of spatial neighborhood window size on classification results.

Third, the activation function can effect on classification results. During the experiment, all parameters of the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network were fixed, and then the activation functions are set to ReLU, Sigmoid and Tanh respectively. Figure 12 shows the curve of classification accuracy corresponding to the three activation functions. Figure 13 shows the curve of loss function values corresponding to the three activation functions.

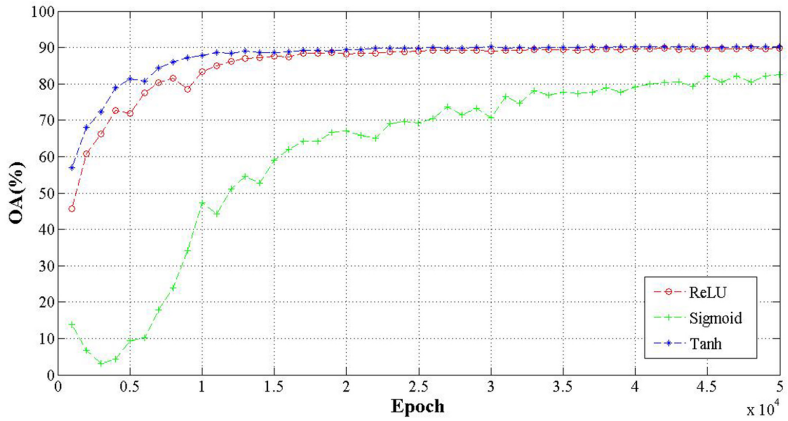

Figure 12. The curve of classification accuracy corresponding to the three activation functions.

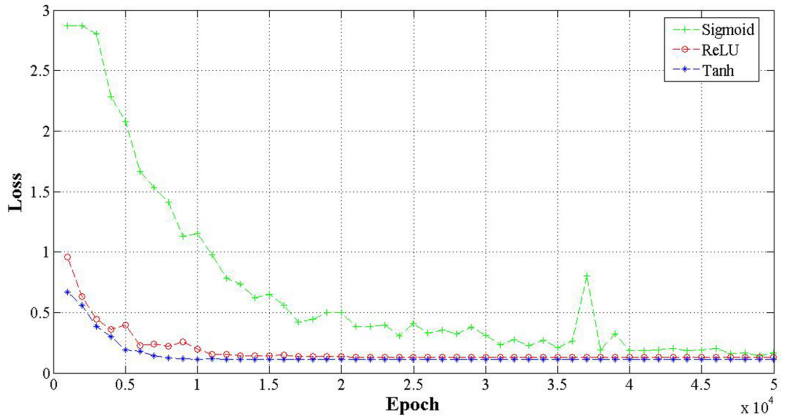

Figure 13. The curve of loss function values corresponding to the three activation functions.

It can be seen from Figure 12 and Figure 13 that with the increase of iterations, the classification accuracy rate corresponding to the three activation functions is gradually increased. However, the classification accuracy corresponding to Sigmoid function is significantly lower than that of ReLU and Tanh. The classification accuracy of Tanh and ReLU is basically the same, but Tanh converges faster. Therefore, Tanh was selected as the activation function of the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network.

Moreover, in convolutional neural networks, the dimensions of the output feature vector in the last layer have a great impact on the accuracy of classification. Therefore, this experiment tests the relationship between the dimensions of output feature vectors and classification results in the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network. The dimensions of the output feature vectors of the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network are set as 50-150 respectively. Then, 50, 100 and 200 samples were randomly selected from the data set as training samples, and the number of test samples was 300. Training the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network and recording the final classification accuracy. The classification results are shown in Figure 14.

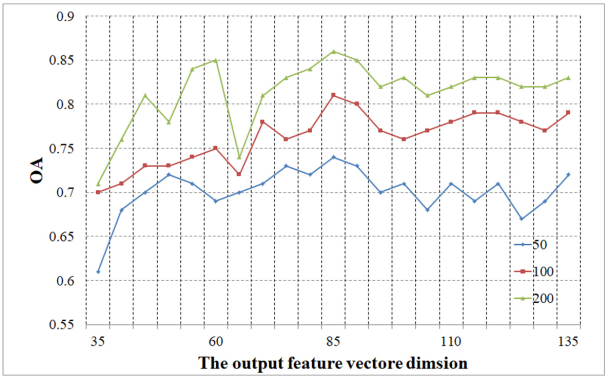

Figure 14. The effect of output feature vector dimension on classification results.

It can be seen from Figure 14 that different dimensions of output vectors have different influences on classification accuracy. With the increase of training samples, the classification accuracy is also improving. For the models trained with training samples of 50, 100 and 200 numbers, the maximum classification accuracy is achieved when the dimensions of the output eigenvector are 80-90. Therefore, the dimension of the output vector dimension is finally selected as 84 in this paper.

During the experiment, the learning rate x was set to 0.0001, and the batch size was set to 16. During the experiment, we found that the influence of other parameters in the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network on the experimental results could be ignored, so we did not conduct further experimental analysis on these parameters.

4.3. Classification results and discussion

The joint representation learning of information from the dual-channels are one of the main contributions of this paper. In this section, we conduct an experiment to show the performance of the proposed dual-channel method compared with single-channel sub-models. In order to verify the classification performance of the proposed model based on dual-channel method. The classification accuracy of the convolutional neural network classification model based on spectral feature extraction, the convolutional neural network classification model based on spatial feature extraction, and the dual-channel convolutional neural network classification model based on spatial-spectral feature extraction were compared and analyzed through experiments.

Table 7 shows the statistical table of classification accuracy OA corresponding to each category of the three models. Figure 15 is the histogram comparing the classification accuracy of the three models for each category.

Table 7. The statistical of classification accuracy OA corresponding to each category of the three models.

Category	Spectral-CNN	Spatial-CNN	Dual-Channel CNN
1	97.83%	97.83%	100.00%
2	67.16%	77.27%	85.71%
3	84.82%	90.37%	83.33%
4	98.64%	100.00%	100%
5	94.09%	97.70%	100%
6	92.19%	95.48%	90.00%
7	100.00%	100.00%	100.00%
8	97.28%	98.60%	50.00%
9	100.00%	100.00%	100.00%
10	80.19%	84.32%	100.00%
11	79.87%	83.75%	100.00%
12	94.65%	95.16%	38.79%
13	100.00%	100.00%	100.00%
14	73.99%	76.08%	100.00%
15	98.49%	100.00%	100.00%
16	100.00%	100.00%	100.00%
OA	82.61%	86.50%	90.12%

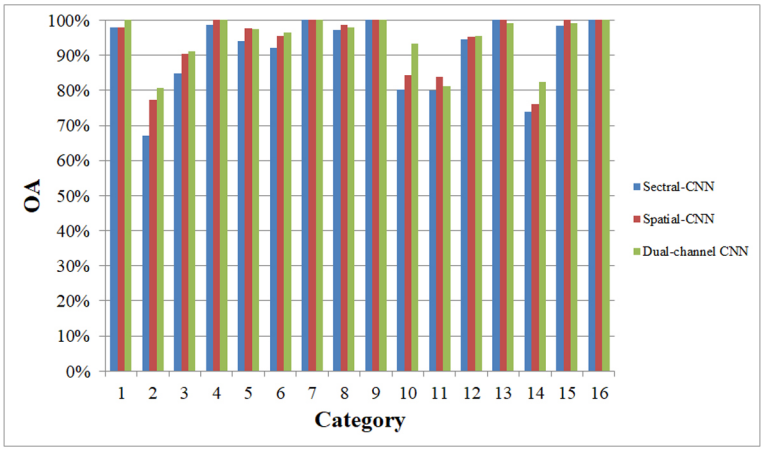

Figure 15. The histogram comparing the classification accuracy of the three models for each category.

As we can see from Table 7 and Figure 15, the classification accuracy of the dual-channel convolutional neural network classification model based on spatial-spectral feature extraction proposed in this paper is significantly higher than that of the other two classification models, with OA reaching 90.12%, the OA of the other two models did not exceed 90.00%. Category 2 has the worst classification result, while category 7, 9 and 16 has the best classification result. The classification accuracy of

category 7, 9 and 16 of the three classification models has reached 100%. Among all the 16 categories to be classified, the three classification models were ranked from low to high in terms of OA is: the convolutional neural network classification model based on spectral feature extraction, the convolutional neural network classification model based on spatial feature extraction, and the Dual-channel convolutional neural network classification model based on spatial-spectral feature extraction. The experiment results show that the proposed method can effectively improve the classification accuracy of hyperspectral images.

The classification results are then compared to some available feature extraction methods, they are SVM [35], MLRsub [35], SVM-GC [35], MLRsubMLL [35]. There also compared with two deep learning based method, stacked AEs basedmethod (J-SAE) [36], 3-D CNN based method (3-D-CNN) [26] for spectral-spatial feature extraction. We demonstrate the results of those feature extraction methods on the experimental datasets. The parameters presented in these contrast methods are respectively set as provided in the corresponding references.

Firstly, Table 8 is the classification results of different methods. From the numerical results, it can be seen from the Table 8, the overall accuracy of different methods is SVM: 77.02%, MLRsub: 63.12%, SVM-GC: 85.92%, MLRsubMLL: 70.45%, J-SAE: 85.09%, 3D-CNN: 86.43% and dual-channel CNN: 90.12%. The classification model proposed in this paper is superior to other classification algorithms in overall classification accuracy, average classification accuracy and Kappa coefficient.

Table 8. The classification results of different methods.

Category	SVM	MLRsub	SVM-GC	MLRsubMLL	J-SAE	3D-CNN	Dual-channel
1	73.17%	46.34%	95.12%	95.12%	97.56%	60.98%	100.00%
2	62.65%	40.93%	68.48%	50.04%	85.68%	78.60%	85.71%
3	52.88%	26.34%	56.49%	13.12%	90.50%	87.42%	83.33%
4	32.39%	17.37%	77.00%	15.02%	68.22%	88.32%	100%
5	91.24%	70.97%	94.47%	73.04%	78.98%	80.60%	100%
6	92.09%	94.37%	97.72%	98.93%	96.11%	92.98%	90.00%
7	36.00%	18.18%	34.42%	37.25%	60.00%	68.00%	100.00%
8	95.58%	96.51%	100%	100%	100%	95.57%	50.00%
9	0%	22.22%	0%	0%	44.44%	77.78%	100.00%
10	61.44%	25.06%	75.06%	19.68%	83.43%	76.91%	100.00%
11	86.92%	78.23%	95.47%	88.82%	95.24%	85.42%	100.00%
12	76.36%	16.51%	99.44%	16.51%	91.17%	82.52%	33.33%
13	91.85%	93.48%	98.37%	99.46%	91.30%	96.20%	100.00%
14	97.01%	99.38%	97.45%	99.91%	97.01%	99.30%	100.00%
15	48.13%	4.32%	76.66%	60.52%	95.98%	89.94%	100.00%
16	91.57%	77.11%	97.80%	83.13%	86.75%	85.54%	100.00%
OA	77.02%	63.12%	85.92%	70.45%	85.09	86.43	90.12%
AA	68.08%	50.57%	76.91%	56.82%	90.89	84.13	85.89%
Kappa	0.7349	0.5313%	0.8378	0.6543	0.8960	0.8450	0.8841

Secondly, Figure 16 is the visualization of the hyperspectral image with different categorys and

Table 9 is the confusion matrix for different cateforys. It can be seen from the Figure 16 and Table 9, the accuracy on different category of the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network is more average. In category 1, 4, 5, 7, 9, 10, 11, 13, 14, 15, 16 the classification accuracy is 100%. The lowest classification accuracy of the 12 category was 38.79%, because the 11, 12 and 13 categories were all different kinds of soybeans, which had similar spectral characteristics.

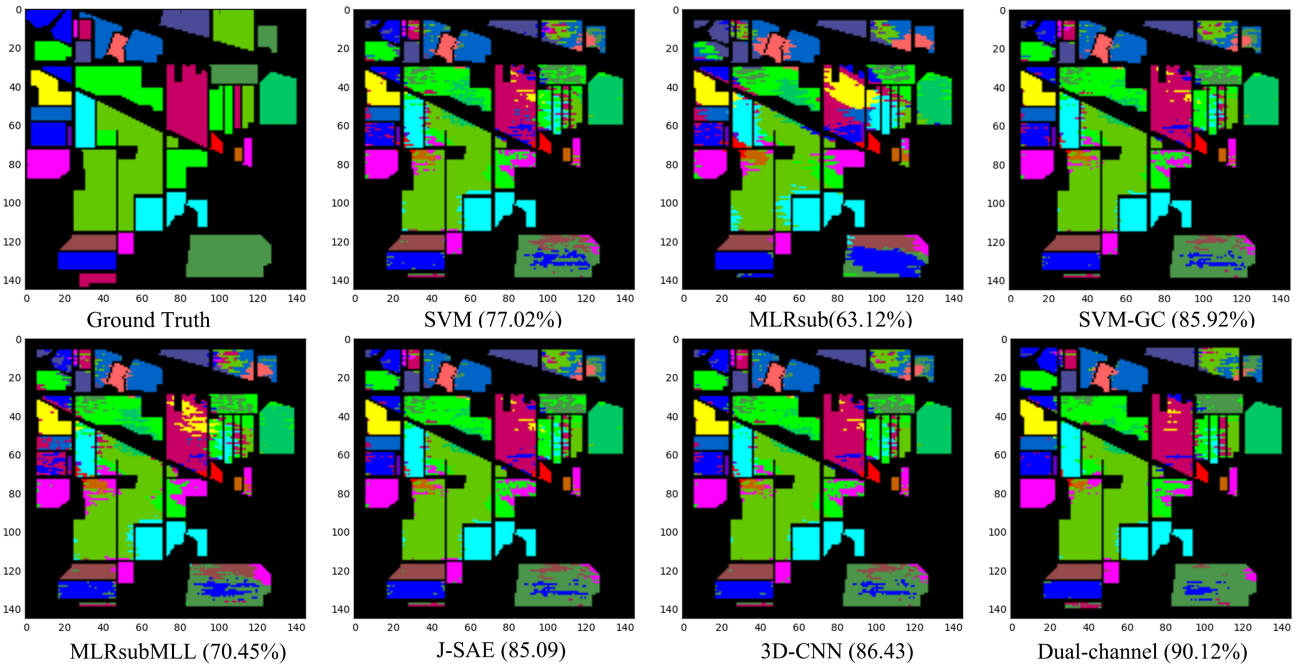

Figure 16. The visualization of the hyperspectral images with different categories.

Table 9. The confusion matrix for different categories.

Category	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Num. of training	OA
1	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	100.00%
2	1	1223	10	0	0	0	0	0	0	42	86	53	0	0	13	0	1428	85.71%
3	0	13	692	0	0	43	0	0	0	0	45	37	0	0	0	0	830	83.33%
4	0	0	0	237	0	0	0	0	0	0	0	0	0	0	0	0	237	100.00%
5	0	0	0	0	483	0	0	0	0	0	0	0	0	0	0	0	483	100.00%
6	0	0	0	0	0	657	0	0	0	45	15	12	0	1	0	0	730	90.00%
7	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	28	100.00%
8	0	0	0	0	45	0	0	239	0	41	39	57	0	0	57	0	478	50.00%
9	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	20	100.00%
10	0	1	0	0	0	0	0	0	0	971	0	0	0	0	0	0	972	100.00%
11	0	0	0	0	0	0	0	0	0	0	2455	0	0	0	0	0	2455	100.00%
12	0	63	87	0	0	30	0	145	0	0	16	236	0	0	12	4	593	38.79%
13	0	0	0	0	0	0	0	0	0	0	0	0	205	0	0	0	205	100.00%
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1265	0	0	1265	100.00%
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	386	0	386	100.00%
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93	93	100.00%

Thirdly, Regarding the computational time, the times required by the different methods are listed in Figure 17. Clearly, J-SAE required the longest time for training. SVM and SVM-GC required a

large number of parameters to achieve its best performance, whereas the accuracy was also not the best.Although MLRsub and MLRsubMLL required the shortest time, its overall accuracy were worst. The 3D-CNN requites about the same time with Dual-channel, but its overall accuracy is lower than the proposed Dual-channel method, the Dual-channel achieved the best overall accuracy.

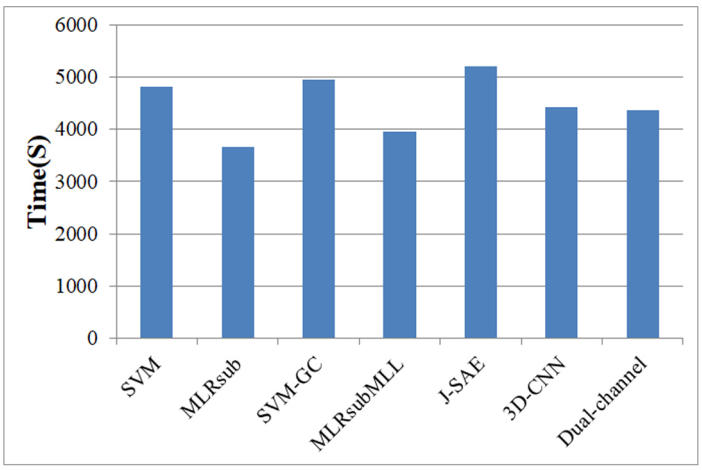

Figure 17. The visualization of the hyperspectral images with different categories.

Finally, the classification accuracy of SVM, SVM-GC and MLRsubMLL in the 9 category is 0%. The main reason is that the number of training samples of this category is few (20), so it is not possible to construct a perfect classification model, resulting in a lower classification accuracy rate. However, the HSI spatial-spectral feature extraction model based on dual-channel convolutional neural network proposed in this paper achieves 100%. In terms of classification accuracy, it is obviously higher than other algorithms, which indicates that dual-channel CNN can effectively extract spectral and spatial features of the original samples and can effectively solve the problem of lack of training samples. So we can conclude that our proposed method gains better classification accuracy than other feature extraction methods.

5. Conclusion

In this paper, we have proposed a novel dual-channel CNN model. It contains two channels of CNN, each of which learns features from spectral and spatial domain, and then a spatial-spectral joint feature is obtained for classification. The model has several distinct advantages. Firstly, the model consists of spectral feature extraction channel and spatial feature extraction channel; the 1-D CNN and 3-D CNN are used to extract the spectral and spatial features, respectively. Secondly, the dual-channel CNN have been used for fusing the spatial-spectral features, the fusion feature is input into the classifier, which effectively improves the classification accuracy. Finally, due to considering the spectral and spatial features, the model can effectively solve the problem of lack of training samples. The proposed method is compared to other well-known classification methods. The experiment results on well-known data sets have shown that the proposed method has better performance in terms of overall classification accuracy, average classification accuracy and Kappa coefficient.

There is still plenty of room to grow in our proposed method, such as more successful strategies in

multi-scale feature fusion and robust classification accuracy to the boundary region. Besides, parallel and distributed fusion strategy, such as [37], will be great in accelerating computation efficiency in practice.

Acknowledgments

The work described in this paper is supported by Natural Science Foundation of China (61672179, 61370083, 61402126), the Natural Science Foundation of Heilongjiang Province (China) (LH2019A030), the Cultivating Science Foundation of Taizhou University(2019PY014, 2019PY015).

Conflict of interest

The authors declare that they have no competing interests.

References

1. T. V. V Bandos, L. Bruzzone, G. Camps-Valls, Classification of hyperspectral images with regularized linear discriminant analysis, *IEEE Trans. Geosci. Remote Sens.*, **47** (2009), 862–873.
2. G. Licciardi, P. R. Marpu, J. Chanussot, J. A. Benediktsson, Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles, *IEEE Geosci. Remote Sens. Lett.*, **9** (2012), 447–451.
3. A. Villa, J. A. Benediktsson, J. Chanussot, C. Jutten, Hyperspectral image classification with Independent component discriminant analysis, *IEEE Transact. Geosci. Remote Sens.*, **49** (2011), 4865–4876.
4. H. Bischof, A. Leonardis, Finding optimal neural networks for land use classification, *NIeee Trans. Geosci. Remote Sens.*, **36** (1998), 337–341.
5. G. Camps-Valls, L. Gomez-Chova, J. Calpe-Maravilla, J. D. Martin-Guerrero, E. Soria-Olivas, L. Alonso-Chorda, et al., Robust support vector method for hyperspectral data classification and knowledge discovery, *IEEE Trans. Geosci. Remote Sens.*, **42** (2004), 1–13.
6. D. L. Civco, Artificial neural networks for land-cover classification and mapping, *Int. J. Geogr. Inf. Syst.*, **7** (1993), 173–186.
7. F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, *IEEE Trans. Geosci. Remote Sens.*, **42** (2004), 1778–1790.
8. 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.
9. 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** (2014), 2094–2107.
10. X. Chen, S. Xiang, C. L. Liu, C. H. Pan, Vehicle detection in satellite images by hybrid deep convolutional neural networks, *IEEE Geosci. Remote Sens. Lett.*, **11** (2014), 1797–1801.
11. Z. Feng, S. Yang, S. Wang, L. Jiao Discriminative spectral-spatial margin-based semisupervised dimensionality reduction of hyperspectral data, *IEEE Trans. Geosci. Remote Sens.*, **12** (2014), 224–228.

12. Z. Wang, N. M. Nasrabadi, T. S. Huang, Spatial-spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization, *IEEE Trans. Geosci. Remote Sens.*, **52** (2014), 4808–4822.

13. S. Bernabe, P. R. Marpu, A. Plaza, M. D. Mura, J. A. Benediktsson, Spectral-spatial classification of multispectral images using kernel feature space representation, *IEEE Geosci. Remote Sens. Lett.*, **11** (2013), 228–292.

14. R. Ji, Y. Gao, R. Hong, Q. Liu, D. Tao, X. Li, Spectral-spatial constraint hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.*, **3** (2014), 1811–1824.

15. M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, J. C. Tilton, Advances in spectral-spatial classification of hyperspectral images, *Proc. IEEE*, **101** (2013), 652–675.

16. W. Zhao, S. Du, Spectral-spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach, *IEEE Trans. Geosci. Remote Sens.*, **54** (2016), 4544–4554.

17. J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, J. Chanussot, Hyperspectral remote sensing data analysis and future challenges, *IEEE Geosci. Remote Sens. Mag.*, **1** (2013), 6–36.

18. G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, J. Calpe-Maravilla, Composite kernels for hyperspectral image classification, *IEEE Geosci. Remote Sens. Lett.*, **3** (2006), 93–97.

19. J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, J. A. Benediktsson, Generalized composite kernel framework for hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.*, **51** (2013), 4816–4829.

20. Jun Li, José M. Bioucas-Dias, Antonio Plaza, Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning, *IEEE Transact. Geosci. Remote Sens.*, **51** (2013), 844–856.

21. Z. Zhong, B. Fan, J. Duan, et al. Discriminant tensor spectral-spatial feature extraction for hyperspectral image classification, *IEEE Geosci. Remote Sens. Lett.*, **12** (2015), 1028–1032.

22. X. Kang, S. Li, J. A. Benediktsson, Spectral-spatial hyperspectral image classification with edge-preserving filtering, *IEEE Transact. Geosci. Remote Sens.*, **52** (2014), 2666–2677.

23. Y. Zhou, J. Peng, C. L. P. Chen, Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.*, **53** (2015), 1082–1095.

24. H. Zhang, Y. Li, Y. Zhang, Q. Shen, Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network, *Remote Sens. Lett.*, **8** (2017), 438–447.

25. Y. Chen, X. Zhao, X. Jia, Spectral-spatial classification of hyperspectral data based on deep belief network, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **8** (2015), 2381–2392.

26. W. Hu, Y. Huang, L. Wei, F. Zhang, H. Li, Deep convolutional neural networks for hyperspectral image classification, *J. Sensor.*, **2015** (2015), 1–12.

27. Z. Lin, Y. Chen, X. Zhao, G. Wang, Spectral-spatial classification of hyperspectral image using autoencoders, *2013 9th International Conference on Information, Communications & Signal Processing*, **Volume** (2014).

28. P. Ghamisi, J. Plaza, Y. Chen, J. Li, A. J. Plaza, Advanced spectral classifiers for hyperspectral images: A review, *IEEE Geosci. Remote Sens. Mag.*, **5** (2017), 8–32.

29. L. Zhang, L. Zhang, B. Du, Deep learning for remote sensing data: A technical tutorial on the state of the art, *IEEE Geosci. Remote Sens. Mag.*, **4** (2016), 22–40.

30. Y. Lecun, Y. Bengio, G. Hinton, ImageNet classification with deep convolutional neural networks, *Nature*, **521** (2015), 436.

31. W. Li, G. Wu, F. Zhang, Q. Du, Hyperspectral image classification using deep pixel-pair features, *IEEE Transact. Geosci. Remote Sens.*, **55** (2017), 844–853.

32. W. Zhao, S. Du, Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach, *IEEE Transact. Geosci. Remote Sens.*, **54** (2016), 4544–4554.

33. Y. Chen, H. Jiang, C. Li, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, *IEEE Transact. Geosci. Remote Sens.*, **54** (2016), 6232–6251.

34. T. V. Nguyen, L. Liu, K. Nguyen, Exploiting generic multi-level convolutional neural networks for scene understanding, *2016 14th International Conference on Control, Automation, Robotics & Vision*, **23** (2016), 1-6.

35. 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** (2018), 2354–2367.

36. Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data, *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, **7** (2014), 2094–2107.

37. Jing Weipeng, Huo Shuaiqi, Miao Qiucheng, Chen Xuebin, A Model of Parallel Mosaicking for Massive Remote Sensing Images Based on Spark, *IEEE Access*, **99** (2017), 18229–18237.

AIMS Press