A Semi-supervised Classification Method for Hyperspectral Images by Triple Classifiers with Data Editing and Deep Learning

Guoming Zhang[1,2], Junshu Wang[3,4], Ge Shi[3,4], Jie Zhang[1], and Wanchun Dou[1](✉)

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
kelvinzhang@smail.nju.edu.cn, njujiezhang@gmail.com,
douwc@nju.edu.cn

[2] Health Statistics and Information Center of Jiangsu Province, Nanjing, China

[3] Key Laboratory for Virtual Geographic Environment, Ministry of Education,
Nanjing Normal University, Nanjing, China
jlsdwjs@126.com

[4] Jiangsu Center for Collaborative Innovation in Geographical Information Resource
Development and Application, Nanjing, China

Abstract. A semi-supervised classification method for hyperspectral remote sensing images based on convolutional neural network (CNN) and modified tri-training is proposed. The abstract features are captured by training a CNN model with the pixels' vectors as inputs. Based on the extracted high-level features, different classifiers will perform different outputs under the same training set, due to the different types of classifiers take on diverse characteristics. Thus, taking multiple classifiers' results into consideration can integrate different prediction labels synthetically from a high level and can perform more credible results. At the meantime, the number of training samples of hyperspectral images is limited, which will hinder the classification effect. Illuminated by tri-training algorithm, we utilize triple different classifiers to classify the hyperspectral images based on the extracted high-level features in the semi-supervised mode. By utilizing triple classifiers jointly to train and update the training samples set when the number of labeled samples is limited. At the meantime, we pick the confident samples via randomize and majority vote into the training set for data editing during the iterative updating process. Experiments performed on two real hyperspectral images reveal that our method performs very well in terms of classification accuracy and effect.

Keywords: Deep learning · Hyperspectral images (HSI) classification · Tri-training · Data editing

1 Introduction

With the development of imaging spectrometry technology, hyperspectral remote sensing application achieves more and more attention, due to the enormous ability of describing the detailed land covers, such as precision agriculture, anomaly detection, mineral

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 171–183, 2020.
https://doi.org/10.1007/978-3-030-48513-9_14

resources, etc. For the hyperspectral remote sensing images, the goal of classification is to assign each pixel with a unique type of land cover label.

Actually, the high dimensional hyperspectral data always lead to some challenging problems, such as Hughes phenomenon. The unbalance between the number of labeled samples and the spectral dimensionality always decades the classification. Feature extraction (FE) is an effect way to tackle this ill-posed problem. Before 2013, the available FE [1, 2] and diversifying methods were generally designed for the shallow models. Lots of popular shallow learning models and algorithms have been developed in hyperspectral images area and achieved great success, such as Bayes [3], SVM [4], conditional random fields [5] and multinomial logistic regression [6]. The shallow expression method needs to rely on the prior knowledge of remote sensing professionals and mainly relies on the manual design features, which is sensitive to parameters and only a few parameters are usually allowed. Meanwhile, affected by external conditions such as the imaging process, the image quality will be greatly different, so the rules of feature extraction/selection should be formulated according to the image's characteristics. Therefore, most feature extraction/selection methods have the problem of poor generalization. Usually, the shallow models is effective to the linear data, but their ability to deal with the nonlinear data, take the hyperspectral images for example, is limited.

High-level representation and classification by deep learning can avoid the complexity of manual design features and automatically learn high-level representation of data through increasing network layers, which brings new development opportunities for hyperspectral image classification. In this work, we propose a semi-supervised classification framework based on abstract features extracted by CNN. The key idea of the semi-supervised model integrates the modified tri-training algorithm and the data editing strategy, to explore the information gain and positive effect among classifiers on the classification task. Since the performance of different classifiers on the same feature is various, our proposed approach, could improve the final classification results by integrating their predicted results.

The rest of this paper is arranged as follows. Section 2 reviews the previous related work. Section 3 depicts the preliminary, and develops the proposed method named DMTri-training, which modified Tri-training with data editing, based on the features extracted by CNN. Section 4 evaluates the proposed method over two real-world hyperspectral images. Finally, Sect. 5 summarizes our study.

2 Related Work

In hyperspectral images application, deep learning methods plays an important role. Stack autoencoder [7] and sparse-constrained autoencoder [8] and deep belief network [9] have been applied to the processing of hyperspectral images. In order to solve the problem that automatic encoders and deep belief networks cannot directly extract spatial features, the advantages of convolutional neural network in extracting image features are utilized to extract spectral features through 1-dimensional CNN and 2-dimensional CNN to extract spatial features [10]. The combination of CNN and other shallow models, such as CRF [5], sparse dictionary learning [11], transfer strategy [12] have been successfully used to provide more comprehensive spectral and spatial information for classification to obtain better classification results.

Semi-supervised sparse expression [13], semi-supervised logistic regression [14], semi-supervised support vector machine (SVM) [15], graph-based method [16], generative model [17], EM [18] and divergence-based method [19] have been well applied in hyperspectral images. Divergence-based methods utilize multiple learners to predict samples and select unlabeled samples from the predicted results of multiple classifiers to assist the classification process. When the number of learners is single, it is self-training learning [20]. When the number of classifiers is two, it is the classic co-training algorithm [21]. When the number of learners is 3, it is the famous tri-training learner [22]. In this work, we take 1-dimensional CNN for consideration and treat each pixel as a spectral vector to extract abstract features. We modified the tri-training algorithm to adjust the hyperspectral data, and introduce data editing and majority voting at the process of adding new samples into the training set at each iteration, which improves the classification results compared with other methods.

3 Methods

3.1 Preliminary

Hyperspectral remote sensing image integrates the spectrum standing for the radiance of the land cover with images on behalf of the spatial and geometric relationships. Usually, a hyperspectral image can be taken as a data cube, $X \in \mathbb{R}^{m \times n \times L}$, where m, n and L denote the number of samples (width), lines (height) and bands (depth) of the HSI respectively. Generally, there are hundreds of bands in the hyperspectral images, that is, the spectral resolution of is very high, which make it possible to describe the land cover in detail. In order to facilitate the following processing for HSI images, the image cubes are often transformed into 2D matrices, $X \in \mathbb{R}^{mn \times L}$. The column L denotes the number of samples in total of the image, and the row mn represents the radiance of each pixel. Feature extraction or representation can remove the redundant data and perform a better representation, at the meantime, reduce the data dimension. In this work, we take a CNN model to learn the high-level feature representation of HSI. Thus, the 2D data, $X \in \mathbb{R}^{mn \times L}$ can be simplified into another formula, $X \in \mathbb{R}^{mn \times d}, d < T$. In a HSI classification task, given the training data set $T \in \mathbb{R}^{mn \times d}$ and the corresponding labels set $y \in \{y_1, \ldots, y_m\}, m \in (1, \ldots, C)$, C is the number of categories. The goal of classification is to assign a class label to each sample in the hyperspectral image.

3.2 Feature Extraction Based on CNN

CNN is a kind of feedforward neural networks, it includes input layer, convolution layer, pooling layer and full connection layer. The CNN structure is shown in Fig. 1.

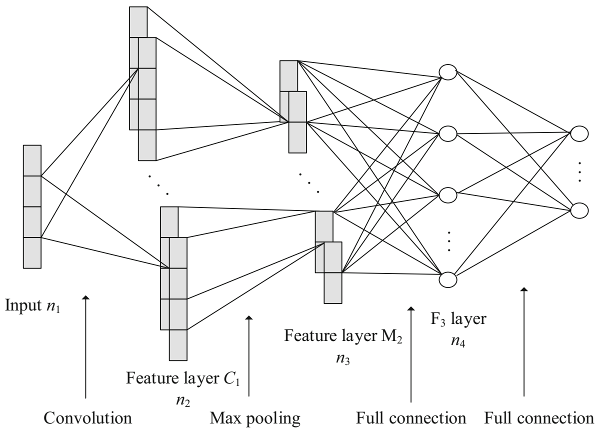


Fig. 1. CNN network structure.

As for hyperspectral images, each pixel can be regarded as a two-dimensional image with the size of 1 × n_1, which serves as input layer of the network, and n_1 is the number of spectral bands. The size of the input layer is (n_1,1). Suppose the convolution layer C_1 contains twenty kernels with size of $k_1 \times 1$. After the convolution operation on the input data by layer C_1, we can obtain 20 × n_2 × 1 nodes, where $n_2 = n_1$-$k_1 + 1$. The kernel size of the maximum pool layer M_2 is (k_2,1), and the M_2 layer contains 20 × n_3 × 1 nodes, where $n_3 = n_2/k_2$. Full link layer F_3 has n_4 nodes, which is the number of extracted features. The output layer has n_5 nodes, which is the number of categories of data to be classified. In this work, we only utilize the training samples to construct CNN feature extraction model. Each hidden layer h_i is multiplied by the input node V and the weight W, and the neuron nodes in each layer share the weight W, which reduces the complexity of network parameter number and parameter selection. In most cases, training samples are randomly selected, and it is generally assumed that training samples have the same feature distribution as test samples. In our work, we take the *relu* function as the activation function of CNN model, and some neurons are randomly discarded by Dropout to prevent overfitting.

3.3 DMTri-Training Algorithm

During the iterative learning process, the semi-supervised classification algorithm usually needs to evaluate the predicted confidence of unlabeled samples, which is always time consuming. Whereas tri-training omits this process, which eliminates the computational complexity to some extent and gives the final prediction via majority voting from the classifiers. Based on the characteristics of hyperspectral image data, we proposes the DMTri-training algorithm, which improves the tri-training algorithm as follows:

1) Tri-training utilizes a single supervised learning algorithm, and the diversity of the initial three classifiers is obtained by training the labeled data via bootstrap sampling from the pool of original labeled samples set. Whereas, DMTri-training trains three

different supervised learning algorithms based on the same data sets generated via randomized sampling from the original labeled samples set.

2) DMTri-training takes the predicted results of labeled and unlabeled samples into account synthetically. Then computes the classification error rate of labeled samples err_1 and the inconsistent predicted results between two classifiers to the total sample in unlabeled sample set err_2. The total error rate err is calculated as follows: $err =$ ERR_WEIGHT*err_1 + (1-ERR_WEIGHT)*err_2.
3) $l_i^{'}$ is the number of new labeled samples added into training set at i-th iteration. In tri-training algorithm, $l_i^{'}$ is very small, that is, the new added training samples are limited at the first time. The predicted error rate of labeled samples maybe do not meet the condition $e_i < e_i^{'}$, which will result in the semi-supervised process cannot continue and a low classification accuracy result. Our algorithm verified the initial value of $l_i^{'}$ to $(|L_i| * e_i)/e_i^{'} + 1$, in order to satisfy the iterative condition and increase more samples at each iteration.
4) When adding new samples into training set, we utilize randomization and data editing strategy, and integrate the samples predicted by the three classifiers to filtrate the misclassified ones. The rules are as follows: first, we adopt random sampling strategy to pick some labeled samples into training set, which can reduce the quantity of training samples and speed up the learning process. Meanwhile, the random sampling ensure more information gain. Second, these samples are filtrated by data editing. According to the rules of neighborhood consistency, if the two classifiers, take h_1 and h_2 as example, do not consider the sample i need to be filtrated, which means i is a high confident sample. Then sample i should be put into the classifier h_3, otherwise the sample i will be removed as a noisy example.
5) If one sample's labels predicted by the three classifiers are not consistent with each other, this sample always brings high information gain, and should be added into training set after manual correction.
6) Furthermore, we take secondary data editing during the training process. After each iteration, rectify the mislabeled samples automatically by nearest neighbor voting rule, then execute data editing to obtain high confident samples, and put them into the training set for next iteration and prediction.

L denotes the initial training set. In each iteration, the classifiers h_1 and h_2 select some samples and predict their labels, then put them into the training set of classifier h_3. L^{t-1} and L^t represent the new added labeled samples into classifier h_3 at the (t-1)-th and t-th iteration respectively, the corresponding training set of h_3 is $L \cup L^{t-1}$ and $L \cup L^t$. η_L denotes the classification error rate of the training set L, and the number of misclassified samples is $\eta_L |L|$. e_1^t represents the upper bound of classification error rate at *t-th* iteration. Suppose that the number of consistent samples predicted by h_1 and h_2 is z, in which the number of samples with correct predicted labels is z', then we can infer $e_1^t = (z - z')/z$. The number of mislabeled samples in set L^t is $e_1^t \left| L^t \right|$. At the t-th iteration, the error classification rate is:

$$\eta^t = \frac{\eta_L |L| + e_1^t |L^t|}{|L \cup L^t|} \tag{1}$$

4 Experimental Results and Comparisons

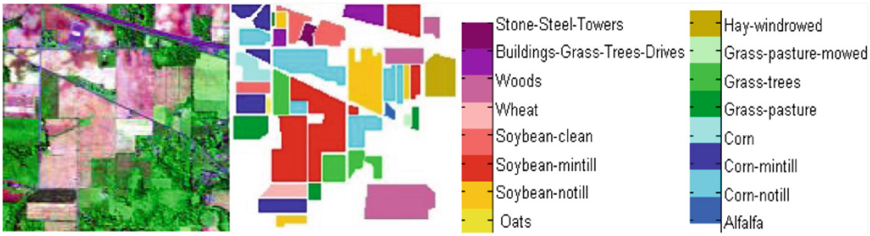
4.1 Data Set

a. Indian Pines

Indian Pines located in northwest Indiana, the data was acquired by an airborne visible/infrared imaging spectrometer (AVIRIS) in 1992. The wavelength range from 0.4–2.5 μm, 220 bands in total, each pixel (with 145 × 145 pixels) has 30 m spatial resolution. After removing bad bands and water absorption band, 200 bands were available for experiment. Figure 2 (a) shows the RGB image of the Indian Pines, with red, green and blue bands of 70, 45 and 9 respectively. Figure 2 (b) is the corresponding ground truth map, with 10,249 samples from 16 different categories of land cover.

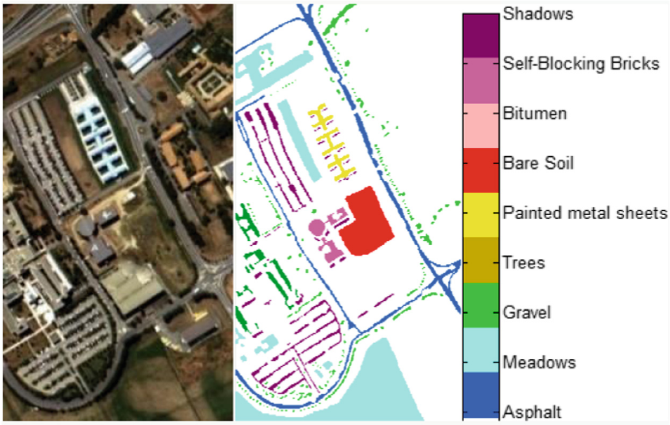
b. Pavia University

Pavia University located in Pavia, Italy, the data was acquired by ROSIS sensor. The spectral range from 0.43 to 0.86 μm, and 115 bands in total, each pixel (with 610 × 340 pixels) has 1.3 m spatial resolution. After removing 12 bad bands, 103 bands were



(a) color image (b) ground truth

Fig. 2. Indian Pines data and ground truth.



(a) color image (b) ground truth

Fig. 3. Pavia University data and ground truth.

available. Figure 3(a) shows the RGB image of Pavia University, with red, green and blue bands of 170, 95 and 9 respectively. Figure 3(b) is the corresponding ground truth map, with a total of 42776 samples from 9 different categories of land cover.

4.2 Experimental Analysis for Feature Extraction Based on CNN

In the feature extraction experiment, the size of input samples is $1 \times n_1$, where n_1 is the number of bands. The CNN architecture consists of five layers, as listed in Sect. 3.

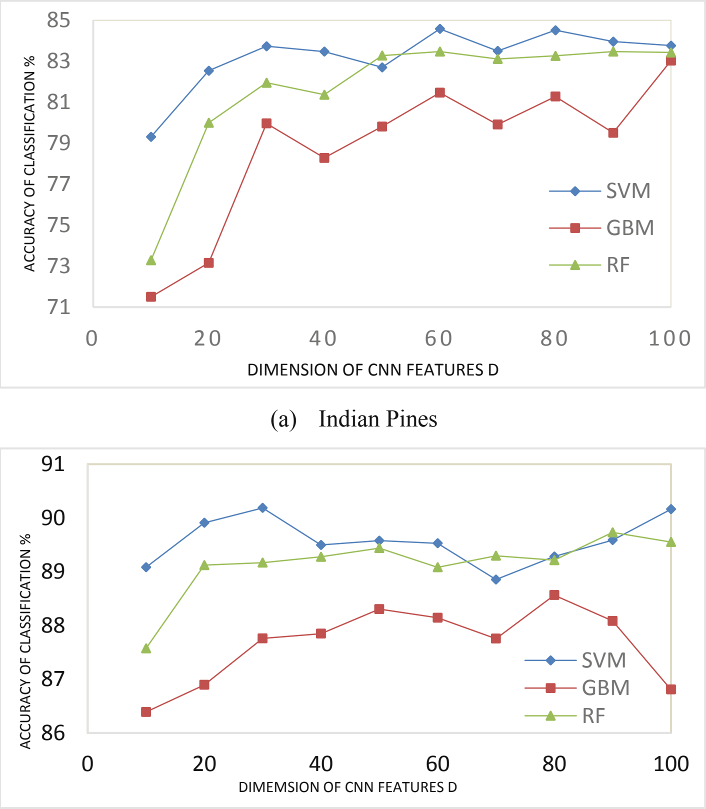
From the experimental results in Table 1 and Table 2, we can see that the classification accuracy based on CNN with only 20 features performs best compared with PCA, LDA feature extraction methods and raw data. At the meantime, the classification accuracy generally presents a linear increase trend with the increase of the number of features, as shown in Fig. 4. Feature extraction based on deep network is more consistent with the characteristics of hierarchical abstraction and layer-by-layer cognition of human vision, and performs better class discrimination. Experimental results in this section demonstrate that the features extracted by CNN can be used for the classification task, and the obtained classification results are competitive with other shallow models.

Table 1. Overall classification accuracy based on different FE methods of Indian Pines

OA (%)	SVM	GBM	RF
PCA/(30 D)	72.34	71.71	70.65
LDA/(15 D)	83.6	76.45	76.74
Raw/(200 D)	78.12	74.28	75.35
CNN/(20 D)	82.54	73.16	80.00
CNN/(30 D)	**83.73**	**79.97**	**81.95**

Table 2. Overall classification accuracy based on different FE methods of Pavia University

OA (%)	SVM	GBM	RF
PCA/(30 D)	85.21	82.82	79.38
LDA/(8 D)	86.95	87.36	87.36
Raw/(103 D)	89.77	81.89	82.21
CNN/(20 D)	89.91	86.89	89.12
CNN/(30 D)	**90.19**	**87.76**	**89.17**



(a) Indian Pines

(b) Pavia University

Fig. 4. The relationship between the classification accuracy of CNN feature extraction and the feature dimension (D) for Indian Pines and Pavia University data

4.3 DMTri-Trainning Experimental Results

In order to verify the classification accuracy of the proposed method, we compared and analyzed the results of three classifiers, SVM, GBM and RF, which are taken as initial three classification models in DMTri-training algorithm. As for Indian pines data, we select randomly 1000 samples as training set, the rest of samples as test set. As for Pavia University data, we select randomly 500 samples as training samples and the remaining 42276 samples as test set.

Table 3. The classification accuracy of different classifiers of Indian Pines data

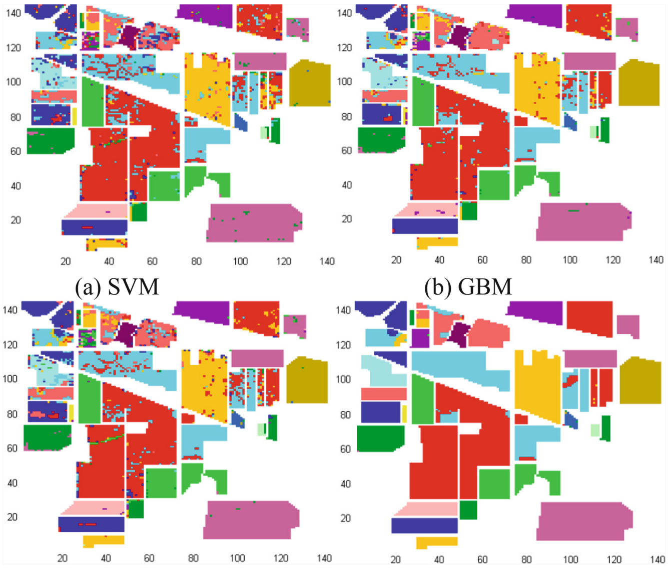
Category	SVM	GBM	RF	DMTri-training
Alfalfa	76.09	52.17	47.83	42.50
Corn-notill	80.46	81.79	82.98	90.27
Corn-mintill	78.31	70.48	72.89	93.72
Corn	65.40	46.84	46.41	97.86
Grass-pasture	88.82	87.58	87.78	98.96
Grass-trees	97.67	96.44	98.08	100.00
Grass-pasture-mowed	89.29	67.86	50.00	100.00
Hay-windrowed	98.95	98.33	99.79	100.00
Oats	55.00	15.00	25.00	30.77
Soybean-notill	79.12	72.12	74.07	89.44
Soybean-mintill	82.97	84.81	83.75	98.61
Soybean-clean	80.61	78.75	80.78	100.00
Wheat	96.10	85.37	90.24	100.00
Woods	95.34	95.49	96.68	100.00
Buildings-Grass-Trees-Drives	62.18	60.88	63.21	83.38
Stone-Steel-Towers	93.55	97.85	97.85	100.00
OA(%)	84.44	82.60	83.48	**95.79**
κ	0.82	**0.80**	0.81	0.94

In Table 3 and Table 4, we list the overall classification accuracy and class-specific classification accuracy corresponding to different algorithms based on CNN features for the two data sets. As for Indian Pines data, the overall accuracy is 84.44%, 82.60% and

Table 4. The classification accuracy of different classifiers of Pavia University data

Category	SVM	GBM	RF	DMTri-training
Asphalt	88.10	91.39	92.50	90.09
Meadows	95.57	96.44	96.37	99.68
Gravel	75.04	67.46	70.75	83.09
Trees	88.87	93.93	93.60	97.88
Painted metal sheet	99.18	99.33	99.26	100.00
Bare Soil	89.56	86.90	88.27	98.13
Bitumen	80.83	71.13	75.11	99.62
Self-Blocking Bricks	84.90	85.36	86.56	97.85
Shadows	99.79	99.58	99.89	100.00
OA(%)	91.05	91.36	92.03	**96.93**
κ	0.89	0.90	0.91	**0.97**

83.48% based on SVM, GBM and RF classifier. The corresponding kappa coefficient is 0.82, 0.80 and 0.81. The overall accuracy and kappa coefficient of DMTri-training is 95.79% and 0.94 respectively, improve 11.35% and 12.2% respectively compared to SVM classification, the best results among the three classifiers. Four classification maps are illustrated in Fig. 5. As for Pavia University data set, we can see that RF classification accuracy is the highest one. The overall classification accuracy is 91.05%, 91.36% and 92.03% based on SVM, GBM and RF respectively, and the corresponding kappa coefficients are 0.89, 0.90 and 0.91. The classification accuracy of DMTri-training algorithm is 96.93%, and the kappa coefficient is 0.97. Compared with the best RF classification result, the overall accuracy is improved by 4.9%, and the kappa coefficient is improved by 6.38%. The classification maps of different classification methods are listed in Fig. 6.



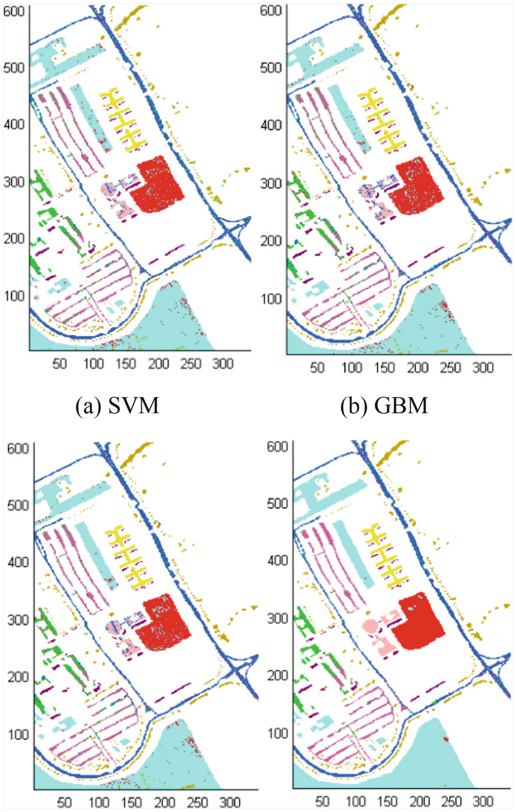
(a) SVM

(b) GBM

(c) RF

(d) DMTri-training

Fig. 5. The classification maps of Indian pines data based on different classifiers.



(a) SVM

(b) GBM

(c) RF

(d) DMTri-training

Fig. 6. The classification maps of Pavia University data based on different classifiers

5 Conclusion

In this paper, we proposed a semi-supervised classification method named DMTri-training for hyperspectral images classification. The proposed framework is composed of two parts, the abstracted feature exaction are based on CNN model and DMTri-training classification is based on modified tri-training algorithm and data editing. The experimental results show that our method can perform better classification results.

Acknowledgement. This work is supported in part by the National Science Foundation of China under Grant No. 61672276, the National Key Research and Development Program of China under Grant No. 2017YFB1400600, Jiangsu Natural Science Foundation of China under Grant No. BK20171037, the Program of Natural Science Research of Jiangsu colleges and universities under Grant No.17KJB170010, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

References

1. Ji, S., Ye, J.: Generalized linear discriminant analysis: a unified framework and efficient model selection. IEEE Trans. Neural Networks **19**(10), 1768–1782 (2008)
2. Fejjari, A., Ettabaa, K.S., Korbaa, O.: Fast spatial spectral Schroedinger Eigenmaps algorithm for hyperspectral feature extraction. Procedia Comput. Sci. **126**, 656–664 (2018)
3. Fang, Y., Xu, L., Peng, J., et al.: Unsupervised Bayesian Classification of a Hyperspectral Image Based on the Spectral Mixture Model and Markov Random Field. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **11**(9), 3325–3337 (2018)
4. Mountrakis, G., Im, J., Ogole, C.: Support vector machines in remote sensing: a review. ISPRS J. Photogrammetry Remote Sens. **66**(3), 247–259 (2011)
5. Alam, F.I., Zhou, J., Liew, A.W.C., et al.: CRF learning with CNN features for hyperspectral image segmentation. In: 36th IEEE International Geoscience and Remote Sensing Symposium, pp. 6890–6893. Institute of Electrical and Electronics Engineers Inc, Piscataway, USA (2016)
6. Qian, Y., Ye, M., Zhou, J.: Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features. IEEE Trans. Geosci. Remote Sens. **51**(4), 2276–2291 (2013)
7. Chen, Y., Lin, Z., Zhao, X., et al.: Deep learning-based classification of hyperspectral data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **7**(6), 2094–2107 (2014)
8. Tao, C., Pan, H., Li, Y., et al.: Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. IEEE Geosci. Remote Sens. Lett. **12**(12), 2438–2442 (2015)
9. Li, C., Wang, Y., Zhang, X., et al.: Deep belief network for spectral–spatial classification of hyperspectral remote sensor data. Sensors **19**(1), 204 (2019)
10. Yue, J., Zhao, W., Mao, S., et al.: Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. Remote Sens. Lett. **6**(6), 468–477 (2015)
11. Liang, H., Li, Q.: Hyperspectral imagery classification using sparse representations of convolutional neural network features. Remote Sens. **8**(2), 99 (2016)
12. Yang, J., Zhao, Y.Q., Chan, J.C.W.: Learning and transferring deep joint spectral–spatial features for hyperspectral classification. IEEE Trans. Geosci. Remote Sens. **55**(8), 4729–4742 (2017)
13. Li, J., Huang, X., Zhang, L.: Semi-supervised sparse relearning representation classification for high-resolution remote sensing imagery. In: 36th IEEE International Geoscience and Remote Sensing Symposium, pp. 2618–2621. Institute of Electrical and Electronics Engineers Inc, Piscataway, USA (2016)
14. Erkan, A. N., Camps-Valls, G., Altun, Y.: Semi-supervised remote sensing image classification via maximum entropy. In: IEEE International Workshop on Machine Learning for Signal Processing pp. 313–318. IEEE Computer Society, Washington DC, USA. (2010)
15. Yang, L., Yang, S., Jin, P., et al.: Semi-supervised hyperspectral image classification using spatio-spectral Laplacian support vector machine. IEEE Geosci. Remote Sens. Lett. **11**(3), 651–655 (2014)
16. Camps-Valls, G., Marsheva, T.V.B., Zhou, D.: Semi-supervised graph-based hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **45**(10), 3044–3054 (2007)
17. Ren, G., Zhang, J., Ma, Y., et al.: Generative model based semi-supervised learning method of remote sensing image classification. J. Remote Sens. **6**, 1090–1104 (2010)
18. Prabukumar, M., Shrutika, S.: Band clustering using expectation–maximization algorithm and weighted average fusion-based feature extraction for hyperspectral image classification. J. Appl. Remote Sens. **12**(4), 046015 (2018)

19. Zhou, Z.: Disagreement-based semi-supervised learning. ACTA AUTOMATICA SINICA **39**(11), 1871–1878 (2013)
20. Aydav, P.S.S., Minz, S.: Granulation-based self-training for the semi-supervised classification of remote-sensing images. Granul. Comput. 1–19(2019)
21. Samiappan, S., Moorhead, R.J.: Semi-supervised co-training and active learning framework for hyperspectral image classification. In: 35th IEEE International Geoscience and Remote Sensing Symposium, pp. 401–404. Institute of Electrical and Electronics Engineers Inc, Milan, Italy (2015)
22. Ou, D., Tan, K., Du, Q., et al.: A novel tri-training technique for the semi-supervised classification of hyperspectral images based on regularized local discriminant embedding feature extraction. Remote Sens. **11**(6), 654 (2019)