

# Encoding Spectral and Spatial Context Information for Hyperspectral Image Classification

Xin Sun<sup>1</sup>, Member, IEEE, Fei Zhou, Junyu Dong<sup>2</sup> Member, IEEE, Feng Gao<sup>1</sup>, Quanquan Mu, and Xinhua Wang

**Abstract**—Hyperspectral image (HSI) classification is a popular yet challenging research topic in the remote sensing community. This letter attempts to encode both spectral and spatial information into deep features for HSI classification. We first propose a semisupervised method for training the stacked autoencoder to obtain discriminative deep features. A batch training scheme is introduced to constrain the label consistency on a neighborhood region. Second, a mean pooling procedure is suggested to further fuse the spectral and local spatial information for deep feature generation. The experimental results on two hyperspectral scenes show that the proposed method achieves promising classification performance.

**Index Terms**—Hyperspectral image (HSI), pattern classification, semisupervised learning, stacked autoencoders (SAE).

## I. INTRODUCTION

WITH the advance of spectroscopy technology, hyperspectral sensors can simultaneously acquire hundreds of narrowband spectral channels. They can provide rich spectral information for object recognition tasks. Generally, pixels of different land-cover materials have different spectral features. Therefore, many research works employ the spectral features as powerful and discriminative information for object detection. However, one crucial problem is that land-cover materials of the same object can exhibit different spectral features, such as asphalt, bricks, and shadows. Fig. 1 shows some examples from the Pavia University (PaviaU) data set. We can see that the distribution of 103 spectral features for the same material is quite different in some cases. It can confuse the classifiers to make the correct decision. Another important information from the hyperspectral image (HSI) is the spatial context features. Since pixels of the same scene in the HSI behave as blocks, the spatial neighbor pixels can provide supporting information for classification. Compared with the spectral features, spatial features are extracted at the contextual region rather than pixel level [4]. Therefore, the spatial features can be used to distinguish the confused pixels.

Manuscript received June 14, 2017; revised September 6, 2017; accepted September 30, 2017. Date of publication October 26, 2017; date of current version December 4, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61401413 and Grant 41576011, in part by the Key Research and Development Program of Shandong Province under Grant GG201703140154, and in part by the Open Funding of State Key Laboratory of Applied Optics. (Corresponding author: Junyu Dong.)

X. Sun, F. Zhou, J. Dong, and F. Gao are with the Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: sunxin@ouc.edu.cn; dongjuyun@ouc.edu.cn).

Q. Mu and X. Wang are with the State Key Laboratory of Applied Optics, Chinese Academy of Sciences, Changchun 130033, China.

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LGRS.2017.2759168

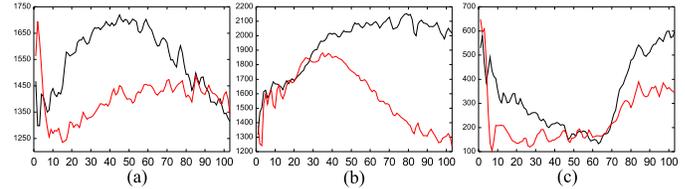


Fig. 1. Distribution of 103 spectral features for some materials. (a) Asphalt. (b) Bricks. (c) Shadows.

Some research works have already combined the spatial and spectral features together for classification, and achieved better performance than using spectral features only [4], [13]. For example, textures of Gabor filters are used to extract the spatial information in the first several principal components [5]. Graph cut is also investigated to form spatial information for HSI classification [11]. Moreover, a semisupervised graph-based method is utilized to incorporate spatial information into spectral features. Fauvel *et al.* [3] introduced morphological profiles to extract the spectral–spatial feature. Currently, spectral–spatial feature extraction has become a popular and effective way for HSI classification.

HSI classification is still a challenging task [14]. Another main difficulty comes from the unbalance of the high dimensionality and limited number of labeled data [1]. To solve the problem, a variety of feature extraction methods have been introduced to reduce redundant spectral information such as PCA, ICA [9], manifold [12], and wavelet analysis [8]. Unfortunately, all above methods just bring limited promotion, in some cases even worse than the direct classification approaches [2]. Deep neural networks (DNNs) have achieved great success in many domains, especially in pattern recognition. The superior performance of DNNs comes from the deep structure which can extract high-level features from original data [10]. Chen *et al.* [2] proposed a framework of training stack autoencoders (SAE) to learn spectral–spatial features. With the learned features, logistic regression or support vector machine (SVM) achieved nice performance. However, in their framework, as the input of SAE is a vector, the image patches have to be flattened into one dimension, which may lose the original spatial structure. It needs long time to fine-tune the whole model according to the accuracy of the classifier. In contrast, we first propose a semisupervised method for the autoencoder to learn separable features without the help of subsequent classifier. The semisupervised training procedure of autoencoder is conducted in a batch learning way,

which enforces the local spatial information into the model. Moreover, a mean pooling approach is designed to fuse the spectral and spatial information for feature generation.

This letter is organized as follows. Section II briefly introduces the SAE. In Section III, we present the proposed method, and Section IV shows the experimental results on real hyperspectral data sets. In Section V, we summarize this letter.

## II. BRIEF INTRODUCTION OF SAE

Shallow autoencoder is a three-layered network with symmetrical structure, which can map input  $x \in R^d$  to a hidden representation  $h \in R^k$  and reconstruct the hidden representation to obtain a reconstruction close to the input in an unsupervised way. In order to avoid linearity, the nonlinear function  $f(\cdot)$  is applied for mapping, e.g., the sigmoid function. The process can be described mathematically as follows:

$$h = f(w_h x + b_h) \quad (1)$$

$$z = f(w_z h + b_z) \quad (2)$$

here:  $w_h$  and  $b_h$  are weights and bias of the input to hidden layer, respectively;  $w_z$  and  $b_z$  of the hidden to output layer. To evaluate the performance of the reconstruction from  $z$  to  $x$ , various distance metrics can be implemented, such as mean squared error (MSE) cross entropy. In this letter, we choose the MSE as cost function  $c_a(x, z)$ . Our goal is to minimize the distance, that is

$$\arg \min_{w, b_h, b_z} c_a(x, z) \quad (3)$$

where  $w$  is called tied weights that means  $w_h = w_z^T$ ; the minimization process can be achieved via backpropagation and stochastic gradient descent [5].

Stacked autoencoder (SAE) is the stack of many autoencoders by making the last hidden representation as input of the following one. The greedy strategy is employed to train the network in layer-wise manner. In this letter, we will also investigate the reason why SAE can improve the classification task for HSI, and introduce a semisupervised SAE (Semi-SAE) method.

## III. SEMISUPERVISED SAE FOR SPECTRAL-SPATIAL CLASSIFICATION

### A. Investigation of SAE Deep Features

SAE has the ability to preserve the most important components for reconstruction. In this section, we first investigate the features of SAE in a visualized way, and then propose a Semi-SAE to improve classification performance.

At first, we want to know how SAE learns deep features. The autoencoder can be regarded as a nonlinear feature dimension reduction approach. In order to visualize the SAE deep features and the intrinsic structure of HSI data, we can set the number of neurons of the last layer as 2. Fig. 2 shows the visualization results. We can see that data is mapped to a two dimension feature space. Meanwhile, in Fig. 2(a)–(c), we can see that, as the network goes deeper and deeper, the samples of the same category become much closer to each other.

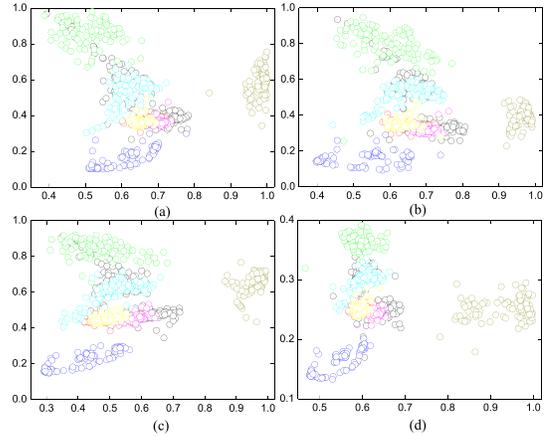


Fig. 2. Two dimension visualization for deep SAE features. (a) Two layers. (b) Three layers. (c) Four layers. (d) Two layers of Semi-SAE.

However, samples which are far away from the center or of complicated distribution will fall into other clusters. This phenomenon motivates us a simple idea that we can reduce the scatter of the same labeled samples in feature space in order to achieve the higher classification accuracy. For the HSI classification task, we prefer the discriminative feature more. So we propose a semisupervised SAE method named Semi-SAE. And Fig. 2(d) shows the 2-Dimension visualized the results of only two layers of Semi-SAE, where the samples of the same category exhibit higher clustering tendency. Section IV will also show the good performance for classification. In the following section, we will describe how our Semi-SAE works.

### B. Semisupervised SAE

Our goal is to train a deep model for discriminative features mapping. As the labels of a small number of the samples are known, we could enforce the partial label information into the training procedure. To achieve the goal, a new label consistency constraint, named discriminative error, is proposed as follows:

$$c_s = \sum_{i,j=1}^N s_{ij} \|h_i^{(k)} - h_j^{(k)}\|_2^2 \quad (4)$$

where  $N$  is the size of samples,  $k$  is the number of layers,  $h$  is the latent representation, and  $s_{ij}$  denotes the relationship between sample  $i$  and  $j$ .  $S$  is the adjacency matrix and each entry  $s_{ij}$  is defined as follows:

$$\begin{cases} s_{ij} = 1 & y_i = y_j \\ s_{ij} = 0 & y_i \neq y_j \end{cases} \quad (5)$$

where  $y$  is the label of sample  $x$ . To accelerate the training procedure, we introduce the batch-based training method for the discriminative ability learning. Another reason for introducing the batch training is that samples are pixels from HSI. Therefore, we can send the samples from a small region to the SAE together in order to better catch the local discriminative information. We divide the HSI into many overlap small areas

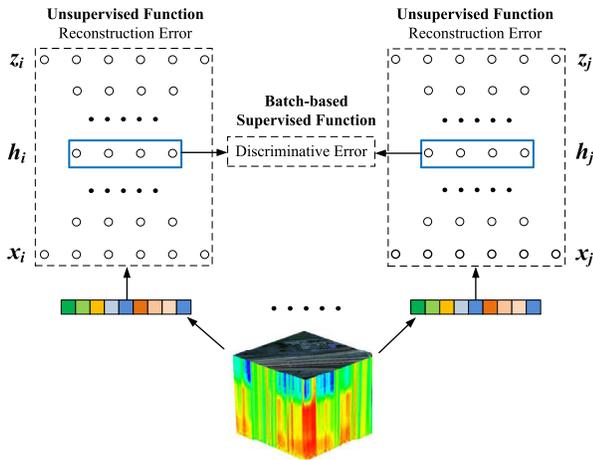


Fig. 3. Framework of batch-based Semi-SAE training, where  $x_i$  and  $x_j$  are two pixels in HSI,  $h_i$  and  $h_j$  are the hidden representation of  $x_i$  and  $x_j$ , respectively,  $z_i$  and  $z_j$  are the reconstruction of  $x_i$  and  $x_j$ .

each of which contains  $n \ll N$  neighborhood samples. The discriminative error for each batch is defined as follows:

$$c_s = \sum_{i,j=1}^n s_{ij} \|h_i^{(k)} - h_j^{(k)}\|_2^2. \quad (6)$$

And  $c_s$  can be reform as

$$c_s = \sum_{i,j=1}^n s_{ij} \|h_i^{(k)} - h_j^{(k)}\|_2^2 = 2\text{tri}(H^T L H) \quad (7)$$

where  $H = \{h_i\}_{i=1}^n$ ,  $L = D - S$ ,  $D \in R^{n \times n}$  is a diagonal matrix  $D_{i,i} = \sum_j s_{i,j}$ .

Then, we combine the discriminative error  $c_s$  with reconstruction error  $c_a$  to form a unified objective function for SAE optimization. Therefore, we can jointly minimize the final objective function  $c_{\text{joint}}$

$$c_{\text{joint}} = c_a + \frac{1}{N_{\text{Batch}}} \cdot c_s \quad (8)$$

where  $N_{\text{Batch}}$  is the number of samples in each batch. Fig. 3 shows the framework of our Semi-SAE training. We first use the deep belief network to initialize the weights of SAE. As both labeled and unlabeled data are used to train our model, the method can be regarded as a Semi-SAE.

### C. Mean Pooling

Neighborhood pixels can be powerful and discriminative spatial context information. For example, Chen *et al.* [2] flatten the neighbor region into one vector to combine the spatial context information for classification. However, the dimension of spectrum should be reduced at first in order to avoid the curse of dimensionality. Besides, it is easy to steal the classifiers attention from the main pixel to the neighbor pixels, and may lose the original spatial structure. In this letter, we introduce a mean pooling way for feature generation procedure. Mean pooling is a method that takes all the spectral-spatial information of a flat neighbor region. Specifically, mean pooling averages the value of a  $z \times z$  size neighbor region of every

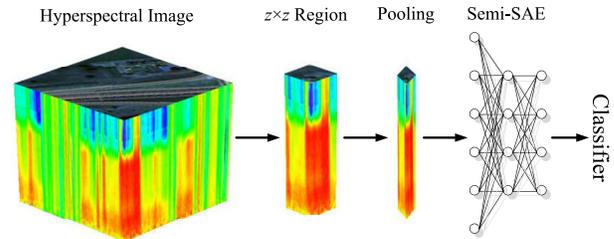


Fig. 4. Flowchart of spectral-spatial deep features extraction.

TABLE I  
CLASSES AND NUMBERS OF PIXELS IN PAVIAU DATA SET

#	Name	Training	Test
1	Asphalt	650	5189
2	Meadows	1894	16755
3	Gravel	221	1878
4	Trees	287	2777
5	Painted metal Sheets	134	1211
6	Bare Soil	486	4543
7	Bitumen	121	1209
8	Self-Blocking Bricks	381	3301
9	Shadows	111	836
Total		4285	38941

spectral channel into one vector, as shown in Fig. 4. Then the averaged vector is used as the input of trained Semi-SAE to generate deep features. Finally, a classifier, e.g., SVM, can take the deep features for classification.

### D. Discussion

This letter tries to extract deep features combining spectral and spatial information for HSI classification. As commonly known, spatial context information is the crucial complementarity for spectral features. The key point is how to fuse spectral and spatial information effectively. We introduce two strategies: batch-based semisupervised training and mean pooling-based feature generation. For HSI data, a sample is the spectral features of one pixel from the HSI image, and the samples of one category form dense blocks in the image. Commonly the misclassified pixels are edges among different blocks. Besides accelerating the training process, batch-based semisupervised training can separate the fine-grained confusing samples. The samples that are located in a small neighborhood are sent together as batch to train the Semi-SAE, which makes the Semi-SAE easy to distinguish the samples from each other. Another problem, as shown in Fig. 1, is the quite different feature distribution of the same category. The reason of such phenomena might be that the material of the same land cover can be different, e.g., bricks. As mentioned that pixels of same category are distributed as blocks, mean pooling gives a possible way to smooth the features of the same block. In Section IV, we will show the performance of these two strategies, respectively.

## IV. EXPERIMENTS

In this section, we mainly used two popular data sets to verify the effectiveness of the proposed method. The first one

TABLE II  
CLASSES AND NUMBERS OF PIXELS IN INDIAN PINES DATA SET

#	Name	Training	Test
1	Alfalfa	4	42
2	Corn-notill	137	1291
3	Corn-mintill	90	740
4	Corn	21	216
5	Grass-pasture	51	432
6	Grass-trees	71	659
7	Grass-pasture-mowed	3	27
8	Hay-windrowed	45	433
9	Oats	2	18
10	Soybeans-notill	110	862
11	Soybean-mintill	246	2209
12	Sybean-clean	51	542
13	Wheat	22	183
14	Woods	134	1131
15	Buildings-grass-trees	43	343
16	Stone-still-towers	13	80
	Total	1043	9208

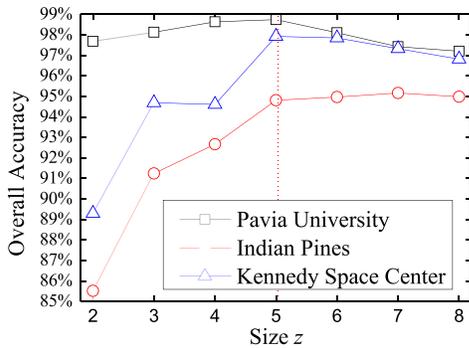


Fig. 5. Accuracy against different size of pooling region.

is the PaviaU data set with nine different land-cover classes as shown in Table I, which has  $610 \times 340$  pixels in size and 103 spectral channels ranging from  $0.43\text{--}0.86 \mu\text{m}$ . The second is the Indian Pines data set of 16 different land-cover classes as shown in Table II, which has  $145 \times 145$  pixels in size and 200 spectral channels ranging from  $0.4\text{--}2.5 \mu\text{m}$ . Both of the two data sets are divided into two parts, the labeled and unlabeled. We choose randomly 10% of labeled as the train data, and the rest labeled samples are used for testing. Moreover, another commonly used Kennedy Space Center data set is also employed to analyze the parameter  $z$ .

First, we present an experimental analysis on the parameter  $z$  of mean pooling. We set  $z$  as 2, 3, 4, 5, 6, 7, and 8, respectively, in order to search the most suitable value of  $z$  for mean pooling. In Fig. 5, we can see that nice overall accuracies (OA) are achieved on all these data set in the case of  $z = 5$ . The reason includes two folds. The small size of neighbor region means insufficient spatial information, whereas the big size may import too much unnecessary information and noise, which makes the key pixel lose the dominant role for classification.

We empirically evaluate the performance of our proposed method by comparing with two typical HSI processing and classification methods: extended morphological profiles (EMP) [3] and EPF-G [6]. The EMP method builds extended morphological profiles by using morphology filter on several principal components, while EPF-G is a joint

TABLE III  
CLASSIFICATION RESULTS ON THE PAVIAU DATA SET

#	SAE	Semi-SAE	SAE*	Semi-SAE*	EMP	EPF-G	SVM
1	94.46	94.70	98.53	98.84	98.01	97.35	93.48
2	98.01	98.13	99.85	99.94	98.00	98.54	98.40
3	76.03	77.48	92.55	93.02	95.69	93.19	79.44
4	85.48	93.55	98.45	98.45	99.28	87.48	90.60
5	96.61	96.04	90.83	92.90	99.92	96.77	93.36
6	85.30	89.10	99.56	99.60	76.14	83.85	81.73
7	86.27	87.34	97.35	98.10	88.56	88.23	83.71
8	88.73	90.37	97.04	97.21	95.88	91.01	88.61
9	66.14	91.02	93.18	98.09	99.04	99.06	98.02
OA	92.08	93.97	98.41	98.73	95.00	93.86	91.09
K	87.52	89.27	97.83	98.25	92.40	91.33	88.48

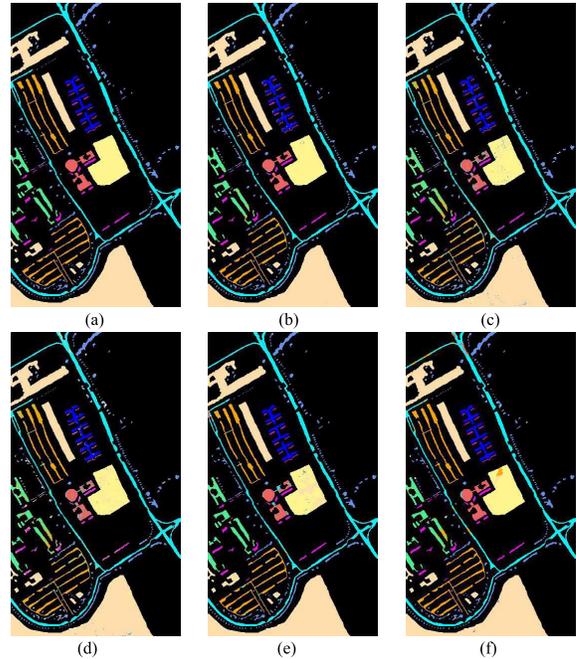


Fig. 6. Pseudo-color maps for the results of PaviaU scene. (a) Ground truth. (b) Semi-SAE\* (98.73%). (c) Semi-SAE (93.97%). (e) EMP (95.00%). (d) SAE (92.08%). (f) EPF-G (93.86%).

spectral-spatial hyperspectral classification method based on edge-preserving filters. The SAE method is currently popular for HSI feature extraction, and is also employed in our proposed method. Thus, it is essential to exhibit the performance comparison between SAE and our Semi-SAE method, in order to verify whether the proposed method can improve the performance of original SAE. In addition, we use SAE\* and our Semi-SAE\* denote the methods that employ mean pooling in the feature generation procedure. Except edge-preserving filtering-Gaussian kernel (EPF-G), all the methods are followed by a radial basis function-SVM classifier ( $c = 4$ ,  $g = 4$  for PaviaU data set and  $c = 10$ ,  $g = 0.2$  for India Pines data set) for classification. Its hyper-plane parameters have been adjusted via cross validation with `grid.py` of `libsvm` toolkit.

For the experiments on the PaviaU dataset, all of the SAE have the same structure of three layers (103-128-48), and Adam [7] is applied to train the networks (1000 epochs with learning rate 0.003). Table III records the OA (%) and Kappa coefficients of different methods on PaviaU data set.

TABLE IV  
CLASSIFICATION RESULTS ON THE INDIAN PINES DATA SET

#	SAE	Semi-SAE	SAE*	Semi-SAE*	EMP	EPF-G	SVM
1	76.19	57.14	69.05	76.19	71.42	95.85	19.05
2	73.10	78.59	93.11	96.15	82.80	93.95	72.73
3	72.33	75.63	94.59	93.27	89.05	96.25	73.24
4	66.38	66.37	89.35	90.50	75.46	67.00	37.96
5	94.22	95.11	98.37	90.95	89.35	98.17	92.59
6	94.91	94.63	98.18	97.43	94.08	97.97	97.40
7	92.85	92.86	74.07	100.0	88.89	100.0	14.81
8	97.88	99.15	98.61	99.75	98.15	99.99	99.07
9	63.63	45.45	88.89	50.00	100.0	99.14	50.00
10	78.39	83.03	95.24	94.18	92.80	80.85	75.75
11	85.83	86.73	97.76	97.63	94.52	95.32	83.46
12	77.22	80.04	84.13	96.16	71.22	87.23	64.94
13	96.87	97.97	99.45	100.0	98.91	100.0	98.36
14	95.16	95.13	99.73	99.45	97.35	99.25	94.69
15	66.66	67.19	95.33	100.0	72.59	78.80	58.60
16	82.05	79.49	96.25	96.20	65.00	87.36	80.00
OA	83.50	85.36	95.67	96.42	89.60	92.43	80.58
K	80.98	83.19	94.86	95.90	87.80	91.33	77.15

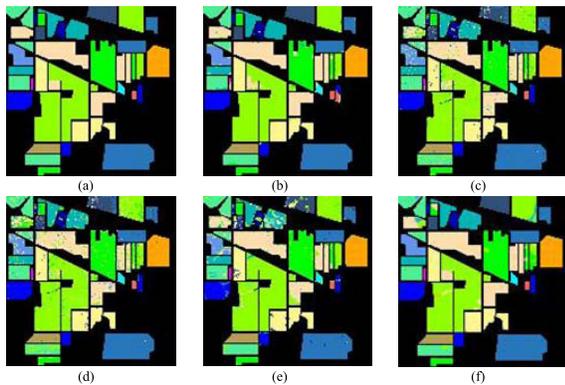


Fig. 7. Pseudo-color maps for the results of Indian Pines scene. (a) Ground truth. (b) Semi-SAE\* (96.42%). (c) Semi-SAE (85.36%). (d) SAE (83.50%). (e) EMP (89.60%). (f) EPF-G (92.43%).

From the SAE and Semi-SAE columns, we can see that our semisupervised training method performs better performance than the original SAE. Moreover the columns Semi-SAE and Semi-SAE\* show that the mean pooling significantly improves the performance. Obviously, Semi-SAE\* achieves much higher performance than others in most categories and the best performance on overall accuracy (98.73%) and Kappa coefficient (98.25).

For illustrative purpose, Fig. 6 shows some of the classification pseudo-color maps obtained after applying comparison methods to PaviaU scene. It can be observed that our method gives the best delineation of the complex urban structures.

Table IV records the classification accuracies of different methods on Indian Pines data set. The structure of SAE is defined as three layers (200-256-60). We can draw a similar conclusion that our method achieves the best performance. For example, our Semi-SAE\* method performs the best performance with 96.42% in overall accuracy. Compare with the original SAE, we obtain 1.89% improvement by utilizing semisupervised method to train our model in PaviaU data set, and the fact is also true for Indian pines data set. Mean pooling outperforms both of SAE and Semi-SAE more than 5% overall accuracy on both of the two data sets. It should be mentioned that the SAE only has two hidden layers and is not fully trained with only 1000 epochs. We could further improve the results by optimizing the network and training procedure.

Fig. 7 shows the classification pseudo-color maps of these comparison methods on Indian Pines scene.

## V. CONCLUSION

In this letter, we investigate why deep features from SAE improve the performance of classification. Then we try to extract discriminative deep features by combining both the spectral and local spatial information for HSI classification. We achieve the goal by proposing two strategies: batch-based semisupervised training and mean pooling-based feature generation. Besides accelerating the training process, batch-based semisupervised training constrains the label consistency on a neighborhood region, which makes the networks can fine-grained separate the confusing samples. The mean pooling further encodes the spectral and local spatial information for generating deep features. Our experimental results, conducted with two widely used hyperspectral scenes, show that our deep features achieve the best performance for HSI classification.

For the future work, we will further investigate the SAE deep features and well train the SAE network. Moreover, the local spatial information can be encoded in a more complex way. For example, the mean pooling scheme can be designed in a multiscaled way.

## REFERENCES

- [1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
- [2] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
- [3] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 11, pp. 3804–3814, Nov. 2008.
- [4] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral-spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.
- [5] L.-Z. Huo and P. Tang, "Spectral and spatial classification of hyperspectral data using SVMs and Gabor textures," in *Proc. Geosci. Remote Sens. Symp.*, Jul. 2011, pp. 1708–1711.
- [6] X. Kang, S. Li, and J. A. Benediktsson, "Spectral-spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [7] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May. 2015.
- [8] C. H. Koger, L. M. Bruce, D. R. Shaw, and K. N. Reddy, "Wavelet analysis of hyperspectral reflectance data for detecting pitted morningglory (*Ipomoea lacunosa*) in soybean (*Glycine max*)," *Remote Sens. Environ.*, vol. 86, no. 1, pp. 108–119, 2003.
- [9] S. Moussaoui *et al.*, "On the decomposition of Mars hyperspectral data by ICA and Bayesian positive source separation," *Neurocomputing*, vol. 71, nos. 10–12, pp. 2194–2208, 2008.
- [10] V. Sze, Y. H. Chen, T. J. Yang, and J. Emer, "Efficient processing of deep neural networks: A tutorial and survey," unpublished paper, 2017. [Online]. Available: <https://arxiv.org/abs/1703.09039>
- [11] Y. Tarabalka and A. Rana, "Graph-cut-based model for spectral-spatial classification of hyperspectral images," in *Proc. Geosci. Remote Sens. Symp.*, Jul. 2014, pp. 3418–3421.
- [12] Q. Wang, J. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, Jun. 2016.
- [13] Q. Wang, Z. Meng, and X. Li, "Locality adaptive discriminant analysis for spectral-spatial classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2077–2081, Nov. 2017.
- [14] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2966–2977, Dec. 2016.