

Convolutional neural networks and local binary patterns for hyperspectral image classification

Xiangpo Wei, Xuchu Yu, Bing Liu & Lu Zhi

To cite this article: Xiangpo Wei, Xuchu Yu, Bing Liu & Lu Zhi (2019) Convolutional neural networks and local binary patterns for hyperspectral image classification, European Journal of Remote Sensing, 52:1, 448-462, DOI: [10.1080/22797254.2019.1634980](https://doi.org/10.1080/22797254.2019.1634980)

To link to this article: <https://doi.org/10.1080/22797254.2019.1634980>



© 2019 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 28 Jun 2019.



Submit your article to this journal [↗](#)



Article views: 3779



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 12 View citing articles [↗](#)

Convolutional neural networks and local binary patterns for hyperspectral image classification

Xiangpo Wei, Xuchu Yu, Bing Liu and Lu Zhi

Information Engineering University, Zhengzhou, China

ABSTRACT

Convolutional neural networks (CNNs) have strong feature extraction capability, which have been used to extract features from the hyperspectral image. Local binary pattern (LBP) is a simple but powerful descriptor for spatial features, which can lessen the workload of CNNs and improve the classification accuracy. In order to make full use of the feature extraction capability of CNNs and the discrimination of LBP features, a novel classification method combining dual-channel CNNs and LBP is proposed. Specifically, a one-dimensional CNN (1D-CNN) is adopted to process original hyperspectral data to extract hierarchical spectral features and another same 1D-CNN is applied to process LBP features to further extract spatial features. Then, the concatenation of two fully connected layers from the two CNNs, which fused features, is fed into a softmax classifier to complete the classification. The experimental results demonstrate that the proposed method can provide 98.52%, 99.54% and 99.54% classification accuracy on the Indian Pines, University of Pavia and Salinas data, respectively. And the proposed method can also obtain good performance even with limited training samples.

ARTICLE HISTORY

Received 16 August 2018
Revised 2 June 2019
Accepted 19 June 2019

KEYWORDS

Hyperspectral image; classification; convolutional neural networks; deep learning; local binary patterns

Introduction

Hyperspectral image with very high spectral resolution can be used to obtain the approximately continuous spectral curve and detect the diagnostic spectral characteristics of ground materials. Thus, ground materials can be more discriminative with detailed spectral information (Ghamisi, Plaza, Chen, Li, & Plaza, 2017). Hyperspectral image classification has been developed for a variety of applications (Bioucas-Dias et al., 2013), such as environmental monitoring (Tegdan et al., 2015) and precision agriculture (Lacar, Lewis, & Grierson, 2001). However, due to the complexity of spectral and spatial structures, high dimensionality and strong correlation between adjacent bands, the classification of the hyperspectral image still remains a challenging task (Gomez-Chova, Tuia, Moser, & Camps-Valls, 2015).

Therefore, extracting discriminative features from complex image data has become a hot topic of hyperspectral image classification. Conventional hyperspectral image classifier solely considers spectral signatures and linear transformation-based methods are utilized to extract potentially better features from spectral domain (Liu et al., 2018b), such as linear discriminant analysis (LDA) (Bandos, Bruzzone, & Camps-Valls, 2009), principal component analysis (PCA) (Licciardi, Marpu, Chanussot, & Benediktsson, 2012) and independent component analysis (ICA) (Villa, Benediktsson, Chanussot, &

Jutten, 2011). Linear transformation-based feature extraction methods usually try to find a transformation matrix to maximize the discriminative power and separate the available classes effectively. Nevertheless, due to the nonlinearity of ground scattering, hyperspectral images are inherently nonlinear (Chen, Jiang, Li, Jia, & Ghamisi, 2016; Du, Wang, Tan, & Xia, 2011), which is not suitable for these linear feature extraction methods. Some nonlinear processing strategies such as kernel-based method (Camps-Valls & Bruzzone, 2005) have been introduced to feature extraction of hyperspectral image and obtained better performance (Li, Chen, et al., 2015).

Due to the phenomenon of the same object with different spectrums and the different objects with the same spectrum in the hyperspectral image, it is difficult to classify different classes precisely using spectral information alone (He, Li, Liu, & Li, 2017). However, spatially adjacent pixels usually share similar characteristics and labels. Recent studies have suggested incorporating spatial information into the feature extraction process (Plaza, Plaza, & Martín, 2009). And jointly combining spatial and spectral information can provide a significant advantage in terms of reducing the uncertainty of the samples. An approach to include spatial information in classification consists of morphological profiles (MPs) to model the spatial information (Miao et al., 2002). MPs applied a set of

mathematical morphological operations on spectral images. Some MP-based methods were also proposed, such as the derivative of MP (DMP) (Miao et al., 2002), extended morphological profiles (EMPs) (Fauvel, Benediktsson, Chanussot, & Sveinsson, 2008), attribute profiles (APs) (Dalla Mura, Benediktsson, Waske, & Bruzzone, 2010), etc. Another set of spectral-spatial classification approaches relies on the concept of composite kernels (CKs) (Li, Marpu, Plaza, Bioucas-Dias, & Benediktsson, 2013). CKs are formed by the spatial and spectral kernels computed by extracted spatial features and original spectral information. In addition, Markov random field (MRF) regularization is also used for refining the classification results by integrating spatial-contextual information (Moser & Serpico, 2013). Nevertheless, MP-based features are hand-crafted and less flexible. Spatial features extracted by CKs are too simple to accurately represent complex spatial structures. And MRF-based methods heavily depend on the initial pixelwise classification (Lei, Mcisaac, & Osinski, 2018).

As a simple but powerful texture descriptor, local binary pattern (LBP) has been successfully applied to texture representation, such as face recognition (Zhao & Pietikäinen, 2007) and texture classification (Guo, Wang, Zhou, & You, 2016). Recently, the LBP model has also been introduced for spatial-domain feature extraction and classification of hyperspectral image. In this context, LBP and a global Gabor filter are employed in a subset of bands selected from hyperspectral image to produce a comprehensive description of spatial texture information (Li, Chen, Su, & Du, 2015). These extracted features are then concatenated with the spectral features together to perform the classification task. Similarly, Yang, Gao, Dong, and Yang (2018) stacked the spectral features and LBP features extracted from selected bands into high dimensional vectors. Then, the stacked features of a specified position are transformed into a 2D image which are fed into PCANet for classification. For extracting spectral and spatial features simultaneously, the conventional 2D-LBP model is extended to 3D domains which jointly considers the spectral and spatial information in a single framework (Jia, Hu, Zhu, Jia, & Li, 2017b). On the other hand, post-processing strategies can also be combined with LBP to refine the classification. LBP-based feature extraction for the class-conditional probability can be coupled with an MRF-based prior (Ye, Fowler, & Bai, 2017). Instead of relegating spatial information solely to the class prior, an LBP feature extraction employing both sign and magnitude components provides a class-conditional probability that comprised both spatial and spectral information. In addition, the Uniform local binary pattern (ULBP) descriptor is combined with superpixel-level decision

fusion strategy together for hyperspectral image classification (Jia, Deng, Zhu, Jia, & Li, 2017a).

More recently, deep learning has been proved to be a more preferable way to extract nonlinear high-level features because of its hierarchical learning framework (Zhao & Du, 2016). And many deep learning-based methods have been used in hyperspectral image classification and yield excellent performance. Stacked autoencoder (SAE) (Chen, Lin, Zhao, Wang, & Gu, 2014; Ma, Wang, & Geng, 2016) and deep belief network (DBN) (Chen, Zhao, & Jia, 2015) are firstly introduced to deal with feature extraction and classification in spatial and spectral domain. And a series of advanced hyperspectral classification methods is proposed to obtain better performance. Then, a deep CNN is also used to classify a hyperspectral image in the spectral domain (Hu, Huang, Wei, Zhang, & Li, 2015; Tao, Pan, Li, & Zou, 2015). Furthermore, CNN-based methods are more suitable for extracting nonlinear spectral-spatial features for hyperspectral image classification by means of the nonlinear activation function (Yue, Zhao, Mao, & Liu, 2015; Liu et al., 2018a; Romero, Gatta, & Camps-Valls, 2016). For instance, Zhang, Li, Zhang, & Shen (2017) adopt one-dimensional CNN (1D-CNN) and two-dimensional CNN (2D-CNN) to extract the hierarchical spectral features and the hierarchical spatial features, respectively, i.e., called dual-channel CNN (DC-CNN). The extracted spectral features and spatial features are thereby fused to complete the hyperspectral image classification. This concatenation strategy can extract features in a high-efficiency manner and achieve decision-level feature fusion.

Due to the discrimination capability of LBP features, the combination of LBP features and the CNNs can lessen the workload of CNNs (Chen et al., 2017). To make the most of the spectral-spatial feature extraction ability of DC-CNN and the discrimination capability of LBP features, a novel classification method combining advanced DC-CNN and LBP features, called LBP Dual-Channel CNN (LBP-DC-CNN), is proposed in this paper. Specifically, original hyperspectral data and LBP features are processed in an advanced DC-CNN framework. Original hyperspectral data is fed into a 1D-CNN model to extract hierarchical spectral features. On the other hand, LBP features are fed into another same 1D-CNN model which is another channel of DC-CNN framework to further extract hierarchical spatial features simultaneously. Next, the fully connected layers of the two 1D-CNN models in DC-CNN framework are concatenated into a fused layer, which complete the fusion of spectral features and spatial features. Finally, the fused features are fed into a softmax layer to conduct classification.

By processing the original hyperspectral data and LBP features, respectively, using two 1D-CNN models, LBP-DC-CNN not only makes full use of

the feature extraction capability of the CNNs and the advantage of LBP features but also implements the effective fusion of various characteristics. The main contributions of this paper can be summarized as follows.

- (1) Features extracted by LBP are taken as the input of 1D-CNN model to further extract spatial features, which can help complete the fusion with spectral features extracted from another 1D-CNN model.
- (2) An advanced dual-channel CNN framework with LBP is proposed. Specifically, a 1D-CNN model is adopted to process original hyperspectral data to extract hierarchical spectral features and then another same 1D-CNN model is applied to process LBP features to extract hierarchical spatial features. Then, the two fully connected layers of the two CNNs are concatenated to fuse spectral features and spatial features. Finally, the fused features are fed into the softmax layer to conduct classification.
- (3) Experiments using three well-known datasets verify the better performance of LBP-DC-CNN compared with conventional classification methods.

The remainder of this paper is organized as follows. In Section 2, the proposed classification method is described in detail. In Section 3, the experimental results and the corresponding analysis are presented. Conclusions are presented in Section 4.

Proposed method

LBP

The LBP is a powerful nonparametric operator for the description of local image features and has been proved to be rotation invariant and translation invariant. Given a center pixel (x_c, y_c) , an ordered binary set defined as LBP is obtained by comparing the gray

value of the center pixel (x_c, y_c) with the pixels of its eight neighbors. Thus, the LBP code is expressed as the decimalized form of an octet binary number:

$$\text{LBP}(x_c, y_c) = \sum_{n=0}^7 S(i_n - i_c) 2^n \quad (1)$$

where i_c represents the gray value of the center pixel (x_c, y_c) , and i_n is the gray value of the pixels of its eight neighbors. LBP code has been proved to be invariant to any monotonous gray level transformation, and the local neighborhood binary code remains unchanged after transformation.

$$S(i_n - i_c) = \begin{cases} 1 & i_n - i_c \geq 0 \\ 0 & i_n - i_c < 0 \end{cases} \quad (2)$$

Figure 1 illustrates an example of LBP calculation with a thresholding process of eight neighbors. Figure 1(a) shows a 3×3 window; difference values are calculated between the gray value of the center pixel and the gray values of neighboring pixels. As shown in Equation (2), the transformation function S resets the difference value to “1” when the difference value is greater than 0, and “0” otherwise. Thus, the binary value set in Figure 1(b) can be obtained with the corresponding binary code $(\text{LBP})_2 = 10,011,100$. Figure 1(c) shows the code set of binary codes, that is, a weighted coefficient set of binary codes. Then, the corresponding code $(\text{LBP}) = 156$.

LBP code only depends on the difference values between the center pixel and neighboring pixels, which possesses 256 patterns to reflect various texture information. LBP code maps the gray value of any pixel in the image to LBP code, which can be used to obtain an LBP histogram of an image through histogram statistics to describe the texture features of the image.

CNNs

A typical CNN is a type of deep model in which convolutional filters and pooling operations are applied alternately on the local neighborhoods of each pixel in

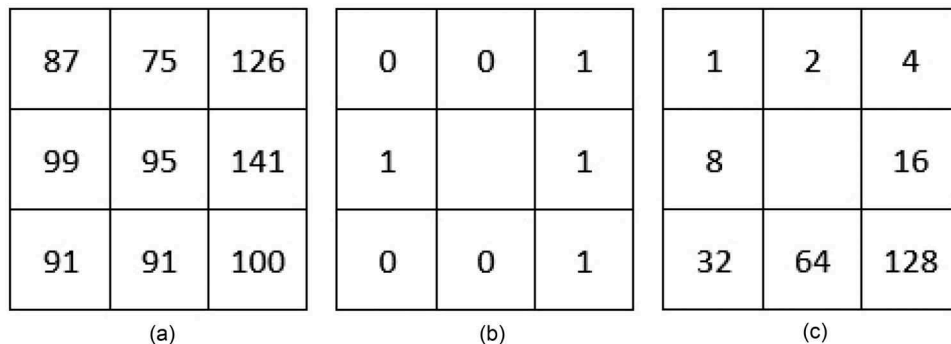


Figure 1. Example of LBP calculation. (a) Gray-scale map, (b) binary value set, and (c) code set.

the raw input, generating complex high-level features (Li et al., 2017). CNNs have been primarily applied to 2D images, and have achieved better performance in image classification.

A 1D-CNN is a type of simplified CNN used to process a 1D input. A 1D-CNN is primarily applied to pixelwise HSI classification that only utilizes a pixel spectral vector as the input (Hu et al., 2015). Unlike the 2D-CNN, the scale of the convolution kernel and pooling operation is usually 1D in 1D-CNN. Formally, a 1D convolution operation is expressed as

$$v_{l,j}^x = f\left(\sum_m \sum_{h=0}^{H_l-1} k_{l,j,m}^h v_{(l-1),m}^{(x+h)} + b_{l,j}\right) \quad (3)$$

where $v_{l,j}^x$ is the value of a unit at position x in the j 'th feature map in the l 'th layer; $f(\cdot)$ is the activation function; m indicates the indexes over the set of feature maps in the $(l-1)$ 'th layer connected to the current feature map; H_l represents the length of the convolutional filter; $k_{l,j,m}^h$ represents the value at position h of the filter connected to the m 'th map; and $b_{l,j}$ is the additive bias. Figure 2 shows the 1D convolution operation with a stride of 1.

The pooling layer (e.g., mean pooling and max pooling) is usually implemented following the convolutional layer to reduce the size of the feature maps and decrease the quantity of parameters. Features become more abstract and discriminative with various convolutions and poolings. In addition, a fully connected layer maps the learned feature representation to the classifier.

LBP-DC-CNN

The proposed classification method illustrated in Figure 3, i.e., LBP-DC-CNN, mainly contains three parts. Firstly, using a 1D-CNN model which is the first channel of LBP-DC-CNN to process original hyperspectral data to extract hierarchical spectral features. Secondly, using another same 1D-CNN model which is another channel of LBP-DC-CNN to process LBP features to further extract hierarchical spatial features simultaneously and concatenating two fully connected layers from the two aforementioned 1D-CNN models to fuse spectral and spatial features for yielding classification finally. In the two channels of LBP-DC-CNN, 1D-CNN model extracts spatial features and spectral features, respectively. Specifically, the 1D-CNN model includes two 1D convolutional layers and one fully connected layer (the tuning of the network structure will be discussed in Sec. 3.2), which is proved to be effective in feature extraction. In the later stage of LBP-DC-CNN, the concatenation of two fully connected layers from the two 1D-CNN models is fed into a softmax layer to conduct classification. The style of concatenation for two fully connected layers will preserve the values of original features and improve the discrimination capability of the fused features. The rectified linear unit (ReLU) is adopted as the nonlinear activation function of the convolutional layers owing to its faster convergence speed (Hara, Saito, & Shouno, 2015). ReLU can solve the problem of gradient vanish which is likely to occur in conventional activation function, e.g., Sigmoid and Tanh. Furthermore, the proposed method is optimized by Adam, which is

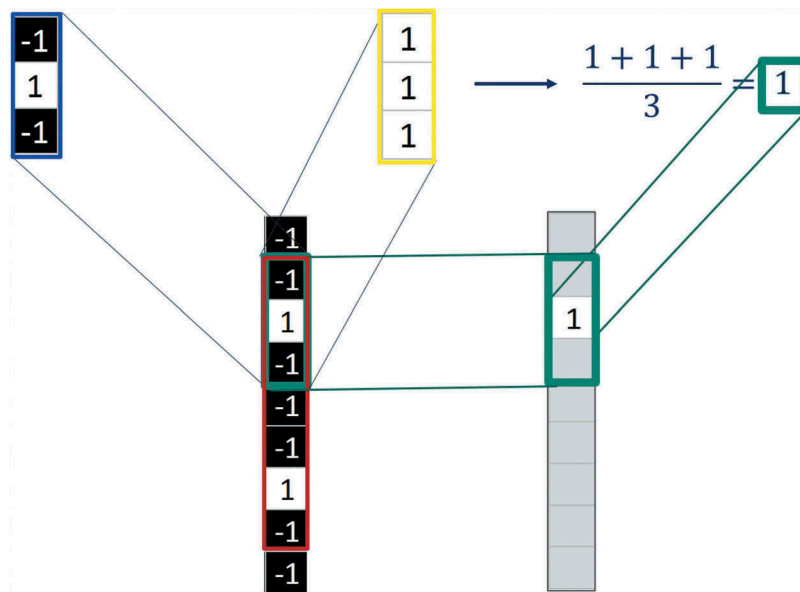


Figure 2. Illustration of 1D convolution with a stride of 1. The red line area of the black image indicates the original image, and the cyan area of the black image indicates the area to convolve. The blue line image represents the convolution kernel, and the yellow line image represents the convolved results. The figure in the cyan area of the gray image indicates the final value.

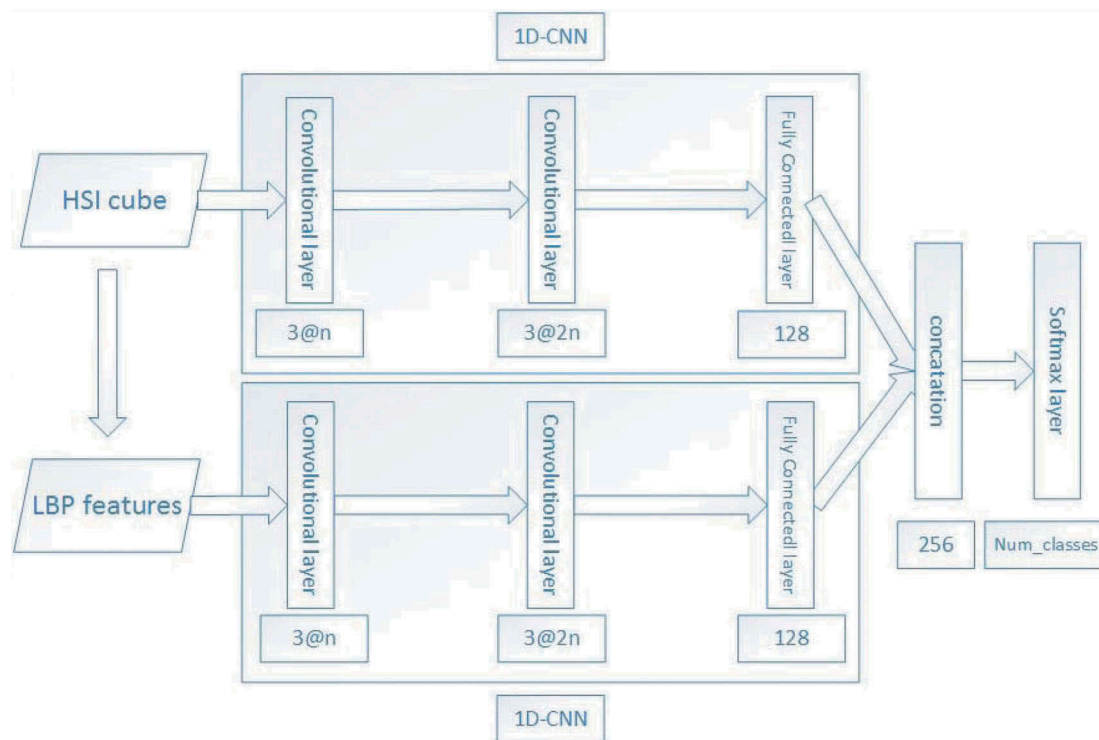


Figure 3. Flowchart of LBP-DC-CNN.

suitable for non-convex optimization and high-dimensional space (Kingma & Ba, 2014). In the output layer, the softmax function is employed as a classifier owing to its simple computation and good performance in classification. In convolutional layer, small fields of 3×3 can yield better results (Simonyan & Zisserman, 2014), and we confirmed the size of the 1D convolution kernel to be 3. The first convolutional layer adopts sixteen 3 filters, and the second convolutional layer adopts thirty-two 3 filters. And the length of the fully connected layer is 128.

Results

Experimental data

The proposed classification method LBP-DC-CNN is implemented using the Python programming language and Keras library (<https://keras.io/>). The results are generated on a PC equipped with an Intel 2.6 GHz Core i7-4720HQ and 16GB of memory. Three hyperspectral datasets, including Indian Pines, University of Pavia and Salinas data (Computational Intelligence Group of the University of the Basque Country), are employed to evaluate the performance of the proposed classification method.

- (1) Indian Pines data is obtained by an Airborne visible infrared imaging spectrometer (AVIRIS) sensor with 200 spectral channels after removing water absorption bands and a 0.4–2.45 μm region of wavelength. The image represents

a scene with 145×145 pixels, where 16 land-cover classes are labeled in the ground truth map. Nevertheless, only nine classes are considered because a few classes have very few training samples (Li, Wu, Zhang, & Du, 2016). The number of training and testing samples are listed in Table 1. The false-color image (band 35, band 165 and band 148) and the ground truth map are shown in Figure 4.

- (2) The University of Pavia data is collected by a Reflective optical system imaging spectrometer (ROSIS) sensor with 103 spectral channels after removal of water bands and a 0.43–0.86 μm region of wavelength. The image represents a scene with 610×340 pixels, where 42776 pixels with nine classes are labeled in the ground truth map. The number of training and testing samples are listed in Table 2. The false-color image (band 32, band 59 and band 92) and the ground truth map are shown in Figure 5.

Table 1. Number of training and testing samples used in the Indian Pines data.

Class no.	Class name	Training samples	Testing samples
1	Corn-notill	200	1228
2	Corn-mintill	200	630
3	Grass-pasture	200	283
4	Grass-trees	200	530
5	Hay-windrowed	200	278
6	Soybean-notill	200	772
7	Soybean-mintill	200	2255
8	Soybean-clean	200	393
9	Woods	200	1065
	Total	1800	7434

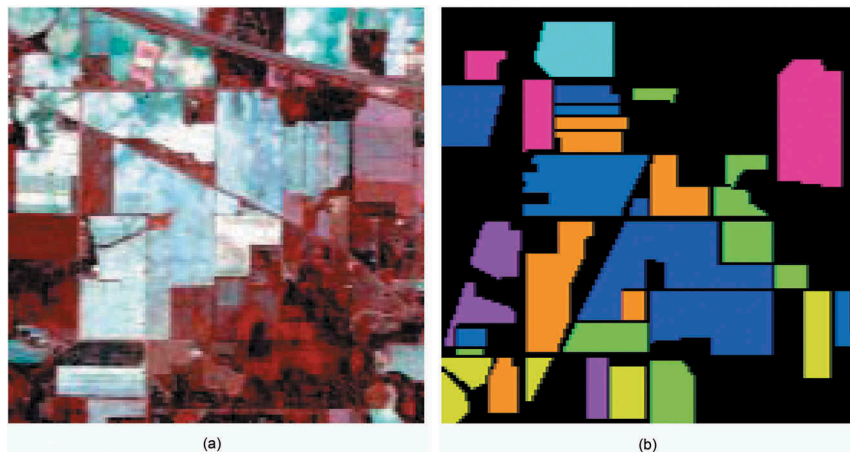


Figure 4. Indian Pines dataset. (a) False color image of three bands and (b) Ground truth map.

Table 2. Number of training and testing samples used in the University of Pavia data.

Class no.	Class name	Training samples	Testing samples
1	Asphalt	200	6431
2	Meadows	200	18,449
3	Gravel	200	1899
4	Trees	200	2864
5	Sheets	200	1145
6	Bare Soil	200	4829
7	Bitumen	200	1130
8	Bricks	200	3482
9	Shadows	200	747
	Total	1800	40,976

- (3) Salinas data is collected by the AVIRIS sensor over Salinas Valley, California with 204 bands after removing 20 water absorption bands. The image comprises 512×217 pixels with a spatial resolution of 3.7 m where 16 classes are in the image. More detailed information about the number of training and testing samples are listed in Table 3. The false-color image (band 40, band 122 and band 166) and the ground

truth map are shown in Figure 6.

In the three datasets, 200 samples are selected from each class randomly and used as training samples (Hu et al., 2015), which is suitable for demonstrating the better performance of LBP-DC-CNN with a limited number of training samples. Moreover, 200 training samples from each class can ensure relatively good performance for the compared classification methods (Lee & Kwon, 2017).

Parameters tuning

There are several important parameters in the CNN structure, such as learning rate, network width, and network depth. The classification performance of CNN framework is also sensitive to different parameters settings. Learning rate determines the convergence speed in the procedure of backpropagation (Hu et al., 2015), which can significantly affect the training performance. According to empirical study (Li et al., 2016), we set the learning rate 0.001, 0.01, and 0.001 for the Indian Pines, University of Pavia, and Salinas data, respectively.

(1) Network width

Firstly, we test the impact of different network widths. Network width determines the number of convolutional filters and the output feature maps. We fix the depth of 1D-CNN model to 5 with two convolutional layers, two max pooling layers, and one fully connected layer. Then, we test the performance with different widths of the first convolutional layer 4, 8, 16, 32, 64. These results are listed in Table 4. The accuracies of different widths have little change and no obvious rules. That is, the accuracies are not very sensitive to the network widths, and thus in our experiments, we select

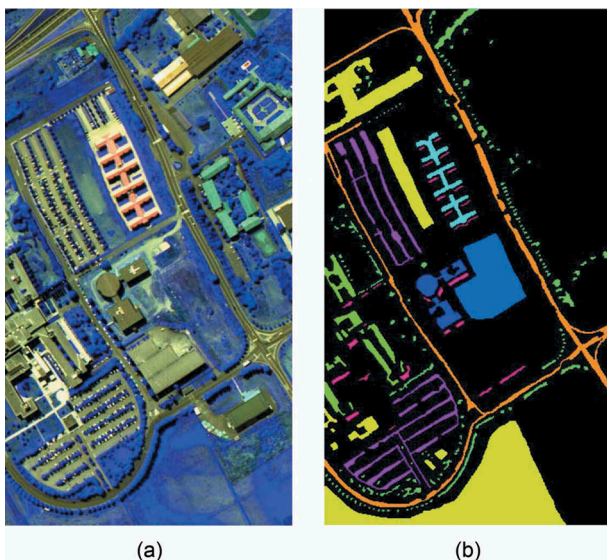
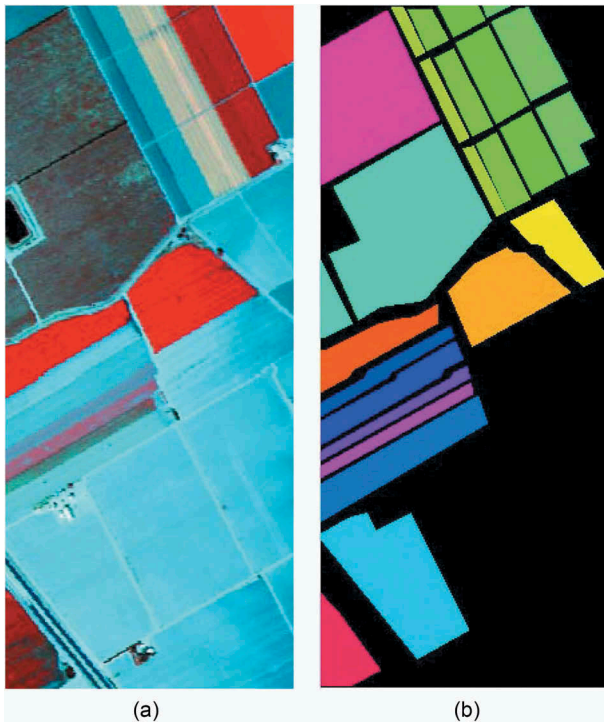


Figure 5. University of Pavia dataset. (a) False color image of three bands and (b) Ground truth map.

Table 3. Number of training and testing samples used in the Salinas data.

Class no.	Class name	Training samples	Testing samples
1	Brocoli_green_weeds_1	200	1809
2	Brocoli_green_weeds_2	200	3526
3	Fallow	200	1776
4	Fallow_rough_plow	200	1194
5	Fallow_smooth	200	2478
6	Stubble	200	3759
7	Celery	200	3379
8	Grapes_untrained	200	11,071
9	Soil_vinyard_develop	200	6003
10	Corn_senesced_green_weeds	200	3078
11	Lettuce_roumaine_4wk	200	868
12	Lettuce_roumaine_5wk	200	1727
13	Lettuce_roumaine_6wk	200	716
14	Lettuce_roumaine_7wk	200	870
15	Vinyard_untrained	200	7068
16	Vinyard_vertical_trellis	200	1607
	Total	3200	50,929

**Figure 6.** Salinas dataset. (a) False color image of three bands and (b) Ground truth map.

16, 8, 16 for the network width of Indian Pines, University of Pavia, and Salinas data, respectively.

(2) Network depth

Next, we test the impact of different network depths. With the network widths 16, 8 and 16 for corresponding data, five networks with depths 3, 5, 7, 9, 11 are tested. The experimental results are summarized in Table 5. From this table, we can conclude that the accuracy is decreased gradually with the increase of network depth, which may be a result of the gradient vanishing problem encountered by deeper neural networks. And we use the network depths of 3, 3, and 3 for Indian Pines, University of Pavia, and Salinas data, respectively.

Table 4. Overall accuracy and times of different network widths.

Dataset	Network width	4	8	16	32	64
Indian Pines	Accuracy (%)	98.32	98.20	98.35	98.33	98.23
	Time (s)	178.0	317.1	668.3	1388.0	3433.8
University of Pavia	Accuracy (%)	98.92	99.09	99.02	98.74	98.59
	Time (s)	178.6	316.8	658.6	1354.6	2816.3
Salinas	Accuracy (%)	98.63	98.82	98.89	98.67	98.54
	Time (s)	682.1	932.2	1484.9	2870.6	5436.3

(3) Max pooling layer

Finally, we test the impact of max pooling layer. According to the given network widths and network depths, we test the accuracies of designed CNN structure with max pooling layer and without max pooling layer. The experimental results are summarized in Table 6. From this table, we can conclude that CNN structure without max pooling layer may slightly increase the classification performance. Therefore, the structure of 1D-CNN model is set two convolutional layers, and one fully connected layer with given network widths.

In this section, we test the performances of the proposed methods with different parameters, such as network width, network depth, and max pooling layer. Different parameters make a difference in the classification accuracy. Network width has little effect on the accuracy and max pooling layer slightly affects the performance of the method. The chosen network width and max pooling layer have relatively good results. Appropriate network depth is very important with smaller one leading to poor extracted features and bigger one leading to gradient vanishing problem. The network depth corresponding to the best classification accuracy is chosen as the optimal parameter.

Classification performance

In order to evaluate the classification performance of LBP-DC-CNN, we compare the performance of LBP-DC-CNN with those of several state-of-the-art classifiers. Specifically, we apply a conventional

Table 5. Overall accuracy and times of different network depths.

Dataset	Network depth	3	5	7	9	11
Indian Pines	Accuracy (%)	98.46	98.35	98.17	97.88	97.65
	Time (s)	390.8	668.3	764.8	803.2	810.9
University of Pavia	Accuracy (%)	99.21	99.09	97.59	96.33	95.24
	Time (s)	199.9	316.8	337.0	351.7	358.4
Salinas	Accuracy (%)	99.16	98.89	98.32	97.18	96.21
	Time (s)	1064.7	1484.9	1639.2	2047.5	2473.2

Table 6. Overall accuracy (%) with or without max pooling layer.

Dataset	With max pooling	Without max pooling
Indian Pines	98.46	98.54
University of Pavia	99.21	99.73
Salinas	99.16	99.56

support vector machine (SVM) classifier with the Radial basis function (RBF) kernel directly to spectral signatures, in which the gamma (spread of the RBF kernel) and c (parameter that controls the amount of penalty during the SVM optimization) are set to be 2 and 16, respectively. Additionally, we employ a plain network CNN (Hu et al., 2015), which consists of two convolutional layers and two fully connected layers. We also compare an LBP-based classification approach denoted as LBP-CNN, which takes the LBP features as the input of 1D-CNN. DC-CNN (Zhang et al., 2017) is also considered to validate the advantage of dual-channel CNN framework. In addition, we compare the performance of LBP-DC-CNN with those of some state-of-art spatial-spectral hyperspectral image classification methods. Contextual CNN method (C-CNN) (Lee & Kwon, 2017) is used as a baseline in this work. A novel feature learning CNN (FL-CNN) (Mei, Ji, Hou, Li, & Du, 2017) which combines SVM and CNN to integrate both spatial context and spectral signature is used for comparison. In FL-CNN, we adopt the supervised FL-CNN and SVM as the classifier. 3D-CNN (Li et al., 2017) is proposed to take full advantage of both spectral and spatial information. Thus, we take 3D-CNN as a compared method. The number of labeled samples for training (200 samples per class) is exactly the same in different classifiers. For all experiments, we perform the experiments 10 times. Class-specific accuracy, overall accuracy (OA), average accuracy (AA), kappa coefficient (k) and the corresponding standard deviation are used to evaluate the classification accuracy.

Table 7–9 list classification accuracies for the three experimental datasets with the bold figure

representing the maximum value within a line. The results are the mean value and standard deviation of 10 times experiments. It is seen from the compared results of each individual method that the classification performance of LBP-DC-CNN is superior to those of compared classifiers. And some conclusions can be drawn from the results.

- (1) The OAs of LBP-DC-CNN are better than those of LBP-CNN and DC-CNN, which validate feature extraction capability of the CNNs and the advantage of LBP features. LBP-DC-CNN provides better accuracy than that of DC-CNN, which means an advantage of LBP features compared to the spatial features extracted by 2D-CNN model. In addition, the accuracy of LBP-DC-CNN is better than that of LBP-CNN, which validates the reasonability and discriminative power of the dual-channel CNN framework.
- (2) The OAs of LBP-DC-CNN are obviously superior to those of compared methods, which makes the best of the advantage of DC-CNN and LBP features. For Indian Pines data, LBP-DC-CNN (i.e., 98.52%) yields approximately 2% accuracy higher than the DC-CNN (i.e., 96.92%) and approximately 4% higher accuracy than the LBP-CNN (i.e., 94.72%). For University of Pavia data, LBP-DC-CNN (i.e., 99.54%) yields approximately 1% higher accuracy than the DC-CNN (i.e., 98.39%) and approximately 3% accuracy higher than the LBP-CNN (i.e., 96.35%). For Salinas data, LBP-DC-CNN (i.e., 99.54%) yields over 2% higher accuracy than the DC-CNN (i.e., 97.24%) and approximately 5% accuracy higher than the LBP-CNN (i.e., 94.62%).
- (3) The OAs of LBP-DC-CNN have an advantage over those of other spatial-spectral hyperspectral image classification methods, such as 3D-CNN, FL-CNN and C-CNN. For Indian Pines data, LBP-DC-CNN (i.e., 98.52%) yields over 3% higher accuracy

Table 7. Classification performance of the Indian Pines dataset.

Class no.	SVM	CNN	3D-CNN	FL-CNN	C-CNN	LBP-CNN	DC-CNN	LBP-DC-CNN
1	71.99 ± 3.57	80.18 ± 3.28	91.45 ± 2.82	91.39 ± 2.16	92.06 ± 1.41	90.08 ± 0.66	94.63 ± 2.81	98.58 ± 0.17
2	83.66 ± 0.97	84.50 ± 2.49	94.34 ± 1.15	96.63 ± 0.82	91.23 ± 2.69	97.15 ± 0.29	97.51 ± 0.65	98.39 ± 0.06
3	94.68 ± 3.21	93.86 ± 1.57	94.62 ± 0.95	99.79 ± 0.06	99.65 ± 0.26	95.86 ± 0.02	95.79 ± 0.10	100.0 ± 0.00
4	98.40 ± 0.13	98.40 ± 0.17	99.18 ± 0.32	100.0 ± 0.00	99.45 ± 0.11	99.27 ± 0.39	99.68 ± 0.17	97.99 ± 0.06
5	100.0 ± 0.00	99.86 ± 0.10	100.0 ± 0.00	100.0 ± 0.00	99.93 ± 0.10	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
6	91.98 ± 0.44	84.46 ± 4.32	90.95 ± 2.56	95.27 ± 0.61	84.48 ± 0.78	94.13 ± 0.22	96.47 ± 2.12	99.21 ± 0.10
7	86.59 ± 1.52	77.92 ± 0.81	94.95 ± 0.72	88.11 ± 0.68	95.73 ± 0.42	91.50 ± 0.27	95.75 ± 1.37	97.87 ± 0.08
8	89.21 ± 1.65	85.22 ± 1.46	93.59 ± 0.45	99.16 ± 0.02	86.82 ± 1.93	96.52 ± 0.08	97.13 ± 0.41	96.63 ± 0.02
9	98.87 ± 0.55	98.73 ± 0.11	98.89 ± 0.16	99.92 ± 0.16	98.72 ± 0.34	99.19 ± 0.10	99.28 ± 0.22	99.37 ± 0.03
OA (%)	88.83 ± 0.98	86.46 ± 0.42	94.96 ± 0.15	94.63 ± 0.31	93.90 ± 0.58	94.72 ± 0.03	96.92 ± 0.17	98.52 ± 0.06
AA (%)	90.59 ± 0.90	89.24 ± 0.55	95.33 ± 0.21	94.72 ± 0.16	94.23 ± 0.50	95.97 ± 0.03	97.36 ± 0.17	98.68 ± 0.05
k (%)	86.56 ± 1.16	84.20 ± 0.49	93.95 ± 0.37	93.72 ± 0.72	92.87 ± 0.67	93.83 ± 0.04	96.38 ± 0.20	98.29 ± 0.07

Table 8. Classification performance of the University of Pavia dataset.

Class no.	SVM	CNN	3D-CNN	FL-CNN	C-CNN	LBP-CNN	DC-CNN	LBP-DC-CNN
1	85.59 ± 2.65	93.09 ± 0.43	95.76 ± 0.11	98.37 ± 0.09	91.49 ± 0.40	90.89 ± 0.46	98.20 ± 0.07	99.16 ± 0.05
2	89.25 ± 0.40	91.17 ± 2.38	89.85 ± 0.37	98.38 ± 0.46	97.17 ± 0.61	97.70 ± 0.43	97.51 ± 0.57	99.48 ± 0.23
3	89.58 ± 1.11	94.06 ± 0.57	98.24 ± 0.29	99.09 ± 0.49	98.00 ± 0.49	98.06 ± 0.46	99.28 ± 0.51	99.98 ± 0.02
4	98.96 ± 0.03	99.27 ± 0.12	97.52 ± 0.54	99.54 ± 0.84	89.74 ± 0.75	89.50 ± 0.63	98.17 ± 0.62	98.86 ± 0.74
5	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	99.40 ± 0.16	99.53 ± 0.15	99.78 ± 0.16	100.0 ± 0.00
6	90.05 ± 0.51	93.92 ± 2.13	98.35 ± 0.09	99.82 ± 0.43	99.83 ± 0.10	99.70 ± 0.34	99.59 ± 0.40	100.0 ± 0.00
7	97.57 ± 0.42	97.47 ± 0.25	99.32 ± 0.08	99.85 ± 0.03	99.47 ± 0.16	99.35 ± 0.09	99.97 ± 0.04	99.90 ± 0.14
8	87.93 ± 1.12	90.39 ± 0.33	99.73 ± 0.62	98.91 ± 0.32	98.53 ± 0.34	98.76 ± 0.49	99.77 ± 0.08	99.76 ± 0.07
9	99.58 ± 0.15	99.89 ± 0.02	99.68 ± 0.06	100.0 ± 0.00	93.63 ± 0.59	92.96 ± 1.15	99.96 ± 0.05	100.0 ± 0.00
OA (%)	90.19 ± 0.42	93.11 ± 0.73	94.41 ± 0.38	98.84 ± 0.29	96.31 ± 0.37	96.35 ± 0.31	98.39 ± 0.27	99.54 ± 0.14
AA (%)	93.17 ± 0.36	95.47 ± 0.03	97.61 ± 0.25	98.65 ± 0.26	96.39 ± 0.31	96.21 ± 0.28	99.14 ± 0.16	99.68 ± 0.09
k (%)	87.25 ± 0.55	91.02 ± 0.91	92.62 ± 0.36	98.47 ± 0.29	95.13 ± 0.48	95.20 ± 0.40	97.88 ± 0.35	99.36 ± 0.15

Table 9. Classification performance of the Salinas dataset.

Class no.	SVM	CNN	3D-CNN	FL-CNN	C-CNN	LBP-CNN	DC-CNN	LBP-DC-CNN
1	99.93 ± 0.09	99.76 ± 0.05	98.46 ± 0.06	99.80 ± 0.29	99.63 ± 0.21	99.70 ± 0.04	99.27 ± 0.78	99.75 ± 0.35
2	97.76 ± 0.34	87.83 ± 4.28	97.42 ± 2.53	99.09 ± 0.43	98.30 ± 0.13	97.98 ± 0.39	98.32 ± 0.66	99.65 ± 0.31
3	96.37 ± 3.56	96.60 ± 1.27	99.19 ± 0.69	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	99.90 ± 0.14	100.0 ± 0.00
4	99.66 ± 0.27	99.59 ± 0.08	99.64 ± 0.09	99.64 ± 0.14	100.0 ± 0.00	99.78 ± 0.06	99.64 ± 0.16	99.88 ± 0.12
5	98.92 ± 0.59	94.70 ± 2.39	98.39 ± 1.46	99.10 ± 0.08	99.80 ± 0.04	97.61 ± 0.59	96.72 ± 1.08	99.76 ± 0.16
6	99.87 ± 0.16	99.70 ± 0.01	100.0 ± 0.00	100.0 ± 0.00	97.83 ± 0.64	97.65 ± 0.37	98.95 ± 0.75	99.65 ± 0.49
7	98.56 ± 0.88	99.44 ± 0.07	99.47 ± 0.38	98.60 ± 0.35	98.70 ± 0.20	96.51 ± 0.71	97.21 ± 0.21	99.40 ± 0.35
8	78.91 ± 1.85	84.33 ± 6.26	86.60 ± 0.49	95.08 ± 0.42	96.34 ± 0.38	82.63 ± 0.45	92.62 ± 0.42	98.85 ± 0.12
9	99.24 ± 0.44	99.26 ± 0.38	99.13 ± 0.25	100.0 ± 0.00	98.79 ± 0.07	97.77 ± 0.38	99.14 ± 0.45	99.98 ± 0.01
10	94.17 ± 2.06	93.38 ± 1.71	97.71 ± 0.86	99.24 ± 0.04	97.19 ± 0.39	93.97 ± 1.52	97.62 ± 1.26	99.70 ± 0.06
11	99.59 ± 0.51	97.50 ± 1.41	100.0 ± 0.00	100.0 ± 0.00	99.00 ± 0.12	98.72 ± 0.62	98.47 ± 1.13	100.0 ± 0.00
12	99.93 ± 0.05	99.63 ± 0.13	100.0 ± 0.00	100.0 ± 0.00	97.92 ± 0.29	97.50 ± 0.48	98.71 ± 0.95	99.96 ± 0.05
13	100.0 ± 0.00	99.08 ± 0.61	100.0 ± 0.00	100.0 ± 0.00	99.56 ± 0.18	99.23 ± 0.18	99.47 ± 0.38	99.81 ± 0.26
14	99.44 ± 0.20	95.48 ± 1.04	99.35 ± 0.28	99.16 ± 0.19	97.54 ± 0.27	97.97 ± 0.77	98.76 ± 0.37	99.69 ± 0.22
15	81.36 ± 7.06	52.33 ± 2.84	83.09 ± 1.64	99.34 ± 0.25	97.88 ± 0.06	97.84 ± 0.31	98.14 ± 0.22	99.61 ± 0.16
16	99.57 ± 0.20	99.07 ± 0.13	100.0 ± 0.46	100.0 ± 0.00	99.94 ± 0.08	100.0 ± 0.00	99.59 ± 0.58	100.0 ± 0.00
OA (%)	92.17 ± 0.87	88.34 ± 0.39	94.30 ± 0.10	98.61 ± 0.12	95.05 ± 0.04	94.62 ± 0.24	97.24 ± 0.07	99.54 ± 0.05
AA (%)	96.45 ± 0.74	93.63 ± 0.22	97.40 ± 0.12	99.32 ± 0.07	97.57 ± 0.07	97.18 ± 0.22	98.28 ± 0.08	99.71 ± 0.05
k (%)	91.30 ± 0.98	87.00 ± 0.47	93.43 ± 0.09	98.15 ± 0.13	94.51 ± 0.04	94.04 ± 0.28	96.93 ± 0.08	99.50 ± 0.05

than the 3D-CNN (i.e., 94.96%) and FL-CNN (i.e., 94.63%), and over 4% higher accuracy than the C-CNN (i.e., 93.90%). For University of Pavia data, LBP-DC-CNN (i.e., 99.54%) yields over 5% higher accuracy than the 3D-CNN (i.e., 94.41%) and approximately 3% higher accuracy than the C-CNN (i.e., 96.31%) and 0.7% higher accuracy than the FL-CNN (i.e., 98.84%). For Salinas data, LBP-DC-CNN (i.e., 99.54%) yields over 5% higher accuracy than the 3D-CNN (i.e., 94.30%) and over 4% higher accuracy than the C-CNN (i.e., 95.05%) and 0.9% higher accuracy than the FL-CNN (i.e., 98.61%). Compared with other spatial-spectral hyperspectral image classification methods, the style of the dual-channel CNN framework can extract spectral features and spatial features effectively and the LBP features enhanced the discriminative power of the dual-channel CNN framework.

- (4) LBP-DC-CNN can improve the class-specific accuracy of some ground materials, such as Corn-no till and Soybean-min till in Indian Pines data, Asphalt and Meadows in University of Pavia data, Grapes_untrained and Vinyard_untrained in Salinas data. The

reason is that LBP features are more discriminative than spatial features extracted by 2D-CNN and LBP-DC-CNN makes full use of LBP features and spectral features in a dual-channel CNN framework.

Figures 7–9 show the thematic maps of the labeled pixels in three experimental data, which include the classification maps of LBP-DC-CNN and compared methods. These maps are consistent with the results listed in Table 7–9, respectively. Some areas in the classification maps obtained by LBP-DC-CNN are obviously less noisy than those of LBP-CNN and DC-CNN, e.g., the regions of Grapes_untrained in Figure 9. Figure 10 illustrates the classification performance with different numbers of training samples per class. Labeled samples are usually insufficient in hyperspectral data for training, and it is necessary to investigate the sensitivity of the training sample sizes. As shown in Figure 10, the number of training samples per class is changed from 50 to 200 with an interval of 50. LBP-DC-CNN consistently performs better than the compared methods with limited labeled training samples.

Table 10 lists the training and testing times of LBP-DC-CNN and the compared methods. As seen from the table, since DC-CNN needs to process two-dimensional data and one-dimensional data at the same time, so the training time of DC-CNN is much longer than that of

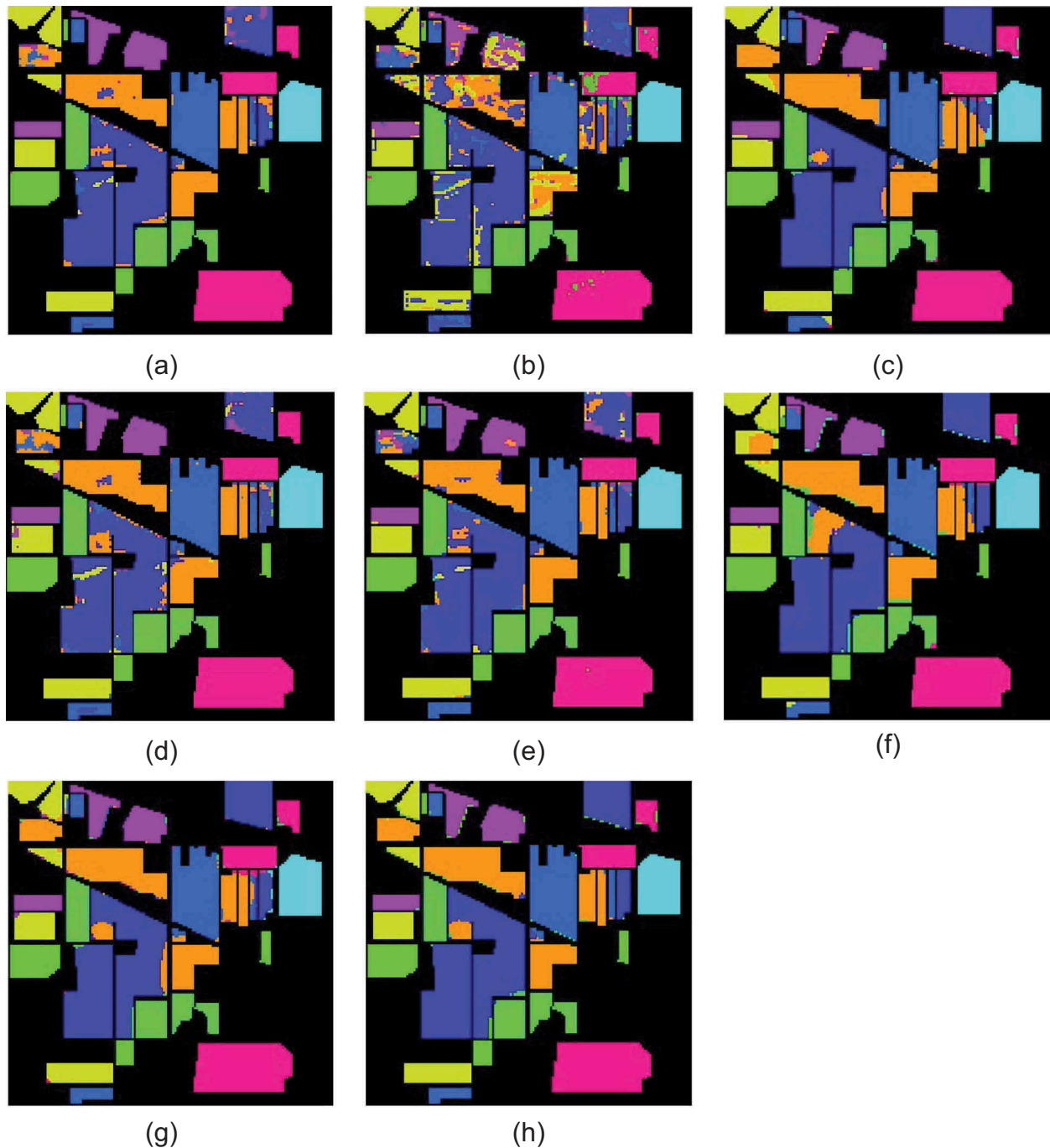


Figure 7. Classification maps obtained for the Indian Pines dataset. (a) SVM, (b) CNN, (c) 3D-CNN, (d) FL-CNN, (e) C-CNN, (f) LBP-CNN, (g) DC-CNN, (h) LBP-DC-CNN.

CNN. In addition, the LBP features are bigger than the original image data, leading to long training time of LBP-CNN. LBP-DC-CNN deals with the LBP features and original image data, making the longest training time than the compared methods.

Conclusion

In this paper, a novel classification method combining an advanced dual-channel CNN framework and LBP features, namely LBP-DC-CNN, was proposed. LBP-DC-CNN can make full use of the

feature extraction capability of the CNNs and the advantage of LBP features in hyperspectral image classification. LBP-DC-CNN adopts a 1D-CNN model to process original hyperspectral data to extract hierarchical spectral features and another same 1D-CNN model to process LBP features to further extract hierarchical spatial features. Next, the extracted features in fully connected layers are combined in the concatenation mode to feed into the softmax layer to conduct classification. In addition, the classification accuracies of LBP-DC-CNN with different numbers of training samples

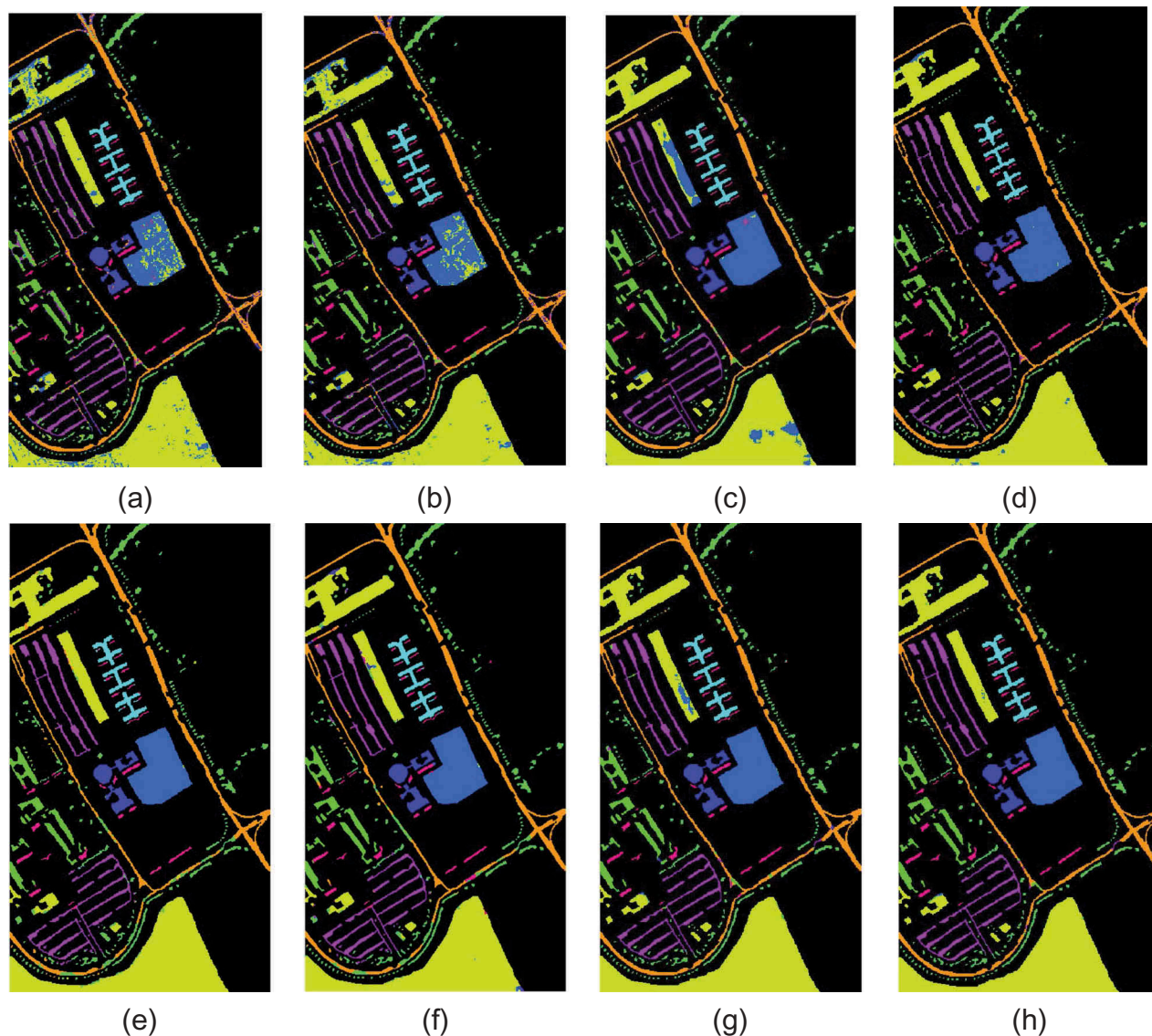


Figure 8. Classification maps obtained for the University of Pavia dataset. (a) SVM, (b) CNN, (c) 3D-CNN, (d) FL-CNN, (e) C-CNN, (f) LBP-CNN, (g) DC-CNN, (h) LBP-DC-CNN.

are validated and compared. The experimental results with three well-known datasets demonstrate that LBP-DC-CNN can effectively improve the classification accuracy even with a small number of labeled samples for training.

The proposed method LBP-DC-CNN conducts the fusion of spectral features and spatial features using dual-channel CNN framework and LBP features. In the subsequent research, extracting more discriminative and robust features in hyperspectral data using advanced CNN models will be a key point, which can take advantage of rich information in the hyperspectral image to promote the application of CNN in hyperspectral image classification.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the [National Natural Science Foundation of China] under Grant [41801388].

References

- Bandos, T.V., Bruzzone, L., & Camps-Valls, G. (2009). Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3), 862–873. doi:10.1109/TGRS.2008.2005729
- Bioucas-Dias, J., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., & Chanussot, J. (2013). Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, 1(2), 6–36. doi:10.1109/MGRS.2013.2244672
- Camps-Valls, G., & Bruzzone, L. (2005). Kernel-based methods for hyperspectral image classification. *Ieee Transactions on Geoscience and Remote Sensing*, 43(6), 1351–1362. doi:10.1109/TGRS.2005.846154.

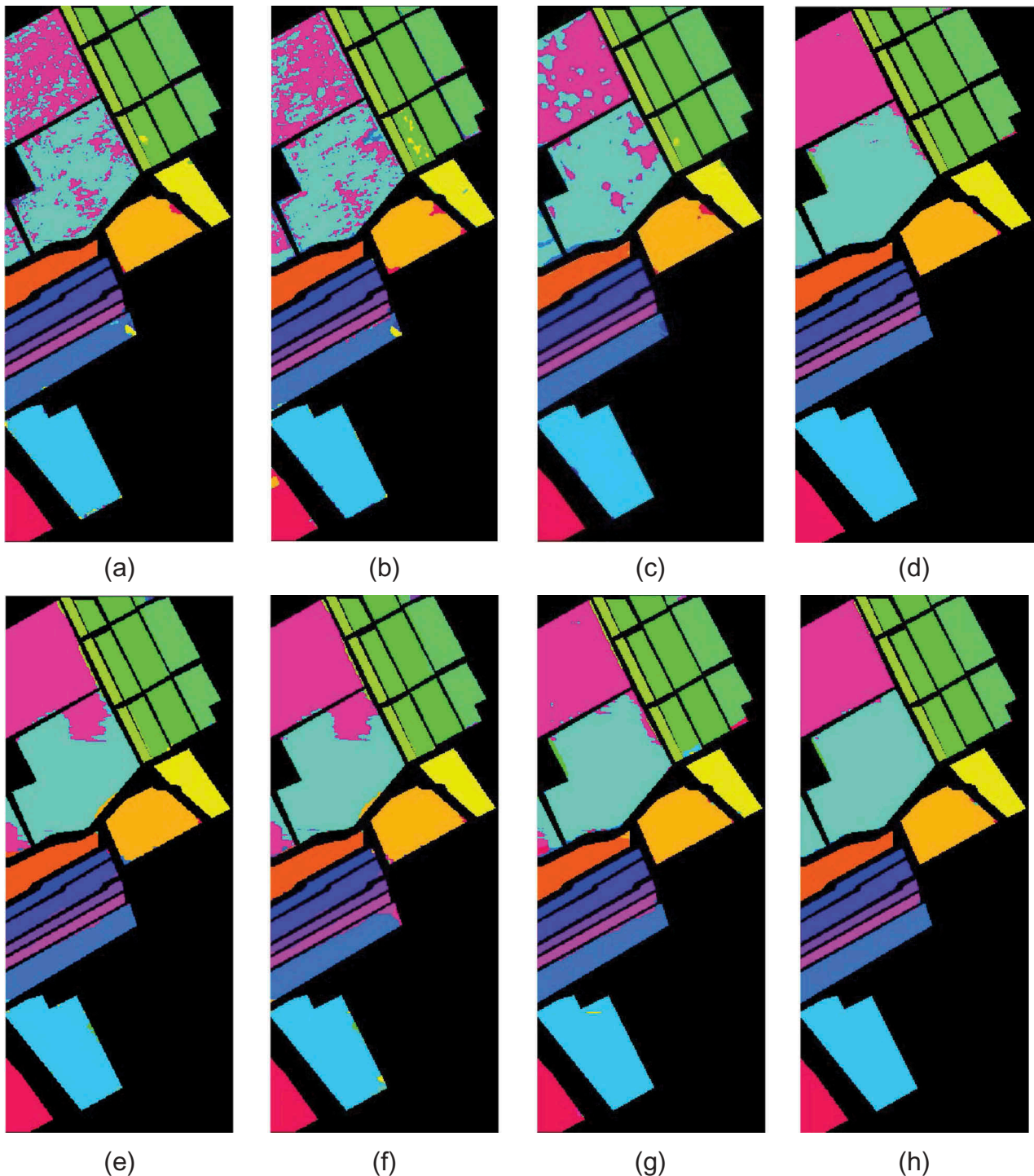


Figure 9. Classification maps obtained for the Salinas dataset. (a) SVM, (b) CNN, (c) 3D-CNN, (d) FL-CNN, (e) C-CNN, (f) LBP-CNN, (g) DC-CNN, (h) LBP-DC-CNN.

Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10), 6232–6251. doi:10.1109/TGRS.2016.2584107

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6), 2094–2107. doi:10.1109/JSTARS.2014.2329330

Chen, Y., Zhao, X., & Jia, X. (2015). Spectral–Spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, 8(6), 2381–2392. doi:10.1109/JSTARS.2015.2388577

Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G., & Tang, L. (2017). Hyperspectral images classification with gabor filtering and convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(12), 2355–2359. doi:10.1109/LGRS.2017.2764915

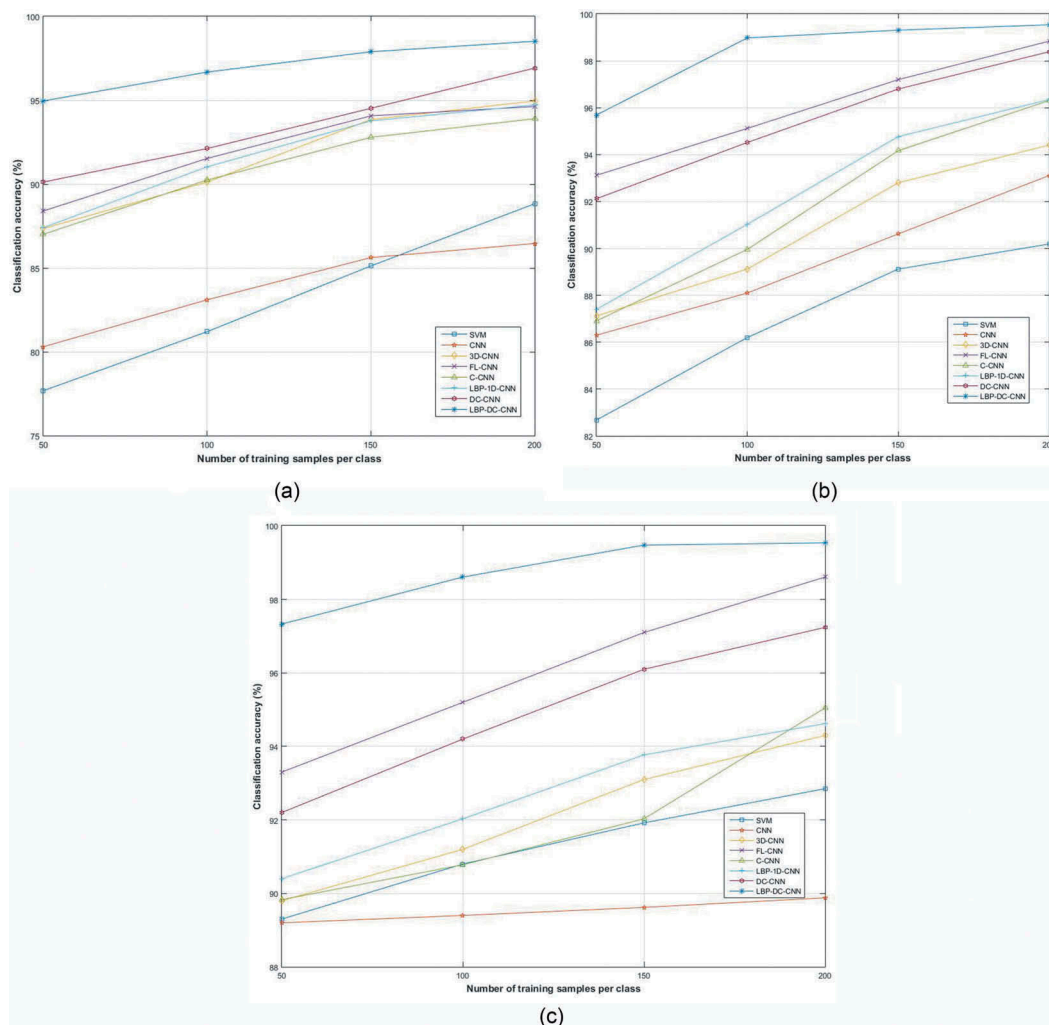


Figure 10. Classification performance with different numbers of training samples. (a) Indian Pines, (b) University of Pavia, (c) Salinas.

Table 10. Training and testing time of different methods.

Dataset	Time	CNN	3D-CNN	FL-CNN	C-CNN	LBP-CNN	DC-CNN	LBP-DC-CNN
Indian Pines	Training (s)	113.7	1132.5	1356.1	583.5	955.4	868.3	1545.8
	Testing (s)	0.46	1.83	0.81	1.46	1.66	1.59	4.63
University of Pavia	Training (s)	102.1	608.9	625.2	562.1	592.4	519.6	708.7
	Testing (s)	1.03	5.23	1.38	3.98	5.14	4.26	13.46
Salinas	Training (s)	378.6	1561.4	1684.5	812.6	1653.4	1439.2	2381.5
	Testing (s)	7.67	24.61	5.26	18.36	26.56	21.05	34.68

Computational Intelligence Group of the University of the Basque Country. Retrieved from http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes)

Dalla Mura, M., Benediktsson, J.A., Waske, B., & Bruzzone, L. (2010). Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10), 3747–3762. doi:10.1109/TGRS.2010.2048116

Du, P., Wang, X., Tan, K., & Xia, J. (2011). Dimension reduction and feature extraction from hyperspectral remote sensing imagery based on manifold learning. *Geomatics and Information Science of Wuhan University*, 36(2), 148–152. doi:10.13203/j.whugis2011.02.027

Fauvel, M., Benediktsson, J.A., Chanussot, J., & Sveinsson, J.R. (2008). Spectral and spatial

classification of hyperspectral data using SVMs and morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 46(11), 3804–3814. doi:10.1109/TGRS.2008.922034

Ghamisi, P., Plaza, J., Chen, Y., Li, J., & Plaza, A. (2017). Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geoscience and Remote Sensing Magazine*, 5(1), 8–32. doi:10.1109/MGRS.2016.2616418

Gomez-Chova, L., Tuia, D., Moser, G., & Camps-Valls, G. (2015). Multimodal classification of remote sensing images: A review and future directions. *Proceedings of IEEE*, 103(9), 1560–1584. doi:10.1109/JPROC.2015.2449668

Guo, Z., Wang, X., Zhou, J., & You, J. (2016). Robust texture image representation by scale selective local binary patterns. *IEEE Transactions on Image Process*, 25(2), 687–699. doi:10.1109/TIP.2015.2507408

- Hara, K., Saito, D., & Shouno, H. (2015). Analysis of function of rectified linear unit used in deep learning. *IEEE International Joint Conference on Neural Networks*. doi:10.1109/IJCNN.2015.7280578
- He, L., Li, J., Liu, C., & Li, S. (2017). Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99), 1–19. doi:10.1109/TGRS.2017.2765364
- Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, (2015(2), 1–12. doi:10.1155/2015/258619
- Jia, S., Deng, B., Zhu, J., Jia, X., & Li, Q. (2017). Local binary pattern-based hyperspectral image classification with superpixel guidance. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99), 1–11. doi:10.1109/TGRS.2017.2754511
- Jia, S., Hu, J., Zhu, J., Jia, X., & Li, Q. (2017). Three-dimensional local binary patterns for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99), 1–15. doi:10.1109/TGRS.2016.2642951
- Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representation*, San Diego.
- Lacar, F., Lewis, M.M., & Grierson, I.T. (2001). Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley, South Australia. *IEEE International Geoscience & Remote Sensing Symposium*, 6, 2875–2877. doi:10.1109/IGARSS.2001.978191
- Lee, H., & Kwon, H. (2017). Going deeper with contextual CNN for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10), 4843–4855. doi:10.1109/TIP.2017.2725580
- Lei, S., Mcisaac, K., & Osinski, G.R. (2018). Learning spatial-spectral features for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, PP(99), 1–10. doi:10.1109/TGRS.2018.2809912
- Li, J., Huang, X., Gamba, P., Bioucas-Dias, J.M., Zhang, L., & Benediktsson, J.A. (2015). Multiple feature learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(3), 1592–1606. doi:10.1109/TGRS.2014.2345739
- Li, J., Marpu, P.R., Plaza, A., Bioucas-Dias, J.M., & Benediktsson, J.A. (2013). Generalized composite kernel framework for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(9), 4816–4829. doi:10.1109/TGRS.2012.2230268
- Li, W., Chen, C., Su, H., & Du, Q. (2015). Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(7), 3681–3693. doi:10.1109/TGRS.2014.2381602
- Li, W., Wu, G., Zhang, F., & Du, Q. (2016). Hyperspectral image classification using deep pixel-pair features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 844–853. doi:10.1109/TGRS.2016.2616355
- Li, Y., Zhang, H., & Shen, Q. (2017). Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sensing*, 9(1), 67. doi:10.3390/rs9010067
- Licciardi, G., Marpu, P.R., Chanussot, J., & Benediktsson, J. A. (2012). Linear versus nonlinear pca for the classification of hyperspectral data based on the extended morphological profiles. *IEEE Geoscience and Remote Sensing Letters*, 9(3), 447–451. doi:10.1109/lgrs.2011.2172185
- Liu, B., Yu, X., Zhang, P., Tan, X., Wang, R., & Zhi, L. (2018a). Spectral-Spatial classification of hyperspectral image using three dimensional convolution network. *Journal of Applied Remote Sensing*, 12, 016005. doi:10.1117/1.JRS.12.016005
- Liu, B., Yu, X., Zhang, P., Yu, A., Fu, Q., & Wei, X. (2018b). Supervised deep feature extraction for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4), 1909–1921. doi:10.1109/TGRS.2017.2769673
- Ma, X., Wang, H., & Geng, J. (2016). Spectral-Spatial classification of hyperspectral image based on deep auto-encoder. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(9), 4073–4085. doi:10.1109/JSTARS.2016.2517204
- Mei, S., Ji, J., Hou, J., Li, X., & Du, Q. (2017). Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, PP (99), 1–14. doi:10.1109/TGRS.2017.2693346
- Miao, P., Han, K., Wang, B., Luo, G., Wang, P., & Chen, M. (2002). A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(2), 309–320. doi:10.1109/36.905239
- Moser, G., & Serpico, S.B. (2013). Combining support vector machines and markov random fields in an integrated framework for contextual image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5), 2734–2752. doi:10.1109/TGRS.2012.2211882
- Plaza, A., Plaza, J., & Martín, G. (2009). Incorporation of spatial constraints into spectral mixture analysis of remotely sensed hyperspectral data. *Machine Learning for Signal Processing, MLSP. IEEE International Workshop On*. doi: 10.1109/MLSP.2009.5306202.
- Romero, A., Gatta, C., & Camps-Valls, G. (2016). Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3), 1349–1362. doi:10.1109/TGRS.2015.2478379
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Computer Science, arXiv*, 1409.1556.
- Tao, C., Pan, H., Li, Y., & Zou, Z. (2015). Unsupervised spectral-Spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 12(12), 2438–2442. doi:10.1109/LGRS.2015.2482520
- Tegdan, J., Ekehaug, S., Hansen, I.M., Aas, L.M.S., Steen, K. J., & Pettersen, R. (2015). Underwater hyperspectral imaging for environmental mapping and monitoring of seabed habitats. *IEEE. OCEANS 2015, Genova*. pp.1–6. doi: 10.1109/OCEANS-Genova.2015.7271703.
- Villa, A., Benediktsson, J.A., Chanussot, J., & Jutten, C. (2011). Hyperspectral image classification with independent component discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 49(12), 4865–4876. doi:10.1109/TGRS.2011.2153861
- Yang, H., Gao, F., Dong, J., & Yang, Y. (2018). Hyperspectral image classification based on local binary patterns and PCANet. *International Conference on Graphics & Image Processing*, Beijing.
- Ye, Z., Fowler, J.E., & Bai, L. (2017). Spatial-spectral hyperspectral classification using local binary

- patterns and Markov random fields. *Journal of Applied Remote Sensing*, 11(3), 035002. doi:[10.1117/1.JRS.11.035002](https://doi.org/10.1117/1.JRS.11.035002)
- Yue, J., Zhao, W., Mao, S., & Liu, H. (2015). Spectral–Spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sensing Letters*, 6(6), 468–477. doi:[10.1080/2150704X.2015.1047045](https://doi.org/10.1080/2150704X.2015.1047045)
- Zhang, H., Li, Y., Zhang, Y., & Shen, Q. (2017). Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sensing Letters*, 8(5), 438–447. doi:[10.1080/2150704X.2017.1280200](https://doi.org/10.1080/2150704X.2017.1280200)
- Zhao, G., & Pietikäinen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 915–928. doi:[10.1109/TPAMI.2007.1110](https://doi.org/10.1109/TPAMI.2007.1110)
- Zhao, W., & Du, S. (2016). Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 113, 155–165. doi:[10.1016/j.isprsjprs.2016.01.004](https://doi.org/10.1016/j.isprsjprs.2016.01.004)