Hyperspectral Image Classification Using Multi Vote Strategy on Convolutional Neural Network and Sparse Representation Joint Feature

Daoming Ye[1,2], Rong Zhang[1,2(✉)], and Dixiu Xue[1,2]

[1] Department of Electronic Engineering and Information Science, USTC, Hefei 230027, China
luckystar_ydm@163.com, zrong@ustc.edu.cn,
xuedixiu@mail.ustc.edu.cn
[2] Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei, China

Abstract. Classification is one of the most popular topics in hyperspectral image (HSI). This paper proposes a method that uses multi vote strategy on convolutional neural network and sparse representation joint feature in hyperspectral image classification. First, the labeled spectral information was extracted by Principal Component Analysis (PCA) as well as the spatial information, at the same time, we feed the convolutional neural network and sparse representation joint feature to SVM. Then, we use multi-vote strategy to get the final result. Experimental results based on public database demonstrate that the proposed method provides better classification accuracy than previous hyperspectral classification methods.

Keywords: Convolutional neural networks · Sparse representation
Support Vector Machines · Multi vote

1 Introduction

Hyperspectral image (HSI) contain hundreds of continuous narrow spectral bands, which span from visible to infrared spectrum. They are widely used in mineralogy, agriculture and surveillance. However, the high dimensionality and limited training samples of hyperspectral data increase the difficulty. It is different from usual image classification that hyperspectral image classification focus on solving problem of pixel-level, which is similar to image segmentation. To get better classification accuracy, we should add environmental factors.

Several traditional methods have been used in hyperspectral image classification, such as KNN, maximum likelihood, Support Vector Machine (SVM), logistic regression, sparse representation and so on. By building a low-dimensional subspace, Sparse representation generates a dictionary, where an unknown test pixel can be sparsely represented by a few training samples, then the sparse representation vector will include the class information. Dictionary-Based Sparse Representation in [1] and Kernel Sparse

Y. Zhao et al. (Eds.): ICIG 2017, Part III, LNCS 10668, pp. 337–346, 2017.
https://doi.org/10.1007/978-3-319-71598-8_30

Representation in [2] are useful in classification in HSI. SVM [3, 4] can handle large input space efficiently as well as deal with noisy samples robustly.

In last several years, convolutional neural network methods achieve promising performance on many vision-related tasks, including image classification, scene labeling and face recognition. This model learns a good representation of features. The idea of CNNs was firstly introduced in [5], improved in [6], and refined in [7]. To deal with classification issue of HSI, the one dimension pixels have been treated as input in [8], simple feature maps generated by original pixels are also used in [9], Graph-based method has been proposed in [10].

By linear transformation, Principal Component Analysis (PCA) can extract the main characteristic components of the data as well as reduce the dimensionality. Convolutional Neural Network (CNN) implicitly learn feature from the training data which is closer to the actual biological neural network. The majority of sensory data can be represented as a superposition several atoms with associated weights by solving an optimization problem constrained by the sparsity level and reconstruction accuracy. In this paper, we propose a method which uses SVM to classify the joint feature generated by sparse representation and convolutional neural network and uses spatial labels to vote for the result, which consists of spatial information twice. First, we get spatial information and generate spatial-spectral samples by PCA. Second, we use classification results around to vote for the final result.

2 Algorithm Architecture

In this section, we first use PCA to extract spatial and spectral information from original HSI. Second, we respectively get CNN feature and sparse representation feature, and splice them into the joint feature as the input of SVM. Last, we vote for the output of SVM result with classification result of neighbors. The framework of the proposed method is shown in Fig. 1.

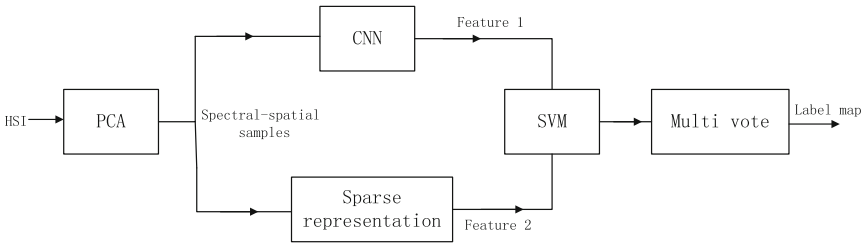


Fig. 1. Algorithm framework.

2.1 PCA

To get better classification results, we take spatial information into consideration [11]. Due to the hundreds of channels along the spectral dimension, to reduce the data dimension to a reasonable scale is necessary. We choose PCA approach to retain main

information as well as reduce the spectral dimension to specified length L. After using PCA on the whole image regardless of whether the pixel has been marked, we select and flatten the eight tangent neighbors to form the spatial data. Then we can extract feature from labeled pixel samples with spatial and spectral information. Figure 2 shows the detailed extraction process.

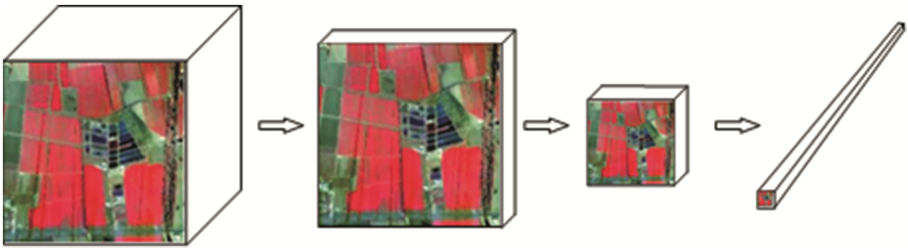


Fig. 2. PCA framework.

2.2 Convolutional Neural Networks

Convolutional neural networks which are designed to learn features directly from the images consist of various combination of convolutional layers, max pooling layers and fully connected layers. CNN learn from training data implicitly instead of extracting the feature explicitly. CNN reduce the number of parameters significantly with the help of weight sharing and local perception.

Each HSI pixel sample can be regarded as a 2D image whose height is equal to 1, so the input layer is (n, 1), where n is the length of spatial-spectral information. The framework of CNN is shown in Fig. 3. The first hidden convolutional layer filters the $n \times 1$ input data with n_1 kernels of size $k_1 \times 1$, then the max pooling layer is the second

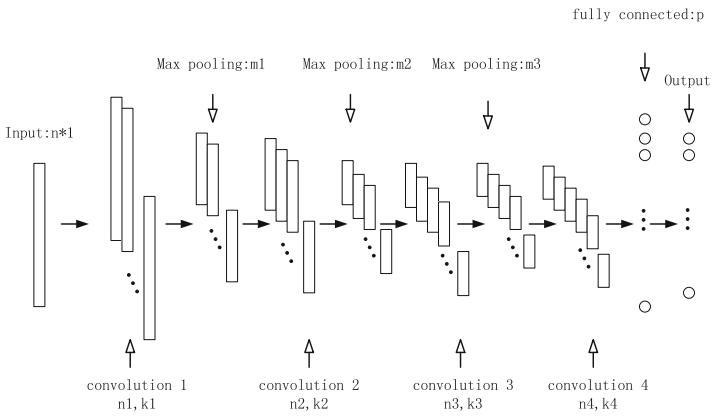


Fig. 3. CNN framework.

hidden layer with the kernel size of $m_1 \times 1$. The later 3 convolutional layers with parameters $n_2, n_3, n_4, k_2, k_3, k_4$ and 2 max pooling layers with parameters m_2, m_3 are similar. The fully connected layer which follows by the flatten layer has p nodes, and the output layer has the same nodes as the number of the classification. We use the result of the fully connected layer as generated feature where the length is p while the output layer assists training.

2.3 Sparse Representation

The pixels belonging to the same class approximately lie in the same low-dimensional subspace. For an unknown test pixel y, there is a similar expression of linear combination of all of the training pixels as:

$$y = Dx \tag{1}$$

Where D is an over-complete dictionary generated by training pixels from all classes, and x is the sparse vector with few non-zero entries. To avoid each feature to be assigned to too many clusters, the sparse constraint on the weight vector x is usually added to the objective. We can obtain x by solving the problem:

$$x = argmin\|y - Dx\|_2 + \lambda\|x\|_0 \tag{2}$$

Where $\lambda > 0$ is used to balance sparsity and reconstruction accuracy. We can solve the problem approximately by Orthogonal Matching Pursuit. During the training period, we calculate the dictionary D, which can be used to get the sparse vector x for the test samples. In this way, we transform the original spectral-spatial pixel samples into a few sparse feature where the length is q, then we can combine them with the feature generated by CNN to participate in later classification.

2.4 SVM

Given a training set of N data points $\left\{x_k, y_k\right\}_{k=1}^{N}$, where $x_k \in R^n$ is the kth input pattern and $y_k \in R^n$ is the kth output pattern [12], the support vector method approach aims at constructing a classifier of the form:

$$y(x) = \text{sign}\left[\sum_{k=1}^{N} a_k y_k \psi\left(x, x_k\right) + b\right] \tag{3}$$

where a_k are positive real constants and b is a real constant. The classifier is constructed as follow:

$$y_k\left[w^T \varphi\left(x_k\right) + b\right] \geq 1 - \xi_k, \ \xi_k \geq 0, \ k = 1, \ldots, N, \tag{4}$$

where $\varphi(.)$ is a nonlinear function which maps the input space into a higher dimensional space. SVM solution is not applied directly on original spectral-spatial data, instead, we

prefer to extract deep feature first. Feature generated by CNN and sparse coding plays a role as the input of SVM method.

2.5 Multi Vote

The pixel samples to be classified are not real independent. We can reconstruct a pre-labeled image with classification results as well as untagged pixel samples which are marked as zero.

As it shows in Fig. 4, the central location (j, i) is the classification result of test sample. The central value is 5 while most results around are 2, then the test sample may be classified by mistake. There are many isolated point which are different from neighbors. We use the spatial label around each test pixel's neighborhood to smooth it. To reduce smoothing error, we use samples nearby both in geometric space and in spectral space.

	i-2	i-1	i	i+1	i+2
j-2	1	1	2	2	2
j-1	2	2	2	2	1
j	2	2	**5**	2	2
j+1	1	2	2	2	2
j+2	1	2	0	0	0

Fig. 4. Multi vote framework.

The proposed framework takes all the tagged pixels in a flat neighbor into consideration, where the k nearest neighbors in spectral dimensional space may vote for the final result.

3 Experiments

In this section, we use two hyperspectral databases, including Salinas, and University of Pavia scenes to verify the feasibility of our algorithm. All the programs are implemented using Python language and Theano library, which is efficient and convenient on CNN computation.

3.1 The Databases

The Pavia University scene is acquired by the ROSIS sensor during a flight campaign over Pavia, north Italy. Pavia University is a 610 * 610 pixels image with 103 bands, but some of the samples contain no information and have to be discarded before the analysis. The geometric resolution is 1.3 m. Image ground truth differentiates 9 classes,

including Asphalt, Meadows, Gravel, Trees, Painted metal sheets, Bare Soil, Bitumen, Self-Blocking Bricks, Shadows and so on (Table 1).

Table 1. Classes and samples number for the Pavia University scene

	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

The Salinas scene is collected by the 224-band AVIRIS sensor over Salinas Valley, California, and is characterized by high spatial resolution (3.7-m pixels). The area covered comprises 512 lines by 217 samples. We discard the 20 water absorption bands, and the left 204 bands are used to describe 16 classes (Table 2).

Table 2. Classes and samples number for the Salinas scene

	Class	Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807

3.2 Parameter Setting

For each scene, we choose 30% from all tagged pixels randomly as training samples, and the rest are testing samples. To spectral-spatial information, we set different length L in PCA method to keep the spectral and spatial information are not far-off. The Pavia University scene has 103 bands, we set L = 10, then the length of spatial information is 3 * 3 * 10. The Salinas scene has 204 bands, we set L = 20, then the length of spatial information is 3 * 3 * 20.

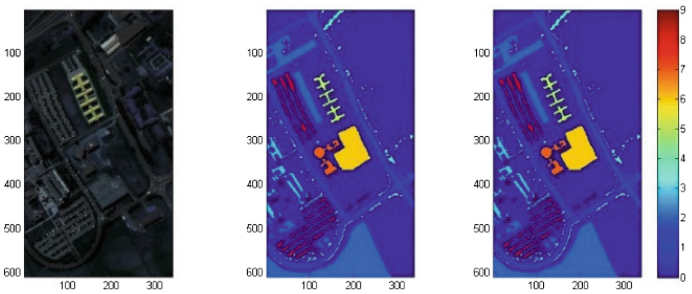
In CNN architecture, we set $n_1 = 12, n_2 = 32, n_3 = 64, n_4 = 128$, $m_1 = 2, m_2 = 2, m_3 = 2, k_1 = 4, k_2 = 5, k_3 = 4, k_4 = 5, p = 1000$.

In sparse representation, we set the length of sparse vector $q = 200$, so the combined feature as the input of SVM classifier has total length of 1200.

In multi vote architecture, we take 5 * 5 window around each testing sample into consideration, while the k nearest neighborhoods contain 11 labels with smaller Euclidean distance in spectral dimensional space. If there are more than 6 labels with the same value not equal to the classification result of the testing sample, we will cover the previous classification result.

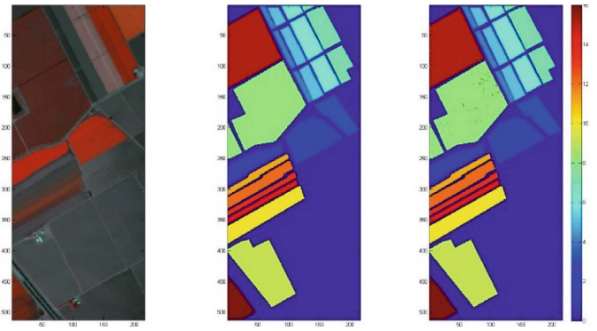
3.3 Result and Analysis

Figures 5 and 6 show the experiment results of proposed method with different public databases. Each figure includes three parts: the original image, the ground truth and the result of proposed method. most test samples are correctly classified with proposed method.



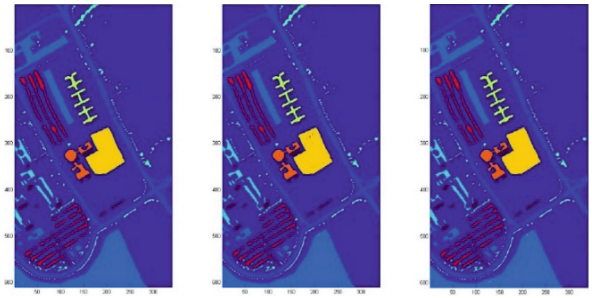
(a) Original image (b) ground truth (c) proposed method.

Fig. 5. Experiment on Pavia University image.



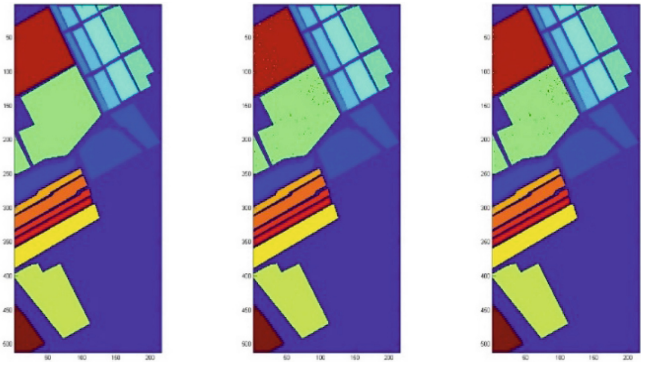
(a) Original image (b) ground truth (c) proposed method.

Fig. 6. Experiment on Salinas image.



(a) Original image (b) before multi vote (c) after multi vote.

Fig. 7. Experiment on Pavia University image.



(a) Original image (b) before multi vote (c) after multi vote.

Fig. 8. Experiment on Salinas image.

Figures 7 and 8 show the result before and after multi vote method as follows. The result of feeding the sparse representation and convolutional neural network joint feature to SVM is quite good, the labels of pixels around each text sample are basically correct,

then the multi vote method effective removes isolated points, which are probably estimated by mistake.

We design several comparative experiments. The CNN result uses only 1-D CNN to extract feature, while the SSDCNN transform 1-D information into 2-D feature maps as the input of CNN. The best performance has been highlighted in bold. As shown in Tables 3 and 4, the proposed model outperforms other methods in each indices. Overall accuracy (OA), average accuracy (AA) and Kappa coefficient (κ) are used as measurement indices.

Table 3. Classification accuracy for University of Pavia

	RBF-SVM	CNN	SSDCNN	Proposed
OA	0.9052	0.9256	0.9518	**0.9836**
AA	0.8927	0.9128	0.9351	**0.9792**
κ	0.8964	0.9143	0.9364	**0.9784**

Table 4. Classification accuracy for Salinas

	RBF-SVM	CNN	SSDCNN	Proposed
OA	0.9166	0.9260	0.9408	**0.9802**
AA	0.9503	0.9581	0.9743	**0.9920**
κ	0.9056	0.9146	0.9339	**0.9780**

4 Conclusion

In this paper, we proposed a classification method for HSI. We use spatial information twice. In PCA part, spatial information is used to form spectral-spatial samples, and in multi vote part spatial information is used to smooth isolated point. SVM is applied to generate the previous classification map with CNN and joint feature. Through further smooth, classification problem was transformed into segmentation problem. The proposed method could achieve higher accuracy on each public database.

References

1. Chen, Y., Nasrabadi, N., Tran, T.: Hyperspectral image classification using dictionary-based sparse representation. IEEE Trans. Geosci Remote Sens. **49**(10), 3973–3985 (2011)
2. Chen, Y., Nasrabadi, N., Tran, T.: Hyperspectral image classification via kernel sparse representation. IEEE Trans. Geosci Remote Sens. **51**(1), 217–231 (2013)
3. Gualtieri, J.A., Cromp, R.F.: Support vector machines for hyperspectral remote sensing classification. In: Proceedings SPIE 27th AIPR Workshop: Advances in Computer Assisted Recognition, Washington, DC, October 1998, vol. 3584, pp. 221–232 (1998)
4. Melgani, F., Bruzzone, L.: Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans. Geosci. Remote Sens. **42**(8), 1778–1790 (2004)
5. Fukushima, K.: Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw. **1**(2), 119–130 (1988)

6. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2323 (1998)
7. Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), vol. 22, pp. 1237–1242, July 2011
8. Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H.: Deep convolutional neural networks for hyperspectral image classification. J. Sensors **2015**, 12 (2015). Article ID 258619, Hindawi Publishing Corporation
9. Yue, J., Zhao, W., Mao, S., Liu, H.: Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. Remote Sens. Lett. **6**(6), 468–477 (2015)
10. Cao, J., Chen, Z., Wang, B.: Graph-based deep Convolutional networks for Hyperspectral image classification. In: IGARSS 2016 (2016). ISBN 978-1-5090-3332-4/16
11. Tarabalka, Y., Benediktsson, J.A., Chanussot, J.: Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques. IEEE Trans. Geosci. Remote Sens. **47**(8), 2973–2987 (2009). View at Publisher, View at Google Scholar, View at Scopus
12. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**, 293–300 (1999)