Spectral–Spatial Classification of Hyperspectral Data Based on Deep Belief Network

Yushi Chen, *Member, IEEE*, Xing Zhao, *Student Member, IEEE*, and Xiuping Jia, *Senior Member, IEEE*

***Abstract*—Hyperspectral data classification is a hot topic in remote sensing community. In recent years, significant effort has been focused on this issue. However, most of the methods extract the features of original data in a shallow manner. In this paper, we introduce a deep learning approach into hyperspectral image classification. A new feature extraction (FE) and image classification framework are proposed for hyperspectral data analysis based on deep belief network (DBN). First, we verify the eligibility of restricted Boltzmann machine (RBM) and DBN by the following spectral information-based classification. Then, we propose a novel deep architecture, which combines the spectral–spatial FE and classification together to get high classification accuracy. The framework is a hybrid of principal component analysis (PCA), hierarchical learning-based FE, and logistic regression (LR). Experimental results with hyperspectral data indicate that the classifier provide competitive solution with the state-of-the-art methods. In addition, this paper reveals that deep learning system has huge potential for hyperspectral data classification.**

***Index Terms*—Deep belief network (DBN), deep learning, feature extraction (FE), hyperspectral data classification, logistic regression (LR), restricted Boltzmann machine (RBM), support vector machine (SVM).**

I. Introduction

HYPERSPECTRAL data can provide spatial and spectral information simultaneously [1], [2]. For this reason, hyperspectral data are used in a wide range of applications such as agriculture [3], mineralogy [4], surveillance [5], astronomy [7], and environmental sciences [9], [10]. Classification of each pixel in hyperspectral imagery is a common technique used in these applications. Because of the importance of classification, a large number of classification methods have been developed in the last two decades.

In the early stage of hyperspectral data classification, a lot of machine learning methods were introduced to solve the problem, and the typical algorithms include k-nearest neighbors, maximum likelihood, minimum distance and logistic regression (LR) [11], [12]. To deal with the Hughes effect [1], many feature reduction methods were proposed including the recent work on sparse representation [6], [8]. There are two kinds of these methods: 1) feature selection (FS) and 2) feature extraction (FE) [13], [14]. FS is to find a good subset of the original spectral bands [15]–[17], while FE is to seek a proper subset in a transformed feature space [18], [19]. In the classification stage, the classifiers use the reduced features instead of the original data to classify.

Support vector machine (SVM) is a powerful classification tool, which exhibits low sensitivity to high dimensionality. In [20], SVM was introduced for hyperspectral data processing. It has become a popular approach and improved classification accuracy compared with other widely used pattern recognition techniques [21], [22], [52]. For a long time, SVM-based classifiers have been the mainstream methods of classification [32].

In recent years, spatial information has been taken into account and some spectral–spatial-based classifiers have been proposed, and these methods provided significant advantages in terms of improving performance [23], [24]. In [25], the proposed method was based on the fusion of morphological information and original data followed by SVM. In [26], a new classification framework was proposed to exploit the spatial and spectral information using loopy belief propagation and active learning. In [27], spatial–spectral kernel sparse representation was proposed to deal with hyperspectral data classification.

The traditional classifiers like linear SVM and LR can be attributed to single-layer classifiers, while decision tree or SVM with kernels are believed to have two layers [28], [29]. Deep architectures with more layers, however, can potentially extract abstract and invariant features for better image or signal classification [30]. Based on neuroscience, human brains process information with multiple stages from retina to cortex, which leads to high performance on object recognition [31]. The applications of deep learning to image classification [32], [33], language processing [34] and speech recognition [35] have been actively studied in recent years. The classification of hyperspectral remote sensing images is a challenging task, due to the complex imaging conditions. In order to extract efficient features of hyperspectral data, it is important to investigate several deep architectures to benefit hyperspectral data classification.

In this paper, a new FE and image classification framework is proposed for hyperspectral data analysis based on deep belief network (DBN). Our work focuses on single-layer restricted Boltzmann machine (RBM) and multilayer deep network (DBN)-based models to learn the shallow and deep

Manuscript received July 31, 2014; revised November 27, 2014; accepted December 22, 2014. Date of publication January 22, 2015; date of current version July 30, 2015. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant HIT. NSRIF.2013028, and in part by the National Natural Science Foundation of China under Grant 61301206, Grant 61371180, and Grant 61471148). *(Corresponding author: Yushi Chen.)*

Y. Chen and X. Zhao are with the Department of Information Engineering, School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: chenyushi@hit.edu.cn; zhaoxing@hit.edu.cn).

X. Jia is with the School of Engineering and Information Technology, The University of New South Wales, Sydney, NSW 1000, Australia (e-mail: x.jia@adfa.edu.au).

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

Digital Object Identifier 10.1109/JSTARS.2015.2388577

features of hyperspectral data, respectively. The learnt features are then used in an LR to address the classification problem of hyperspectral data.

The main contributions of this paper can be summarized as follows: 1) We introduce DBN for hyperspectral data FE for the first time. DBN extracts the deep and invariant features of hyperspectral data, which will contribute to a reliable classification. 2) The original DBN has a requirement of one-dimensional data, while hyperspectral image is three-dimensional data. We address this problem and develop a new framework for 3-D hyperspectral image, which combines principal component analysis (PCA), hierarchical learning-based FE, with LR. 3) Three DBN-based deep learning architectures (DLAs) are proposed with spectral, spatial, and spectral–spatial features, respectively.

The rest of the paper is organized into four sections. Section II is a brief description of deep learning, RBM, and DBN models used in this paper. In Section III, we present three classification frameworks based on spectral features, spatial features, and spectral–spatial features, respectively. Experimental results with two hyperspectral data sets are shown in Section VI. Section V summarizes the observations and completes this paper by pointing out some probable future works.

II. Deep Learning, RBM, and DBN

A. Feature Learning and Deep Learning

Feature learning is a critical component of a classification system. The performance of classification is largely dependent on the learnt features [36]. For that reason, great effort has been made to extract effective features from original data. Especially, deep learning-based techniques have been developed to solve this challenging problem [30].

Deep learning is a kind of neural network which typically has more than three layers. Deep models can hierarchically extract the features of the data, and the learnt deep features are invariant to most local changes of the input. According to some recent publications, deep architectures achieve the state-of-the-art accuracy in many application areas such as object recognition [38] and natural language processing [39].

Typical deep neural network models include DBN [40], stacked autoencoder (SAE) [42], and deep convolutional neural networks (CNNs) [41].

Since the original DBN paper was published in Science [33], DBN has become one of the most important models of deep learning. It uses generative model in the pretraining procedure, and uses back-propagation in the fine-tuning stage [37]. This is very useful when the number of training samples is limited [37], such as the case in hyperspectral remote sensing. DBN is also a fast learning algorithm that can find the optimal parameters quicker [40]. In this paper, we investigate the effectiveness of DBN for hyperspectral data classification.

B. RBM

RBM is commonly used as a layer-wise training model in the construction of a DBN. It is a two-layer network, presenting

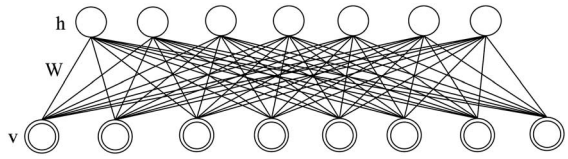


Fig. 1. Illustration of RBM. The top layer represents the hidden units and the bottom layer represents the visible units.

a particular type of Markov random filed with "visible" units $\mathbf{v} = \{0,1\}^D$ and "hidden" units $\boldsymbol{h} = \{0,1\}^F$ (Fig. 1). A joint configuration of the units has an energy given by

$$\begin{aligned} E(\boldsymbol{v},\boldsymbol{h};\theta) &= -\sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j - \sum_{i=1}^{D}\sum_{j=1}^{F} w_i v_i h_j \\ &= -\boldsymbol{b^T v} - \boldsymbol{a^T h} - \boldsymbol{v^T W h} \end{aligned} \tag{1}$$

where $\theta = \{b_i, a_j, w_{ij}\}$, w_{ij} is the weight between visible unit i, and hidden unit j; b_i and a_j are bias terms of visible and hidden unit, respectively. They are the model parameters.

The joint distribution over the units is defined by

$$P(\boldsymbol{v},\boldsymbol{h};\theta) = \frac{1}{\mathrm{Z}(\theta)}\exp(-E(\boldsymbol{v},\boldsymbol{h};\theta)) \tag{2}$$

$$Z(\theta) = \sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}} E(\boldsymbol{v},\boldsymbol{h};\theta) \tag{3}$$

where $Z(\theta)$ is the normalizing constant. The network gives a probability to every input vector via the energy function. The probability of the training vector can be raised by adjusting θ to lower the energy as given in (1).

The conditional distributions of hidden unit $\boldsymbol{h}$ and input vector $\boldsymbol{v}$ are given by logistic function

$$p(h_j = 1|\boldsymbol{v}) = g\left(\sum_{i=1}^{D} W_{ij} v_i + a_j\right) \tag{4}$$

$$p(v_i = 1|\boldsymbol{h}) = g\left(\sum_{j=1}^{F} W_{ij} h_j + b_i\right) \tag{5}$$

$$g(x) = \frac{1}{1+\exp(-x)}. \tag{6}$$

Once the states of hidden units are chosen, the input data can be reconstructed by setting each v_i to 1 with the probability of (5). The hidden units' states are then updated, so that they represent the features of the reconstruction.

The learning of W is done by a method called contrastive divergence (CD) [43]. The change in a weight is given by

$$\Delta w_{ij} = \epsilon\left(v_i h_{j\,\mathrm{data}} - v_i h_{j\,\mathrm{reconstruction}}\right) \tag{7}$$

where ϵ is a learning rate. Through the learning process, we can obtain proper value of W.

The power of RBM lies in the form of reconstruction oriented learning. During reconstruction, it only uses the information in hidden units, which is learnt as features from input. If the model can recover original input perfectly, it means that the hidden units retain enough information of the input, and the learned weights and biases can be deemed as good measures of the input data.

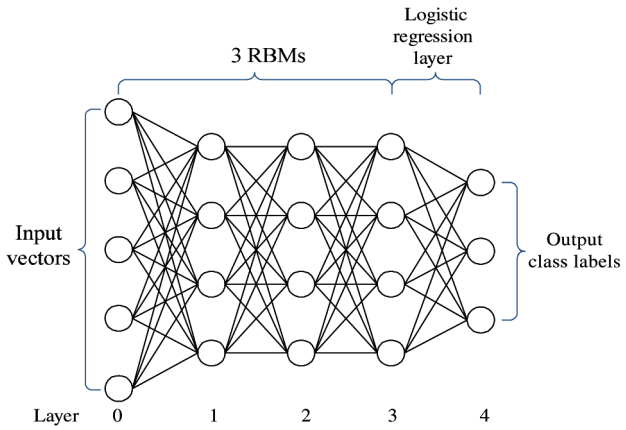


Fig. 2. Instance of a DBN connected with a LR layer. It has five layers: 1) one input layer; 2) three hidden layers; and 3) one output layer.

C. DBN

A single hidden layer RBM is not the best way to capture the features in the data. After the training of RBM, the learnt features can be used as the input data for a second RBM. This kind of layer-by-layer learning system can be used to construct DBN [45], [46]. In this way, DBN can progressively extract deep features of input data. That is to say, DBN learns a deep feature of input via pretraining in a hierarchal manner. Fig. 2 shows a typical instance of a DBN connected with a subsequent classifier.

The first RBM maps input data in zeroth-layer to a first-layer feature. It is trained in the same manner as aforementioned RBM. After the training, the first layer RBM is completed; subsequent layers of RBM are trained via the output of its previous layer. The features of the last RBM are the learnt features of the whole training system.

An LR layer is added to the end of feature learning system. This LR classifier is used to fine-tune the whole pretrained network to integrate the layers of neural networks and perform classification by utilizing the learnt features. The process of fine-tuning is back-propagation, searching for a minimum in a peripheral region of parameters initialized by DBN [47], [48].

III. Classification Frameworks Based on DBN

In this section, we develop three DBN-based frameworks for hyperspectral data classification, with spectral features, spatial features and spectral–spatial features, respectively.

A. Spectral Classification Framework

In this section, we propose a DLA for hyperspectral data classification with the pure spectral features.

Several existing approaches for hyperspectral data classification are shallow in their architectures, such as SVM, KNN, and maximum likelihood [11], [12]. Instead, we advocate a deep architecture in this paper. For hyperspectral data with complex characteristics, one single hidden layer usually would not be enough in describing the complicated relations between original data and the detailed class taxonomies. A deep architecture

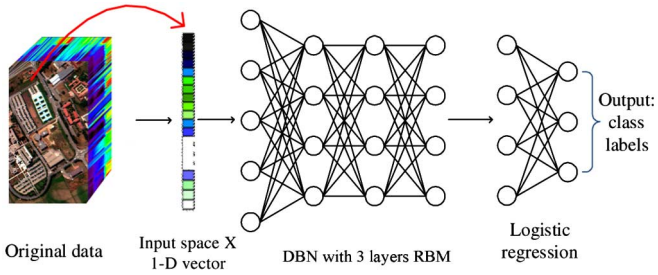


Fig. 3. Spectral classification using DBN-LR framework.

can show its advantage in dealing with these complicated relations. In addition, the deep architecture could learn features with as less prior knowledge as possible [49].

Here, we employ a DBN for unsupervised feature learning and add an LR layer above the DBN to constitute a DBN-LR framework. Training a deep multilayered neural network is actually difficult because the error gradient would explode or vanish as the number of layers increases. Recent work on deep learning has made deep neural network training more effective since Hinton's breakthrough in 2006 [33]. Then, we use an LR at the top layer in our approach so that we can perform supervised fine-tuning on the whole architecture easily [40].

Our deep architecture for hyperspectral data classification using the pure spectral features is illustrated in Fig. 3. Input space X is, generally, the raw spectral data collected as a one-dimesional (1-D) vector for each pixel. To make full use of the limited prior knowledge from a network perspective, we take the responses of all the spectral channels into the input space.

Unlike the binary RBM, as introduced in Section II-B, we replace it with real-valued units [30], [44], [50] that add Gaussian noise to model the input data. Energy function and conditional probability distributions are as follows:

$$E\left(\boldsymbol{v},\boldsymbol{h};\theta\right)=-\sum_{i=1}^{D}\frac{\left(v_i-b_i\right)^2}{2\sigma_i^2}-\sum_{j=1}^{F}a_jh_j-\sum_{i=1}^{D}\sum_{j=1}^{F}w_{ij}\frac{v_i}{\sigma_i}h_j \tag{8}$$

$$P(h_j|\boldsymbol{v};\theta)=g\left(\sum_{i=1}^{D}w_{ij}v_i+a_j\right) \tag{9}$$

$$P\left(v_j|\boldsymbol{h};\theta\right)=\mathrm{N}\left(b_i+\sigma_i\sum_{j=1}^{F}h_jw_{ij},\sigma_i^2\right) \tag{10}$$

where σ is the standard deviation of a Gaussian visible unit, and $N(\mu,\sigma^2)$ is the Gaussian distribution with mean μ and variance σ. Since the responses of all the spectral channels are used as input data, we have to regularize the model for sparsity [41]. We encourage each hidden unit to have a predetermined expected activation by a regularization penalty of the following form:

$$\lambda\sum_{j=1}^{F}\left(\rho-\frac{1}{m}\left(\sum_{k=1}^{m}E\left[h_j|\boldsymbol{v}^k\right]\right)\right)^2 \tag{11}$$

where ρ determines the sparsity and $\boldsymbol{v}^k$ is a sample in the training set. m is the number of training samples.

We can see each layer RBM layer in the DBN is a process of nonlinear feature transformation. Features learned in the top

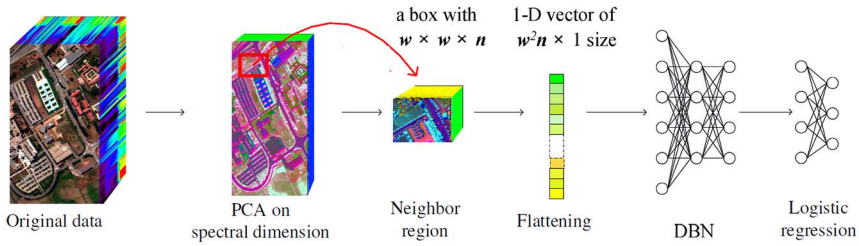


Fig. 4. Spatial classification using DBN-LR framework.

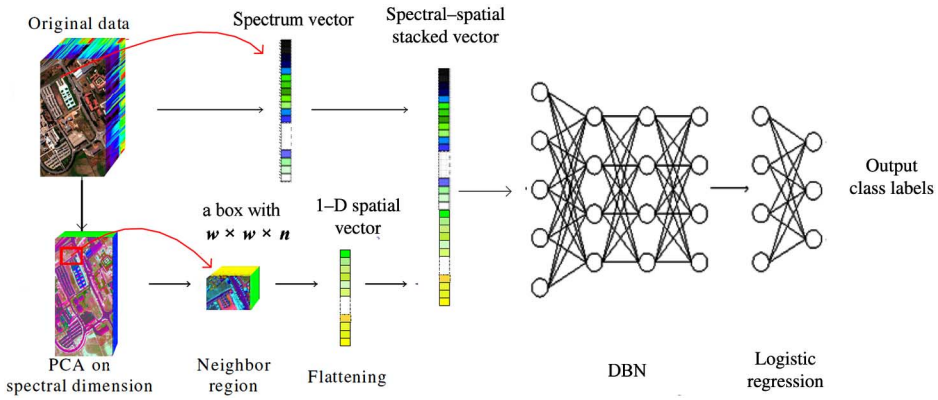


Fig. 5. Spectral–spatial classification using DBN-LR framework.

layer of the DBN are the most representative features for modeling the data. It can be denoted by $H_p = h_{p1}, h_{p2}, \ldots, h_{pn}$, where p represents the top layer, and n is the number of features in the top layer. The representative features are learned in an unsupervised way and can be used for various tasks, such as classification and regression. In our architecture, the most representative features H_p are used as the input vector for classification (the top LR layer). Moreover, since we employ LR in the top, the whole structure can be seen as a complete structure of a neural network. The top LR fine-tunes features learned from the DBN via error back propagation on the whole structure by using labeled samples. Then for the complete structure, the DBN is the feature learning model, and the LR layer is the classification model.

In summary, we first use the raw data of all the spectral channels as the input. Then, a DBN is applied to learn the representative and robust features from the inputs via several layers of nonlinear feature transformation to describe the complex mapping of inputs and features. Finally, an LR layer is used to produce the class labels from the features learned by the DBN.

B. Spatial Classification Framework

Aimed to make full use of the spatial information around each pixel's neighborhood, the proposed framework takes all the pixels in a flat neighbor region into consideration, and employs the DBNs to learn the features by itself. The flowchart of the proposed method is detailed in Fig. 4.

In the first step, PCA is conducted to reduce the data dimension to an acceptable scale and reserve spatial information. We use PCA along the spectral dimension and only retain the first several principal components (PCs). The PCA transformation matrix is fitted on the whole image, both for tagged and untagged pixels. This step does cast away part of the spectral information, but the spatial information is less affected. Due to the use of a few PCs instead of the hundreds of original spectral channels, it can prevent subsequent processes from producing tens of thousands of dimensions for the FE system (DBN).

Second, we extract a neighbor region around the labeled pixels in the condensed data, which has only several PCs in spectral dimension. For each pixel, there are $\boldsymbol{w} \times \boldsymbol{w}$ neighbor pixels with a region size of $\boldsymbol{w}$. Given the number of PCs is $\boldsymbol{n}$, a pixel can be represented as a box with. $\boldsymbol{w} \times \boldsymbol{w} \times \boldsymbol{n}$. members.

After these processes, we "flatten" the box, i.e., stretch it to a 1-D vector with $\boldsymbol{w}^2\boldsymbol{n} \times 1$ elements. Without any artificial FE and selection, 1-D vectors are then fed into a DBN. The subsequent includes a layer-wise pretraining DBN of and a fine-tuning the whole model with LR. These steps are similar to the previous section which deals with the spectral features.

C. Spectral–Spatial Classification Framework

In this section, we integrate the spectral and spatial features together to construct a spectral–spatial-based classification framework. The whole flowchart is shown in Fig. 5.

As discussed above, pure spectral features and spatial features both provide a discriminating power for the pixel-wise classification. The spectrum of a pixel contains important information for discriminating different kinds of ground categories. With spatial information, the statistics of the pixels in a neighbor region decreases of the intra-class variance which can lead to improved classification performance [51]. Taking into

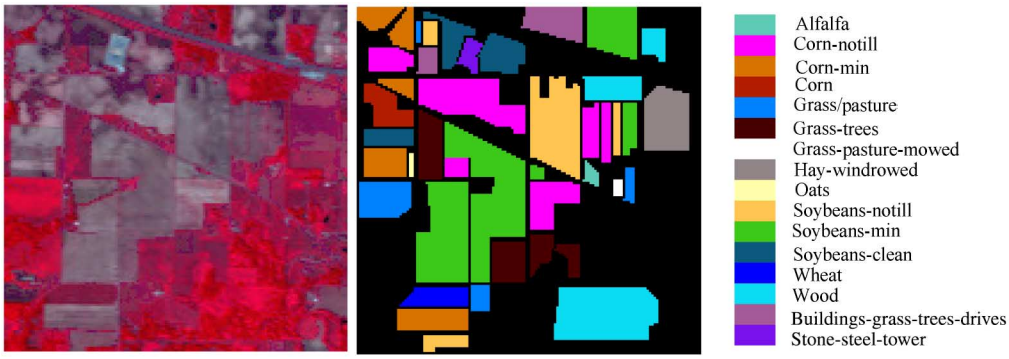


Fig. 6. AVIRIS the Indian Pines data set. False-color composite (Band 50, 27, 17) and representing 16 land-cover classes.

TABLE I
LAND-COVER CLASSES AND NUMBERS OF PIXELS IN THE INDIAN PINES DATA SET

Class Code	Name	No. of training samples	No. of testing samples	Class code	Name	No. of training samples	No. of testing samples
1	Alfalfa	23	23	9	Oats	10	10
2	Corn-notill	708	711	10	Soybean-notill	483	482
3	Corn-mintill	412	412	11	Soybean-mintill	1228	1222
4	Corn	119	118	12	Soybean-clean	296	295
5	Grass-pasture	241	241	13	Wheat	102	102
6	Grass-trees	364	363	14	Woods	631	629
7	Grass-pasture-mowed	14	14	15	Buildings-grass-trees	189	193
8	Hay-windrowed	235	239	16	Stone-steel-towers	45	46
Total		5100	5100				

account the different emphases, it is agreed that the complement of spectral and spatial features can present more reliable classification. The integration of multiple features is addressed by using a vector stacking (VS) approach in this study. That is to say, for each pixel, the 1-D vector processing in Section III-B (Fig. 4) is added to the end of the spectral vector. After forming a hybrid set of spectral–spatial features, we feed it into DBN-LR without any preprocessing of FE and selection. Following pretraining and fine-tuning steps similar to above, we can eventually assign a class label to each pixel.

IV. EXPERIMENTS AND RESULTS

A. Data Description and Experimental Setup

In our experiments, two hyperspectral data sets were applied to evaluate the proposed methods. They are a mixed vegetation site over the Indian Pines test site in North-Western Indiana (Indian Pines) and an urban site over the city of Pavia, Italy (Pavia University scene).

The Indian Pines data set was acquired by the airborne visible/infrared imaging spectrometer (AVIRIS). The image has a size of 145×145 pixels and 220 spectral bands in the wavelength range of $0.4 - 2.5$ μm. The false color composite image is shown in Fig. 6. The number of bands was reduced to 200 by removing the bands covering the region of water absorption. 16 different land-cover classes are available in the ground truth and the number of samples of each class is listed in Table I.

The second data set, Pavia data, was gathered by a sensor known as the reflective optics system imaging spectrometer (ROSIS-3) over the city of Pavia, Italy, with 610×340 pixels (Fig. 7). 115 bands were collected over $0.43 - 0.86$ μm range of the electromagnetic spectrum. The spatial resolution is 1.3 m. In the experiment, some bands were removed due to noise; the remaining 103 channels were used for the classification. Nine land-cover classes were selected, which are shown in Fig. 7 and the numbers of samples for each class are given in Table II.

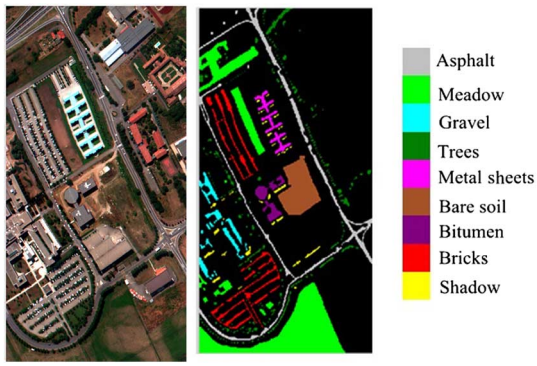


Fig. 7. ROSIS-3 data, Pavia, Italy. False-color composite (Band 10, 27, 46) and representing nine land-cover classes.

For evaluating the classification accuracy, labeled samples are randomly divided into training set and test set with a ratio of 1:1. So the processes of training and test share the same number of samples. The details are shown in Tables I and II. In order to investigate the performance of the proposed methods, experiments were organized step by step. The characteristic

TABLE II
LAND-COVER CLASSES AND NUMBERS OF PIXELS IN THE PAVIA DATA SET

Class code	Name	No. of training samples	No. of testing samples
1	Asphalt	3414	3419
2	Meadows	9316	9318
3	Gravel	1099	1102
4	Trees	1716	1712
5	Metal sheets	688	687
6	Bare soil	2540	2542
7	Bitumen	678	673
8	Bricks	1936	1936
9	Shadow	513	511
Total		21 900	21 900

of RBM was first examined. Classification with spectral features, classification with spatial features and classification with spectral–spatial features were conducted separately.

As SVMs with kernels have been widely used, this method was implemented for comparison in this study. To convincingly compare and estimate the capabilities of the proposed methods, for both SVM and DBN-LR, we run the experiments 20 times with different initial random training samples, and then confidence intervals (obtained by the mean and standard deviation) of overall accuracy (OA), kappa statistic, average accuracy (AA), and computational time are reported.

Furthermore, a paired t-test between SVM and DBN-LR was performed to validate whether the observed increase in the OA is statistically significant (at the confidence level of 95%) [51]. We use mean test to test whether the mean OA of our model ($\overline{\mathrm{a_1}}$) is higher than the mean OA of some control groups such as SVM ($\overline{\mathrm{a_2}}$). We accept the hypothesis of $\mathrm{a_1}$ being larger than $\mathrm{a_2}$ if and only if

$$\frac{(\overline{a_1}-\overline{a_2})\sqrt{n_1+n_2-2}}{\sqrt{\left(\frac{1}{n_1}+\frac{1}{n_2}\right)(n_1 s_1^2+n_2 s_2^2)}} > t_{1-\alpha}[n_1+n_2-2] \quad (12)$$

where $\mathrm{s_1}$ and $\mathrm{s_2}$ are the observed standard deviations for the two models, $\mathrm{n_1}$ and $\mathrm{n_2}$ are the number of realizations of experiments reported, and $\mathrm{t_{1-\alpha}}$ is the αth best quantile of the Student's law (typically $\alpha = 0.05$ is used) [51].

B. RBM: Characteristic and Analysis

In this section, the characteristic of RBM is investigated to indicate the validity for hyperspectral data classification.

1) Reconstruction of Spectral Curves With Hidden Units: First, we examine the quality of FE by checking the quality of the reconstructed spectral curve. We used single-layer RBMs with different numbers of hidden units (10, 50, 100, 150, and 200) and trained them on Indian Pines data. After the single-layer RBM learned with hundreds of iterating epochs, the reconstruction is computed with the hidden units (Section II-B). The reconstruction owns the same dimensionality with the original curve. Fig. 8 shows the reconstructions operated by RBMs. The signal-to-noise ratio (SNR) is computed in the condition that the difference between the original and the reconstruction data is regarded as noise. If the hidden units contain enough information of the input data, the reconstruction will be good. So, RBM can be thought as an effective FE method for hyperspectral data.

2) RBM's Filter Characteristics: The hyperspectral data has N bands, and RBM has N input neurons and H hidden neurons. The input-to-hidden layer of an RBM is fully connected, so every single hidden unit has its connections to every input neuron. For each hidden unit, it has N connection weights. The N connection weights can be viewed as a *filter*, by filtering away the content of certain wavelengths from input, and at the same time exaggerating others. So, an RBM with H hidden units can be viewed as H filters.

Aimed to make the filters visible, the N connection weights are horizontally folded to form a matrix $\mathbf{M}$. The matrix $\mathbf{M}$ has N entries and for the whole network, there are H matrixes. Each matrix $\mathbf{M}$ can be shown as a two-dimensional gray image, and the intensities of N entries are the connection weights. Hence, we can directly observe the interesting characteristics of these filters.

Fig. 9(a) and (b) shows the filters acquired after RBMs' learning on Indian Pines and Pavia data sets, respectively. The 1-D connection weights are folded into 20×10 and 13×8 matrices corresponding to the 200 and 103 input sizes of the Indian Pines and Pavia data. In Fig. 9(a), there are 10 hidden units in the trained RBM, thus 10 tiny filter images are in the plot. For the Pavia data, 40 hidden units are used.

The intensity of each pixel stands for the value of corresponding weights. Some hidden units have large weights over parts of input units and small weights over others, which suggest that the filters just care about a certain wavelength interval. The horizontal folding way that leads to the filters is trying to extract "horizontal" features, especially for Fig. 9(a). In addition, other weights have more complex connecting patterns, having ripples over different input units, or showing Gaussian-like noises in some bands.

C. Spectral Classification

After examining various characteristics of RBM, we exploit the classification potential of DLA by conducting it with spectral features. Here, we have explored the influence of depths (the number of RBMs' layers) on classification, the running time of DBN, as well as comparisons with SVM.

1) Influence of Depths: Depth plays an important role in the classification accuracy because it determines the quality of learned features from various aspects such as invariance and abstraction. In consideration of the computational complexity, the depth also affects the running time of the proposed method.

Here, several DBN-LRs with different depths were conducted. For the Indian Pines data, which has 200 spectral bands and 16 land-cover classes, the number of hidden units for each hidden layer is set to 30. The neural networks were constructed as a framework with 200-30-$\cdots$-30-16 units. For the Pavia data set, the neural networks were 103-50-$\cdots$-50-9. "Depth" corresponds to the number of 30- or 50-sized layers in the DLA. In this part of the experiment, the RBMs were

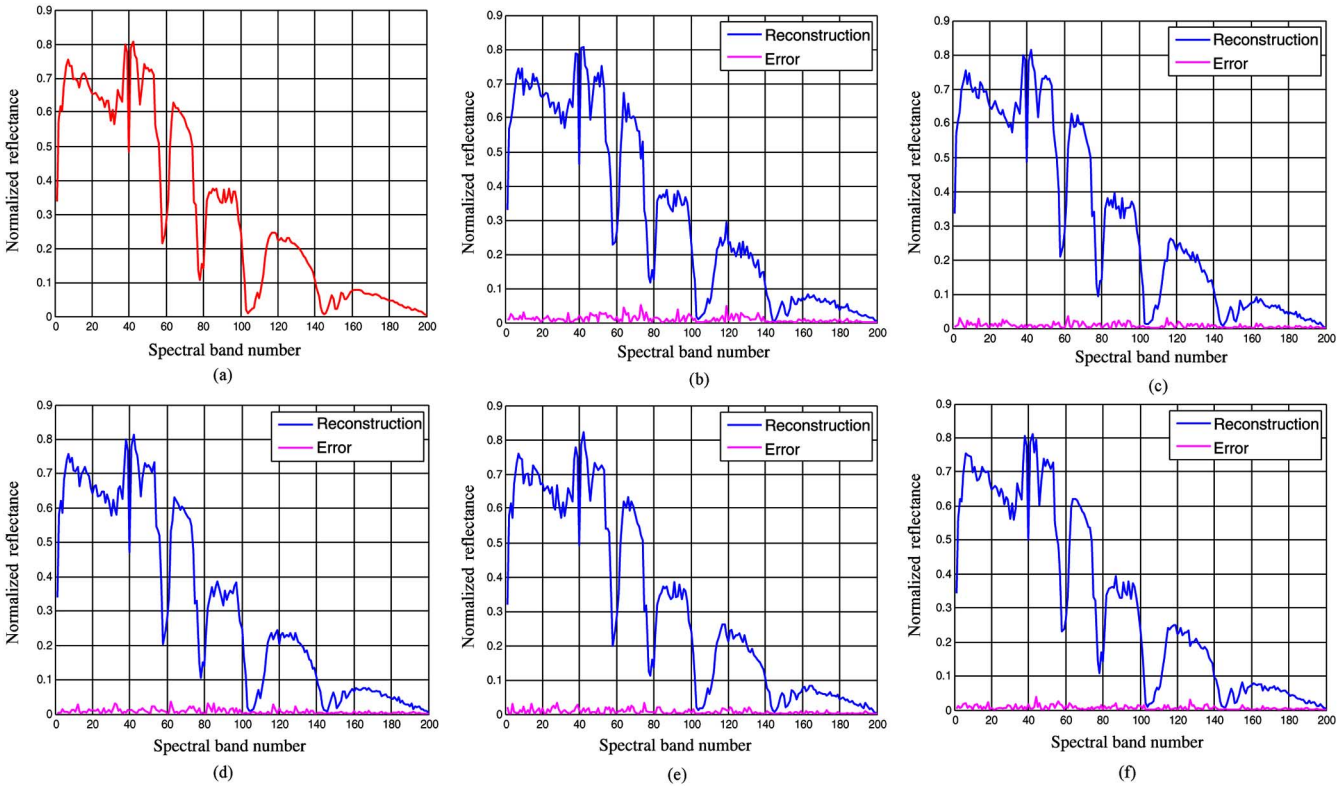


Fig. 8. Reconstructions operated by RBMs with different numbers of hidden units for the class of Soybean-mintill in the Indian Pines data. (a) The original curve. (b)–(f) Reconstructions of (a) with 10, 50, 100, 150, and 200 hidden units, respectively.

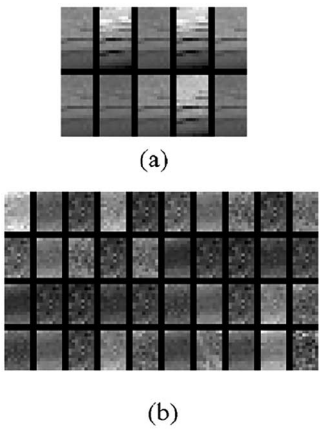


Fig. 9. Filter images learned by an RBM on (a) Indian Pines and (b) Pavia. Each N-pixel tiny rectangle stands for N input to hidden weights that connect each input unit to a same hidden unit.

tuned, with only 1000 epochs of pretraining and 5000 epochs of fine-tuning.

Tables III and IV show the classification results, training time, and test time of DBNs with different depths. Note that the best accuracy of each case is highlighted by the shaded part. Experiments show that depth does help to improve classification accuracy. However, given the characteristic of hyperspectral data, too deep will act inversely. The best depths are 2 and 4 for two data sets, respectively. The depth has a significant influence on the classification accuracy. If the depths are set improperly, the performance of DBN-LR preforms even no better than SVM.

2) Running Time of DBN: Generally speaking, deep learning-based methods take longer time to train the models compared with other machine learning algorithms such as SVM. However, deep learning algorithms can be implemented into parallel version with little modification when facing a large scale data set.

Tables III and IV show the running time of different methods. We can see that DBN-LRs indeed cost more time on the training stage, but they are super-fast on testing time (classification of unknown data). The super-fast classification stage is a great advantage when large hyperspectral images are processed.

In the experiments, we use NVIDIA GT770M graphics card to make the training procedure faster. However, the DBN codes used was a basic design, instead of a professional version. In the realization of SVM-based hyperspectral data classification, we use LibSVM [54] as a toolbox, which is a fast implementation of SVM.

3) Comparative Experiments With SVMs: SVMs are one of the state-of-the-art classifiers for hyperspectral data. Aimed to test the performance of the proposed DBN-LR method, the comparison with SVM models were conducted. The SVM parameters, a regularization parameter $\boldsymbol{c}$ and a kernel parameter g (for the RBF kernel), were determined by a grid search using cross validation.

The results are shown in Tables III and IV. With regard to the classification performance, the proposed DBN-LR outperforms SVM in most of the cases. For the Pavia data set, when

TABLE III
CLASSIFICATION WITH SPECTRAL FEATURE ON DBN-LR AND SVM ON INDIAN PINES DATA SET THE MEAN VALUES ± STANDARD DEVIATION

Classifier	Depth	OA (%)	AA (%)	Kappa statistic	Training time (s)	Test time (s)
DBN-LR	1	91.04 ± 0.2835	88.63 ± 0.6699	0.8912 ± 0.0032	33.5040 ± 0.4442	0.0001 ± 0.0000
	2	**91.34 ± 0.2679**	**89.70 ± 0.3408**	**0.9013 ± 0.0031**	40.0800 ± 0.9628	0.0140 ± 0.0000
	3	90.82 ± 0.2389	87.22 ± 0.6595	0.8953 ± 0.0027	45.6720 ± 0.4900	0.0155 ± 0.0000
	4	90.56 ± 0.3485	86.03 ± 0.3171	0.8924 ± 0.0040	47.0160 ± 0.5476	0.0173 ± 0.0001
	5	89.19 ± 0.6073	85.34 ± 0.6477	0.8881 ± 0.0070	50.8680 ± 0.5186	0.0198 ± 0.0001
Linear SVM		85.49 ± 0.0713	85.97 ± 0.0869	0.8340 ± 0.0001	5.2759 ± 0.0984	3.3522 ± 0.1002
RBF SVM		90.81 ± 0.1204	88.60 ± 0.1993	0.8973 ± 0.0001	3.3988 ± 0.0752	4.4562 ± 0.0856

TABLE IV
CLASSIFICATION WITH SPECTRAL FEATURE ON DBN-LR AND SVM ON PAVIA, ITALY DATA SET THE MEAN VALUES ± STANDARD DEVIATION

Classifier	Depth	OA (%)	AA (%)	Kappa statistic	Training time (s)	Test time (s)
DBN-LR	1	95.71 ± 0.1045	93.98 ± 0.2217	0.9437 ± 0.0013	475.1780 ± 4.7888	0.0781 ± 0.0001
	2	96.12 ± 0.1138	94.60 ± 0.1966	0.9491 ± 0.0015	592.9500 ± 7.1190	0.0622 ± 0.0001
	3	96.21 ± 0.1337	94.86 ± 0.1982	0.9502 ± 0.0018	668.0000 ± 16.0468	0.0634 ± 0.0001
	4	**96.42 ± 0.1461**	**95.09 ± 0.4901**	**0.9530 ± 0.0019**	761.2000 ± 8.1670	0.0788 ± 0.0002
	5	96.36 ± 0.1527	95.00 ± 0.4851	0.9522 ± 0.0021	933.5260 ± 10.2401	0.1213 ± 0.0013
Linear SVM		91.49 ± 0.0985	88.01 ± 0.5523	0.8876 ± 0.0015	57.2033 ± 0.1850	20.9867 ± 0.2063
RBF SVM		95.84 ± 0.1852	94.11 ± 0.8892	0.9453 ± 0.0068	15.3584 ± 0.4130	15.4892 ± 0.3660

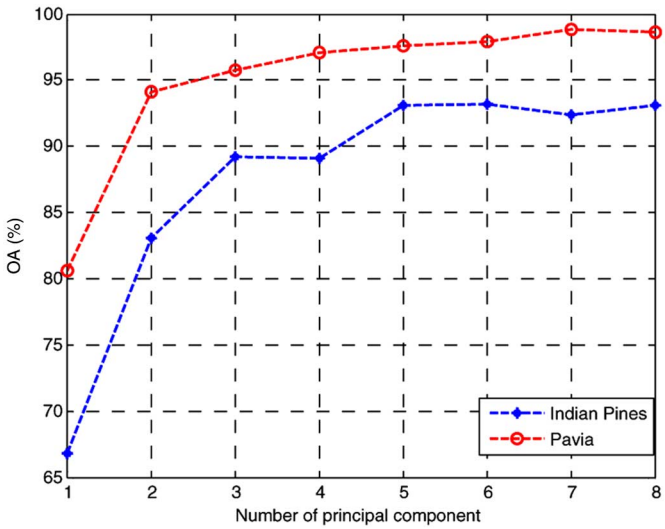


Fig. 10. Influence of the number of PCs.

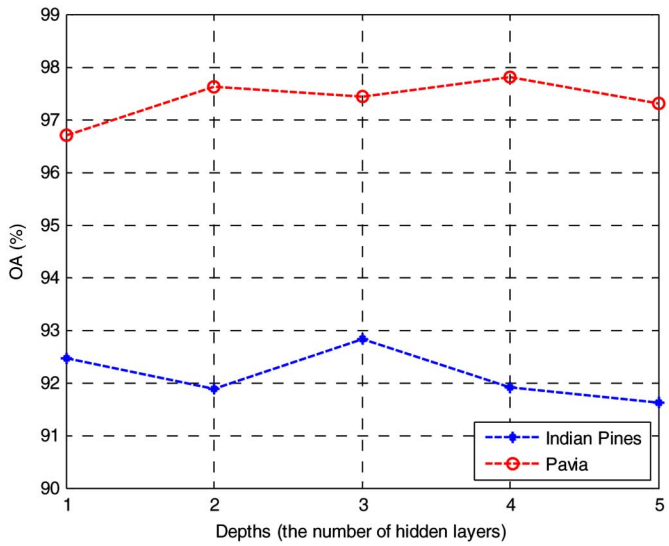


Fig. 11. Influence of depths (the spatial framework).

the architecture is combined by one RBM (the depth is 1), DBN-LR gives lower accuracies than RBF SVM. However, as the architecture gets deeper, DBN-LR increases the mean OAs by 0.28%–0.58%. Between the two SVM methods, SVM with linear kernel does not perform as well as SVM with RBF kernel.

The paired t-test between RBF SVM and DBN-LR with the best depth were made and the results show that improvements on OA are statistically significant (at the level of 95%) for the two data sets.

D. Spatial Classification

In this section, spatial information was incorporated. Furthermore, the spatial FE method results were examined by varying the number of retained PCs, and the depth of neural network.

1) Influence of the Number of PCs: Although the proposed spatial method mainly focuses on extracting spatial information of hyperspectral data, the number of PCs affects the classification accuracy. The amount of spectral information can be measured by the number of PCs to keep. Here, we varied the number of retained PCs from 1 to 8, and check how the final classification accuracy was affected. In Fig. 10, the results of the DBN-LR models with two hidden layers are presented. It shows that as the number of PCs increases, the classification accuracies of both data sets become higher. As a tradeoff between accuracy and computational complexity, a reasonable number of PCs is 5.

2) Influence of Depths: A series of DBNs with different depths but with fixed principal component numbers (5 PCs) and hidden unit numbers (50 hidden units) were trained, aimed to observe how the depth of the features affects overall classification accuracy. The results are show in Fig. 11. The best depths are 3 and 4 for two data sets, respectively. Compared with spectral information, deeper features are required for spatial information to get the best OA. The drops of OAs with 5 or

TABLE V
SPATIAL AND SPECTRAL–SPATIAL CLASSIFICATION OF DBN-LR AND SVM ON TWO DATA SETS: THE MEAN VALUES ± STANDARD DEVIATION

Datasets	Measurements	DBN-LR		RBF-SVM		EMP RBF-SVM
		Spatial	Spectral-Spatial	Spatial	Spectral-Spatial	
Indian Pines	Overall Accuracy (%)	93.20 ± 0.2594	**95.95 ± 0.1872**	92.42 ± 0.1341	95.53 ± 0.1325	95.10 ± 0.1893
	Average Accuracy (%)	92.12 ± 0.1980	**95.45 ± 0.1745**	91.26 ± 0.1822	95.12 ± 0.1248	94.71 ± 0.2465
	Kappa Coefficient	0.9226 ± 0.0022	**0.9539 ± 0.0014**	0.9227 ± 0.0024	0.9511 ± 0.0069	0.9497 ± 0.0021
Pavia	Overall Accuracy (%)	98.62 ± 0.1288	**99.05 ± 0.0711**	98.17 ± 0.1422	98.38 ± 0.2250	97.84 ± 0.1121
	Average Accuracy (%)	97.95 ± 0.1517	**98.48 ± 0.1005**	97.04 ± 0.1756	98.16 ± 0.1897	97.75 ± 0.1822
	Kappa Coefficient	0.9819 ± 0.0012	**0.9875 ± 0.0009**	0.9733 ± 0.0035	0.9836 ± 0.0022	0.9780 ± 0.0011

more hidden layers indicate that too deep architectures bring opposite effect.

E. Spectral–Spatial Classification

Spectral–spatial classification framework combines the spectral and spatial information together to form a hybrid input, and uses the DLA to classify hyperspectral data, as detailed in Section III-C. Aimed at investigating the classification performance of proposed method, several experiments were conducted.

1) Comparisons With Spatial Classification and SVMs: Here, we compared spectral–spatial classification with the aforementioned spatial method. We also performed RBF-SVM using both kinds of information to form a control group.

In Table V, we can see that for both the DBN-LR and RBF-SVM methods, spectral–spatial features perform better than spectral features and spatial features in terms of classification performance. While comparing the two methods within each feature set, DBN-LR is better. As illustrated in Section III-C, the dimensionality of spectral–spatial feature is higher than the other two, and improved classification accuracy with spectral–spatial features indicates the potential of DBN-LR in dealing with hyper-dimensional feature space.

2) Comparing With Other Spatial Method: Extended morphological profile (EMP) has been developed in recent years, which integrates spatial information into spectral-based classifiers [53]. EMP followed by SVM is an advanced spatial–spectral classification method for hyperspectral data. This method was tested in this study. We used opening and closing operations on the first three PCs of hyperspectral data to extract structure information. In the experiments, the structure shape used was disk and the structure sizes were from 1 to 4. Therefore, 24 spatial features were generated. A range of c, g values for the SVM was searched in the EMP with RBF-SVM method, and for the Indian Pines data they were configured as $c = 0.0313$ and $g = 1$, while those in the Pavia data were $c = 128$, $g = 32$. Table V shows the results obtained by EMP.

Compared with the EMP followed by RBF-SVM, DBN-LR performs better in terms of OA, kappa statistic, and AA. Paired t-tests of OA also show that the spectral–spatial framework with DBN-LR is consistently better than the EMP with RBF-SVM.

3) Influence of Depths: In accordance with the spectral and spatial frameworks, the depth of network also has an important influence on the classification performance of spectral–spatial classification framework.

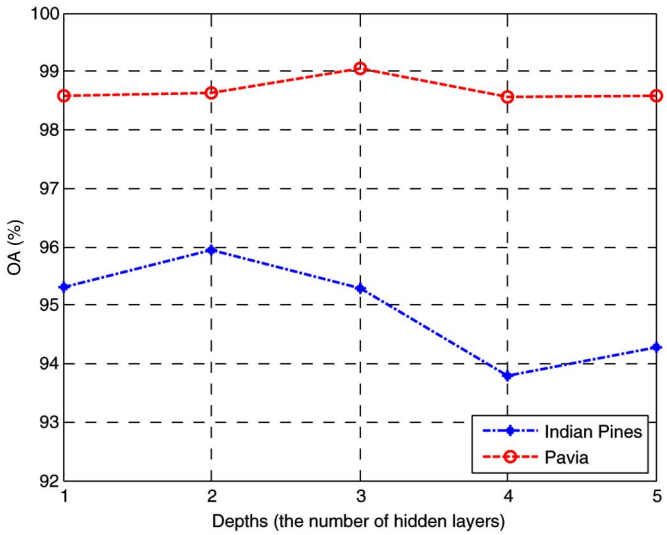


Fig. 12. Influence of depths (the spectral–spatial framework).

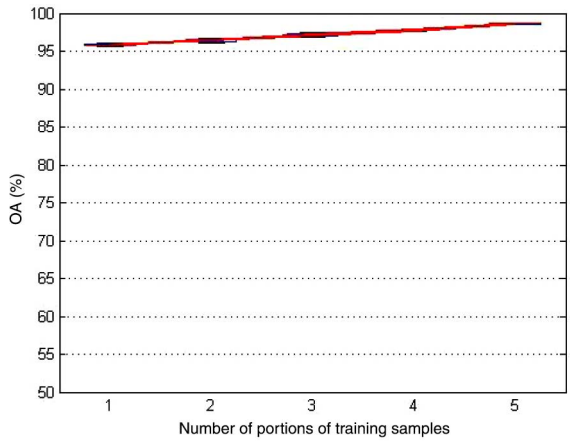


Fig. 13. Influence of the training sample size (Pavia). The ratio between training and test samples varies from 1:5 to 5:5.

In this section, we implement several DBNs with different depths but with fixed principal component numbers (5 PCs). The numbers of hidden unit are 60 and 50 for the two data sets, respectively. The influence of depth on the OA is shown in Fig. 12. For the Indian Pines data, the OAs float up and down, which are sensitive to the depths. In contrast, the OAs of Pavia data range in a smaller degree. The best depths are 2 and 3 for

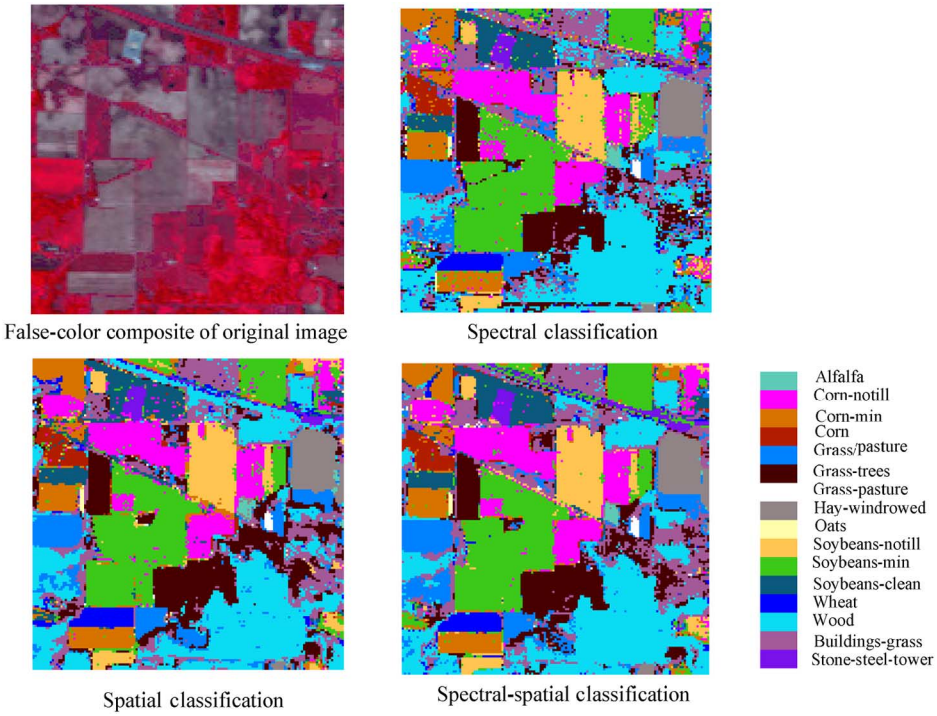


Fig. 14. Spectral, spatial, and spectral–spatial classification using DBN-LR on the whole image of Indian Pines data set.

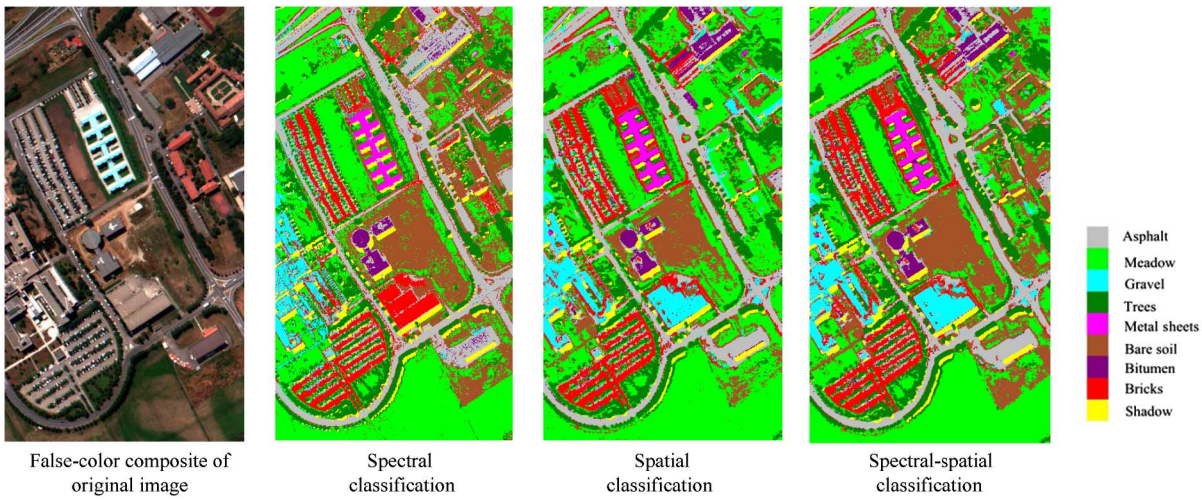


Fig. 15. Spectral, spatial, and spectral–spatial classification using DBN-LR on the whole image of Pavia data set.

two data sets, respectively. Aimed to get the best OA, the selection of depths are different for the three frameworks. The drops of OAs with 4 or more hidden layers indicate that too deep architectures bring opposite effect, which has been discussed in the spectral and spatial frameworks.

4) Influence of the Training Sample Size: In this section, experiments are conducted to explore the performance of DBN-LR with limited training samples.

Here, we varied the ratio between training and test samples from 1:5 to 5:5, and checked how the final classification OAs were affected. Fig. 13 shows the results of DBN-LR models. The confidence intervals obtained by the mean and standard deviation are shown as box plots. It can be seen that the OA decreases only slightly with the reduction of training samples. The performance is similar when the portion of training samples varies from 1 to 5, which is promising.

F. Visual Inspection on the Whole Image

In this section, we examine the classification accuracy from a visual perspective. We used the best DBN-LR models for the spectral, spatial, and spatial–spectral sets of features to classify the whole images of Indian Pines and Pavia. All the parameters in these models were optimized.

From the resulting images, we can figure out how the proposed spatial information extraction method affects the classification results. In Figs. 14 and 15, we can see that spectral classification always results in noisy scatter points, and that

spatial features correct this shortcoming. However, spatial features have their own flaws. They misclassify certain small regions, like the mixture of meadow and bare soil in the Pavia data. This can be found in the area of bare soil on the top left side of the Pavia data. Finally, for the spectral–spatial classification, it gives a satisfying tradeoff. It retains the shape and detail of some objects, while simultaneously eliminating noisy scattered points of misclassification.

V. Discussion and Conclusion

In this study, a spectral–spatial classification strategy based on DBN was proposed, tested in experiments, and discussed in the context of hyperspectral data. Four research questions are stated regarding DBN's FE, classification using DBN with spectral features, spatial features, and spectral–spectral features, respectively.

Based on our experiments, it can be assessed that DBN is an effective FE method, which reduces the dimension of feature and presents a good reconstruction computed with extracted activations. For hyperspectral data classification with three kinds of features, our proposed DBN-LR methods provide better classification performance than SVM in most cases.

Parameters selections, such as the depth of features and the number of hidden units, have a large influence on the classification accuracy and computational complexity. If the depths are set improperly, the performance of DBN-LR may not be even better than SVM. To find the best number of layers and the number of hidden units to use requires extensive exploration of all combinations of values (i.e., grid search). However, the grid search actually costs excessive computational complexity. The optimization of the hyper-parameter is a new research topic in deep learning field. We will try to explore on more efficient methods for the parameter selection as a part of our future work. However, our experimental results provide some guidelines with regard to reliable ranges for the two parameters, i.e., about 2–4 hidden layers of RBMs with 30–60 hidden units per layer appear sufficient.

It is conceded that the training complexity of DBN is a disadvantage, but they are super-fast on the testing time (classification of unknown data). The super-fast classification stage is a great advantage when large hyperspectral images are processed.

For our proposed spatial and spectral–spatial feature-based classification, both DBN-LR and SVM have shown the effectiveness of the PCA-window spatial information extraction method. The combination of spectral–spatial feature and the DBN-LR classifier yields the highest classification accuracy. It also reveals the potential processing power of DBN-LR in hyper-dimensionality feature space. In addition, our proposed spectral–spatial framework shows sound performances than EMP with RBF-SVM.

In our future work, we explore on other deep architectures and examine their use for hyperspectral image classification. In addition, various spatial features will be tested for effective spatial information extraction.

References

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002.

[2] J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction*. New York, NY, USA: Springer, 2013.

[3] F. M. Lacar, M. M. Lewis, and I. T. Grierson, "Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley, South Australia," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, vol. 6, Sydney, Australia, 2001, pp. 2875–2877.

[4] F. V. D. Meer, "Analysis of spectral absorption features in hyperspectral imagery," *Int. J. Appl. Earth Observ. Geo. Inform.*, vol. 5, no. 1, pp. 55–68, Jan. 2004.

[5] P. W. Yuen and M. Richardson, "An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition," *Imag. Sci. J.*, vol. 58, no. 5, pp. 241–253, 2010.

[6] H. Zhang, J. Li, Y. Huang, and L. Zhang, "A nonlocal weighted joint sparse representation classification method for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2056–2065, Jun. 2014.

[7] E. K. Hege *et al.* "Hyperspectral imaging for astronomy and space surveillance," in *Proc. SPIE's 48th Annu. Meet. Opt. Sci. Technol.*, 2004, pp. 380–39.

[8] H. Yuan, Y. Yan Tang, Y. Lu, L. Yang, and H. Luo, "Hyperspectral image classification based on regularized sparse representation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2174–2182, Jun. 2014.

[9] T. J. Malthus and P. J. Mumby, "Remote sensing of the coastal zone: An overview and priorities for future research," *Int. J. Remote Sens.*, vol. 24, no. 13, pp. 2805–2815, Nov. 2003.

[10] J. B. Dias *et al.*, "Hyperspectral remote sensing data analysis and future challenges," *Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, 2013.

[11] S. Rajan, J. Ghosh, and M. M. Crawford, "An active learning approach to hyperspectral data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 4, pp. 1231–1242, Apr. 2008.

[12] G. M. Foody and M. Ajay, "A relative evaluation of multiclass image classification by support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 6, pp. 1335–1343, Jun. 2004.

[13] L. O. Jimenez and D. A. Landgrebe, "Hyperspectral data analysis and supervised feature reduction via projection pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2653–2667, Nov. 1999.

[14] X. Jia, B. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," in *Proc. IEEE*, vol. 101, no. 3, pp. 676–679, Mar. 2013.

[15] C. I. Chang, Q. Du, T. Sun, and M. L. G. Althouse, "A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2631–2641, Nov. 1999.

[16] S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1360–1367, Jul. 2001.

[17] F. Samadzadegan, H. Hasani, and T. Schenk, "Simultaneous feature selection and SVM parameter determination in classification of hyperspectral imagery using Ant Colony Optimization," *Can. J. Remote Sens.*, vol. 38, pp. 139–156, 2012.

[18] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2331–2338, Oct. 2002.

[19] J. C. Harsanyi and C. I. Chang, "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.

[20] F. Melgani and B. Lorenzo, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[21] A. Ambikapathi, T.-H. Chan, C.-H. Lin, and C.-Y. Chi, "Convex geometry based outlier-insensitive estimation of number of endmembers in hyperspectral images," *Signal*, vol. 1, p. 1–20, 2012.

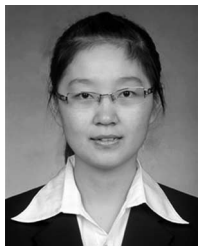
[22] A. B. Santos, A. de Albuquerque Araujo, and D. Menotti, "Combining multiple classification methods for hyperspectral data interpretation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1450–1459, Jun. 2013.

[23] A. Plaza, J. Plaza, and G. Martin, "Incorporation of spatial constraints into spectral mixture analysis of remotely sensed hyperspectral data," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2009, pp. 1–6.

[24] Y. Qain and M. Ye, "Hyperspectral imagery restoration using nonlocal spectral-spatial structured sparse representation with noise estimation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 499–515, Apr. 2013.
[25] M. Fauvel *et al.*, "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.
[26] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.
[27] J. Liu *et al.*, "Spatial-spectral kernel sparse representation for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2462–2471, Dec. 2013.
[28] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.
[29] Y. Bengio and Y. LeCun, *Scaling Learning Algorithms Towards AI*. Cambridge, MA, USA: MIT Press, 2007.
[30] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[31] N. Kruger *et al.*, "Deep hierarchies in primate visual cortex what can we learn for computer vision?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1847–1871, Aug. 2013.
[32] G. Wang, D. Hoiem, and D. Forsyth, "Learning image similarity from flickr groups using fast kernel machines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2177–2188, Nov. 2012.
[33] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[34] D. Yu, L. Deng, and S. Wang, "Learning in the deep structured conditional random fields," in *Proc. Neural Inf. Process. Syst. Workshop*, Dec. 2009, pp. 1–8.
[35] A. D. Mohamed and G. Hinton, "Deep belief networks for phone recognition," in *Proc. Neural Inf. Process. Syst. Workshop*, Dec. 2009, pp. 1–9.
[36] Z. Zuo and G. Wang, "Learning discriminative hierarchical features for object recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1159–1163, Sep. 2014.
[37] H. Larochelle *et al.*, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 473–480.
[38] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area v2," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2008, pp. 873–880.
[39] D. Yu, G. Hinton, N. Morgan, and J. Chien, "Introduction to the special section on deep learning for speech and language processing," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 1, pp. 4–6, Jan. 2012.
[40] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, pp. 1527–1554, 2006.
[41] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, pp. 193–202, 1980.
[42] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Neural Inf. Process. Syst.*, vol. 19, 2007, pp. 153–160.
[43] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1711–1800, 2002.
[44] H. Chen and A. F. Murray, "Continuous restricted Boltzmann machine with an implementable training algorithm," *IEE Proc. Vis. Image Signal Process.*, vol. 150, no. 3, pp. 153–158, Jun. 2003.
[45] N. LeRoux and Y. Bengio, "Deep belief networks are compact universal approximators," *Neural Comput.*, vol. 22, no. 8, pp. 2192–2207, 2010.
[46] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 448–455.
[47] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proc. Int. Joint Conf. IEEE Neural Netw.*, 1989, pp. 593–605.
[48] I. Sutskever and G. E. Hinton, "Deep, narrow sigmoid belief networks are universal approximators," *Neural Comput.*, vol. 20, no. 11, pp. 2629–2636, 2008.
[49] W. Huang, G. Song, H. Hong, and K. Xie "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1–11, Oct. 2014.
[50] R. Salakhutdinov and G. Hinton, "Using deep belief nets to learn covariance kernels for Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2008, pp. 1249–1256.
[51] Z. Zhu, C. E. Woodcock, J. Rogan, and J. Kellndorfer, "Assessment of spectral, polarimetric, temporal, and spatial dimensions for urban and peri-urban land cover classification using Landsat and SAR data," *Remote Sens. Environ.*, vol. 117, pp. 72–82, 2012.
[52] Y. Chen, X. Zhao, and Z. Lin, "Optimizing subspace SVM ensemble for hyperspectral imagery classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1295–1305, Apr. 2014.
[53] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles" *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.
[54] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

Yushi Chen (M'11) received the Ph.D. in information and communication engineering degree from Harbin Institute of Technology, Harbin, China, in 2008.

Currently, he is an Associate Professor with the School of Electrical and Information Engineering, Harbin Institute of Technology, Harbin, China. He has authored more than 20 peer-reviewed papers, and he is the inventor or co-inventor of three patents. His research interests include hyperspectral data analysis, ensemble learning, deep learning, and remote sensing applications.



Xing Zhao (S'14) received the Bachelor's degree in information antagonizing technology from the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China, in 2013.

Currently, she is a Graduate Student with the Institute of Image and Information Technology, Harbin Institute of Technology. Her research interests include hyperspectral image processing, machine learning, and deep learning.

Xiuping Jia (M'93–SM'03) received the B.Eng. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1982, and the Ph.D. degree in electrical engineering from the University of New South Wales, Sydney, Australia, in 1996.

Since 1988, she has been with the School of Information Technology and Electrical Engineering, University of New South Wales, Sydney, Australia, where she is currently a Senior Lecturer. She is also a Guest Professor with Harbin Engineering University, Harbin, China, and an Adjunct Researcher with China National Engineering Research Center for Information Technology in Agriculture, Beijing, China. She is the co-author of the remote sensing textbook titled *Remote Sensing Digital Image Analysis* (Springer-Verlag, 3rd ed. (1999) and 4th ed. (2006)]. Her research interests include remote sensing and image data analysis.

Dr. Jia is an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. She has served as the Inaugural Chair of IEEE ACT&NSW Section GRSS Chapter from 2010 to 2013.