Digging More in Neural World: An Efficient Approach for Hyperspectral Image Classification Using Convolutional Neural Network

Adnan Iltaf[1], Matee Ullah[1], Junling Shen[1], Zebin Wu[1],
Chuancai Liu[1](✉), and Zeeshan Ahmad[2]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China
khsaani@qq.com, {mateeullahkhan,wuzb,
chuancailiu}@njust.edu.cn, sjl02468@163.com

[2] School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China
engr.zeeshan@hotmail.com

Abstract. Classification of hyperspectral images (HSI) can benefit from deep learning models with deep architecture in remote sensing. In this letter, a novel method based on Convolutional Neural Network (CNN) is proposed for the classification of hyperspectral images. Due to using more spatio-spectral features for the classification of hyperspectral images, the proposed method outperforms the existing state-of-the-art classification techniques. Our proposed method first reduces the dimension of hyperspectral images using Principle component analysis (PCA). The spatial and spectral features are then exploited by a fixed size convolutional filter to generate the combine spatio-spectral feature maps. Finally, these feature maps are fed into a Multi-Layer Perceptron (MLP) classifier that predicts the class of the pixel vector. To validate the effectiveness of our proposed method, computer simulations are conducted using three datasets namely Indian Pines, Salinas and Pavia University and comparisons with existing techniques are made.

Keywords: CNN · Hyperspectral classification · PCA
Multi-Layer Perceptron (MLP) · Remote sensing

1 Introduction

Hyperspectral image classification is an important research topic in remote sensing. In the presence of commercial hyperspectral sensors e.g. Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), HSI data is easily available to researchers. AVIRIS which is operated by the NASA Jet Propulsion Laboratory covers 224 continuous spectral bands across the electromagnetic spectrum with a spatial resolution of 3.7 m. The information collected by AVRIS is used to classify the objects on earth surface. Supervised or unsupervised classification algorithms have the ability to quickly obtain categorical information from remote-sensing images and classify the objects present in the image.

H. Yuan et al. (Eds.): GSKI 2017, CCIS 849, pp. 117–126, 2018.
https://doi.org/10.1007/978-981-13-0896-3_12

Consequently, such algorithms play an important role in remote-sensing image applications.

The basic purpose of image classification is to classify the labels for each pixel in HSI image, which is a challenging task. The performance of classification techniques is closely affected by high dimensionality of the data, limited labeled samples and spatial variability of spectral information. To overcome such issues, various techniques, such as independent component analysis (ICA) [1], neighborhood preserving embedding [2], linear discriminant analysis (LDA) [3] and wavelet analysis [4], have been proposed for the classification of hyperspectral images. Investigations show that the afore-mentioned techniques did not bring significant improvement in classification accuracy. However, support vector machine(SVM) based methods and Neural networks(NN) present a more attractive solution to image classification in terms of computational cost and classification accuracy [5]. Due to the high diversity of HSI data, it is difficult to determine which feature is more relevant for the classification task.

Moreover, recently introduced deep learning (DL) models automatically learn high-level features from data in a hierarchical manner. Typical deep learning models includes Deep Belief Networks [6], Deep Boltzmann Machines [7], Stacked Denoising Autoencoders [8] and Convolutional Neural network (CNN) [9]. More specifically Autoencoders (AE) [10] has been an efficiently used for the classification of HSI images, basically the input of Autoencoders (AE) is high dimensional vector i.e. flatten the high dimensional image into a vector then feed it to the model later classify it by using logistic regression classifier. A recent state-of-the-art technique proposed by Lee et al. [11], called a contextual deep CNN, consist of nine layers in total, jointly obtained the spatio-spectral features maps and classified by Softmax activation function.

In a similar fashion inspired by [11], in this paper we try to assess the effectiveness of a DL technique namely, Convolutional Neural network (CNN). The basic motivations for us to consider Convolutional approach have two main reasons: the effectiveness of this approach recently proved in numerous remote sensing applications; main characteristics of this technique, which makes it a potential candidate to classify hyperspectral data. In this context, we proposed a Conventional Multi-Layer Perceptron (MLP) network for the classification of remote sensing hyperspectral data. Our proposed structure basically combines the spectral-spatial attributes in initial stage resulting in a high-level spectral-spatial features construction and then implement MLP classifier for probabilistic multiclass HSI classification.

The rest of the paper is organized as follows: In Sect. 2, we provide details of the proposed network. The description of datasets and performance comparison are given in Sect. 3. Finally, Sect. 4 summarizes the process and some probable future work is pointed out.

2 Proposed Architecture

In this section architecture of the proposed system is briefly described. In the first stage the reduction of dimensionality is presented and then the deep structure of CNN and MLP is described.

2.1 Dimensionality Reduction

Usually, HSI data consist of several band/channels along the spectral dimension. Thus, it always has tens of thousands of dimensions resulting in a large amount of redundant information. In most of the cases, the first few band/channels have significant variance and they contain almost 99.9% of information [12]. So in the first layer of our proposed network we introduced PCA, to reduce the dimension to an acceptable scale while reserving the useful spatial information in the meantime. As our main concern is to incorporate the spatial information, so we use PCA along-with the spectral dimension only and retain first several principal components. During our experimentation process on state-of-the-art hyperspectral datasets, we used only 10 to 30 principal components respectively for each dataset.

2.2 Classification Framework

For CNN, Image input data is expressed as a 3-dimensional matrix of width * height * channels (h * w * c). In order to input an HSI image, we have to decompose HSI into patches, each one of which contains spectral and spatial information for a specific pixel. Our proposed network contains 12 convolutional layers. First convolutional layer in network contains 32 features with a filter whose dimension is 3 * 3. The batch size of 30 samples is used and the block size is set to 11. In first convolutional layer, we use a filter of dimension 3 * 3 and get feature maps in subsequent layers as shown in Fig. 1. In a similar manner for further layers filter size remains same but the number of feature maps is increased. For preserving local spatio-spectral correlation we do not increase the filter size. The first convolutional layer is followed by further hidden layers in the network.

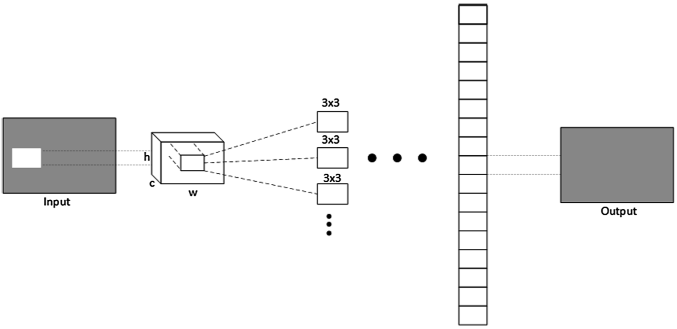

Fig. 1. Filter size and feature map representation.

During the training, network parameters keep changing repeatedly which cause a change in activations, this refers to as "internal covariate shift". To resolve this problem we adopt Batch normalization (BN) [13] which allows us to use much higher learning rate.

Algorithm:

```
Input: Values of x over a mini-batch: $\beta = \{x_1 \ldots x_m\}$
Parameters to be learned: $\gamma, \beta$
$Output: \{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$
$\mu_\beta \leftarrow \frac{1}{m}\sum_{i=0}^{m} x_i$    (1)   // mini − batch mean
$\sigma_\beta^2 \leftarrow \frac{1}{m}\sum_{i=0}^{m} (x_i - \mu_\beta)^2$    (2)   // mini − batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \in}}$    (3)   // normalizaiton
$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$    (4)   // Scale and Shift
```

The algorithm given above presents Batch normalization (BN) transforms where $\beta = \{x_1 \ldots x_m\}$ are the values over mini-batch. Equation (3) implements normalization operation while Eq. (4) implements scaling and shifting learned by γ and β parameters to get the final result y_i. The main characteristic of BN is that it is based on simple differentiable operations, which can be inserted anywhere in CNN network to normalize improper network initialization. BN boost up the performance as well.

After convolving the image fed the neurons to max-pooling layer, the purpose is to take the maximum values from the input and shorten the size of selected features. The pool size is 2 * 2. Next, pooling layer is followed by the Flatten layer which converts the 2D matrix to a vector called Flatten. It allows the output to be processed by standard fully connected layers. ReLU (Rectified linear unit) and dropout are also employed here. The threshold value for dropout is 0.3. The purpose of using ReLU is that it is much faster than other nonlinear functions and Dropout is used to prevent overfitting and complex co-adoptions phenomena.

For classification purpose Softmax activation [14] function issued to output probability-like predications according to the number of classes. Softmax is a generalization of logistic function, and its output can be used to represent the categorical distribution, which is basically a gradient-log-normalizer:

$$p\left(y = j|z^{(i)}\right) = \phi_{soft\ \max}\left(z^{(i)}\right) = \frac{e^{z^{(i)}}}{\sum_{j=0}^{k} e^{z_k^{(i)}}} \tag{5}$$

where z is the net input can be defined as

$$z = w_0x_0 + w_1x_1 + \ldots + w_mx_m = \sum_{l=0}^{m} w_lx_l = \boldsymbol{w}^T\boldsymbol{x} \tag{6}$$

where w is the weight vector, w_0 is for bias and x is the feature vector. $z^{(i)}$ is basically a classification function of $j - th$ class which takes "x" as an input and compute probability "y" for each class label. Therefore, Softmax is adopted here because it is a potential candidate for probabilistic multiclass HSI classification problem.

Stochastic gradient descent (SGD) is a classical approach for training deep learning architecture is employed here. SGD algorithm is used to calculate the error and propagate it back to adjust the MLP weights and filters. The architecture of our proposed approach is presented in Fig. 2.

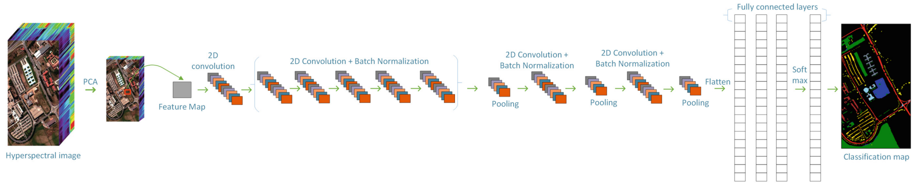

Fig. 2. Graphical representation of proposed network for HIS classification.

3 Experimental Results and Comparative Analysis

3.1 Datasets

AVIRIS and ROSIS sensor datasets are the classical datasets [15]. Particularly, in our experiment the Indian Pines, Salinas and Pavia university datasets are used. Indian Pines dataset depicts a test site in North-western Indiana and consists of 145 * 145 pixels with 224 spectral reflectance bands in the wavelength range from 0.4 to 2.5 μm while spatial resolution is 20 m. Basically, it contains 16 classes but we only use 8 classes because they have a large number of samples among others.

The University of Pavia dataset depicts the scenes acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy whose number of spectral bands are 102 contains 610 * 340 pixels. It contains 9 classes.

The number of spectral bands and spatial resolutions are 103 and 1.3 m respectively. While the spectral reflectance range from 0.4 to 0.8 μm.

Third dataset "Salinas" is also acquired by AVIRIS sensor over Salina Valley, California. It consists 224-bands with 512 * 217pixels with high spatial resolution 3.7 m. Number of classes of this data set are 16. For both datasets (University of Pavia, Salinas) we use all the classes for training and testing because they have a relatively large number of samples. For all datasets, selected classes and samples are listed in Tables 1, 2 and 3.

Table 1. Number of training and testing samples along with selected classes used from the Indian Pines DataSet.

ID	Class	Train	Test
1	Corn-notill	200	1234
2	Corn-mintill	200	634
3	Grass-pasture	200	297
4	Hay-windrowed	200	289
5	Soybean-notill	200	768
6	Soybean-mintill	200	2268
7	Soybean-clean	200	414
8	Woods	200	1094
Total		**1600**	**6998**

Table 2. Number of training and testing samples along with selected classes used from the University of Pavia DataSet.

ID	Class	Train	Test
1	Asphalt	200	6431
2	Meadows	200	18449
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**	**40976**

Table 3. Number of training and testing samples along with selected classes used from salinas dataset.

ID	Class	Train	Test
1	Broccoli greenweeds 1	200	1809
2	Broccoli-greenweeds 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	11071
9	Soil vineyard develop	200	6003
10	Corn-green weeds	200	3078
11	Lettuce romaine, 4wk	200	868
12	Lettuce romaine, 5wk	200	1727
13	Lettuce romaine, 6wk	200	716
14	Lettuce romaine, 7wk	200	870
15	Vineyard untrained	200	7068
16	Vineyard vertical trellis	200	1607
Total		**3200**	**50929**

3.2 Comparative Analysis

For comparison, we randomly select 200 samples per class for training and all remaining samples for testing. The basic purpose of selecting 200 samples per class is to evaluate our proposed method with the state of the art approaches reported in [11]. To successfully accomplish all the experiments the CNN Tensor flow framework [16] is used on GPU GTX1060.

Table 4 provides a comparative analysis of classification among the proposed method and the one reported in [11]. The contextual deep CNN used in [11] has 9 convolutional layers while our proposed network has twelve layers, we can say that our network is much deeper than contextual deep CNN [11]. It is obvious that our network has much better performance as compare to contextual deep CNN on all datasets. To further evaluate our network we compare our performance with state-of-the-art RBF kernel-based SVM method [17], which consist two convolutional and two fully connected layer much shallower than our technique. In recent research [18], for a diversified Deep Belief Networks(D-DBN) has much better performance as compared to [17], we also use (D-DBN) as a baseline to in our comparative analysis. For all the datasets, we also use other types of methods which are evaluated in [11]: two-layer NN, three-layer NN, shallower CNN and LeNet-5.

Our proposed network out-performs the baseline approaches on all the datasets. More specifically as compared to [11] for Indian Pines dataset the proposed network gained more than 2% accuracy while in the cases of University of Pavia and Salinas datasets, it gained 1.3% and 2.04% classification accuracy respectively. The significant performance of proposed architecture is just because of its deeper nature which proves, that digging more in the convolutional network leads to high classification accuracy. Figure 3 shows the classification maps of each data set corresponding to their ground truth images.

3.3 Impact of Epochs

During network training weights are updated due to back propagation phenomena, One round of updating the network or the entire training dataset is called an epoch [19]. Figure 4 shows validation loss and classification accuracy on the bases of epoch size. From validation loss plotted in Fig. 4a we observe the performance of the proposed network i.e. the number of lost samples decreased when the number of epochs increased meanwhile the classification accuracy is improved significantly as can be seen in Fig. 4b.

For all the data sets these observations proved that deepness of our network greatly improves overall accuracy meanwhile preserving lower validation loss.

Table 4. Classification accuracy comparison among proposed networks and the base lines on three datasets(%). The best performances among all methods are indicated in bold

Data sets	Two-Layer NN [11]	RBF-SVM [11]	Three-layer NN [11]	LeNet-5 [11]	Shallower CNN [11]	D-DBN [18]	DCNN [11]	Proposed Network
Indian Pines	86.49	87.60	87.93	88.27	90.16	91.03	93.61	**96.13**
University of Pavia	–	90.52	–	–	92.56	93.11	95.97	**97.32**
Salinas	–	91.66	–	–	92.60	–	95.07	**97.11**

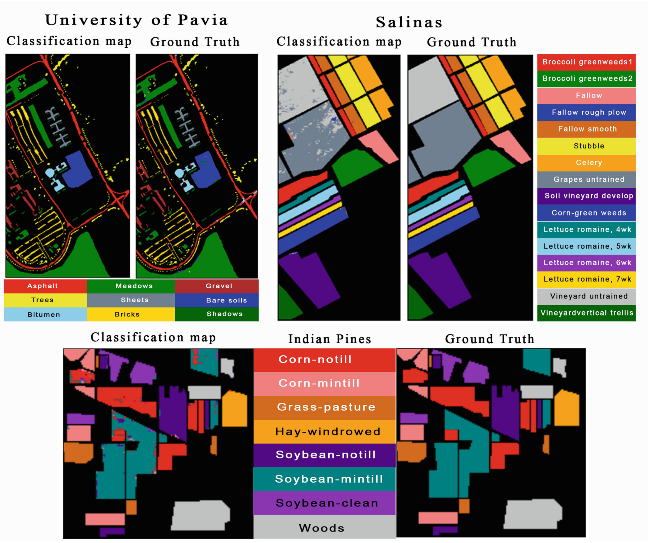

Fig. 3. RGB compositions of resulted classification map by proposed network along with ground truth are shown for University of Pavia, Salinas and Indian Pines datasets.

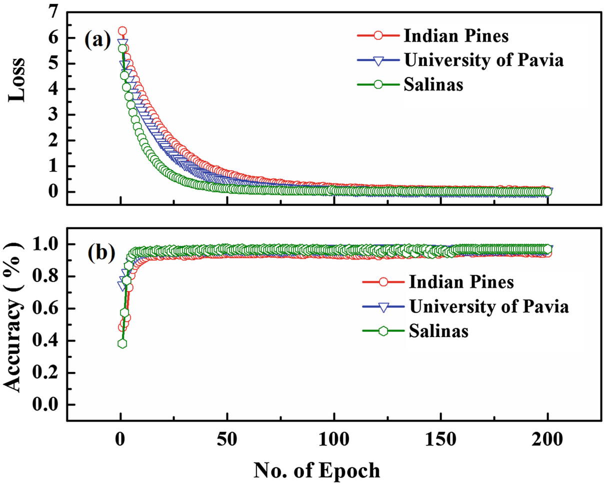

Fig. 4. Classification performance under different set of parameters for all experimental data sets. (a)Validation loss vs. Number of epochs, (b) Classification accuracy over the course of epochs.

4 Conclusion

In this letter, we propose a CNN-based classification method for remote sensing data. The proposed method is much deeper, faster and utilizes more spatio-spectral features for the classification of hyperspectral images. The proposed method and existing state-of-art techniques are compared using three data sets. It is shown that our method achieves better classification accuracy. Simulation results demonstrate the superiority of the proposed method. The future research prospects include to combine the proposed network with a shallower convolutional based network for more enhanced classification performance.

Acknowledgments. This work is sponsored by the National Natural Science Foundation of China under Grant No. 61373063 and 61373062; the project of Ministry of Industry and Information Technology of China (Grant No. E0310/1112/02-1).

References

1. Falco, N., Bruzzone, L., Benediktsson, J.A.: A comparative study of different ICA algorithms for hyperspectral image analysis. In: 2013 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), pp. 1–4 (2013)
2. Zhao, L.Y., Zou, D., Gao, G.: Subsampling based neighborhood preserving embedding for image classification. In: Proceedings - 2013 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2013, pp. 358–360 (2013)
3. Yuan, H., Tang, Y.Y., Lu, Y., Yang, L., Luo, H.: Spectral-spatial classification of hyperspectral image based on discriminant analysis. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **7**, 2035–2043 (2014)
4. Gangodagamage, C., Foufoula-Georgiou, E., Brumby, S.P., Chartrand, R., Koltunov, A., Liu, D., Cai, M., Ustin, S.L.: Wavelet-compressed representation of landscapes for hydrologic and geomorphologic applications. IEEE Geosci. Remote Sens. Lett. **13**, 480–484 (2016)
5. Yu, H., Gao, L., Liao, W., Zhang, B., Pizurica, A., Philips, W.: Multiscale superpixel-level subspace-based support vector machines for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **14**, 2142–2146 (2017)
6. Chen, Y., Zhao, X., Jia, X.: Spectral-Spatial classification of hyperspectral data based on deep belief network. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **8**, 2381–2392 (2015)
7. Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: AISTATS, pp. 448–455 (2009)
8. Vincent, P., Larochelle, H.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion Pierre-Antoine manzagol. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
9. Yu, S., Jia, S., Xu, C.: Convolutional neural networks for hyperspectral image classification. Neurocomputing **219**, 88–98 (2017)
10. Lin, Z., Chen, Y., Zhao, X., Wang, G.: Spectral-spatial classification of hyperspectral image using autoencoders. In: 2013 9th International Conference Information, Communication Signal Process, pp. 1–5 (2013)
11. Lee, H., Kwon, H.: Going deeper with contextual CNN for hyperspectral image classification. IEEE Trans. Image Process. **26**, 4843–4855 (2017)
12. Jablonski, J.A.: Reconstruction error and principal component based anomaly detection in hyperspectral imagery. Master thesis, Air Force Institute of Technology, USA (2014)
13. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on International Conference on Machine Learning, pp. 448–456 (2015)

14. Raschka, S.: Michigan State Uni., USA. https://www.kdnuggets.com/2016/07/softmax-regression-related-logistic-regression.html
15. Hyperspectral remote sensing scenes. http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
16. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., Brain, G.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016), pp. 265–284 (2016)
17. Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H.: Deep convolutional neural networks for hyperspectral image classification. J. Sens. **2015**, 1–12 (2015)
18. Zhong, P., Gong, Z., Li, S., Schonlieb, C.-B.: Learning to diversify deep belief networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **55**, 3516–3530 (2017)
19. Brownlee J.: Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras, 1.7th edn. Machine Learning Mastery, Melbourne (2016)