A methodology for unsupervised feature learning in hyperspectral imagery using deep belief network

C. Sherin Shibi* and R. Gayathri

Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Sriperumbudur 602 117, India

Deep learning approaches have received major interest in the field of remote sensing. Hyperspectral imaging has rich data that are distributed in multi-dimensions. It is challenging to apply deep learning algorithms due to the limited amount of labelled data. So, unsupervised feature extraction approaches are used to overcome this limitation. In this study, we propose an unsupervised feature learning approach using deep belief network (DBN). In the proposed framework, the input hyperspectral image is segmented using entropy rate superpixel segmentation and filtered by domain transform recursive filter which extracts spatial and spectral information effectively. Then the features are learned by improved DBN. In the traditional methods, DBN is stacked with restricted Boltzmann machine (RBM) which is suitable for only binary value data. In the proposed framework, we used Gaussian–Bernoulli RBM which was constructed for real value data such as images. The experiments were carried out using Pavia University dataset. The results show that the proposed network has good performance in terms of classification accuracy and computation time.

Keywords: Deep belief network, hyperspectral image, remote sensing, spatial–spectral classification, superpixel segmentation.

In recent years, hyperspectral imaging has received much attention in the field of computer vision. In hyperspectral imagery, the images are collected at visible or near-infrared wavelength, which identifies objects that are not visible to the human eye. Hyperspectral images are collected at hundreds of narrow and continuous spectral bands. Hyperspectral imaging shows better performance in target separation and identification compared to multi-spectral imaging. Thus hyperspectral images are widely used in both military and civilian applications such as environment monitoring, fire tracking, vegetation, etc.[1].

In traditional classification and detection algorithms, pixel-level analysis is used. It has been noted that when pixels are taken individually for analysis, they are sensitive to intra-class variability and noise. To avoid this problem, the entire image is characterized as multiple local regions defined by a window or grid. These approaches can also affect the performance of classification algorithms when there is a complex background. Thus, segmentation is preferred over fixed window representation, as it has well-defined boundaries. Superpixel segmentation divides the image into spectrally similar regions consisting of several contiguous pixels[2]. Superpixels are widely used in image-processing techniques to reduce the complexity of images from thousands of pixels to few hundreds of superpixels. The superpixel algorithm segments an image into small, non-overlapping, homogeneous regions with adaptive shape and size.

In the past decade, many algorithms have been proposed for superpixel segmentation in the hyperspectral domain. It is necessary to perform segmentation prior to endmember extraction while calculating region-specific endmembers. This technique provides true boundaries with the optimal feature vectors and is more compact than the original image. According to the techniques used to develop a superpixel, the superpixel segmentation algorithms can be classified into graph-based methods[3], gradient-ascent-based methods[4] and cluster-based methods[5]. Entropy rate superpixel segmentation (ERS)[6] and simple linear iterative clustering (SLIC)[7] are the most widely used superpixel segmentation algorithms in the hyperspectral image processing. In this study, we have adopted ERS for superpixel segmentation.

Due to the high dimensionality of data, hyperspectral images lead to Hughes phenomenon[8], which degrades the accuracy of classification algorithms. Feature extraction is an effective approach which reduces the dimension of the data. In particular, the hyperspectral image is projected onto another feature space and only important features are maintained for classification. The most widely used feature extraction techniques are principal component analysis (PCA)[9], independent component analysis (ICA)[10], linear discriminant analysis (LDA)[11], and so on. PCA shows superior performance compared to the other methods as it holds the most significant spectral features of hyperspectral image (HSI) in a few principal components (PCs). In ICA, the components are independent and

*For correspondence. (e-mail: shibicharles@gmail.com) computation complexity is high when compared to PCA. LDA fails when the features between classes are nonlinearly separated in the input space. Moreover, these techniques consider only the spectral features, whereas the spatial information is not included. The filtering techniques are used to extract the features in HSI which consider both spatial and spectral information. Recently, many researchers have shown interest on the edge-preserving filtering algorithms (EPF) in hyperspectral imaging[12–15]. In this study, we have adopted domain transform recursive filter (DTRF) to extract the spatial and spectral features.

Recently, many deep learning algorithms have been introduced for hyperspectral classification. Support vector machine (SVM) is a widely used classification algorithm which has good performance in terms of classification accuracy with few training samples, and it does not depend on spectral dimension[16,17]. The traditional classification methods have only one layer which will not extract the features potentially. The neural network models have achieved tremendous growth in the field of computer vision. Convolutional neural networks (CNN) is an effective model to extract the deep features and shows remarkable performance in classification[18]. These deep learning models need huge training data to get good results, but hyperspectral imaging has only a limited number of labelled training data. Deep belief network (DBN) is a deep learning model which shows good performance with less number of training data[19]. Here, we adopt DBN with three hidden layers to extract the deep features. Softmax classifier is used for classification.

In this study, the superpixel segmentation algorithm has been adopted to reduce the complexity of processing. Also, it helps retrieve the optimal features. DTRF is carried out to combine the spatial and spectral features. The deep spatial and spectral feature extraction architecture based on DBN stacked with Gaussian–Bernoulli Restricted Boltzmann Machine (GBRBM) is proposed for efficient classification of hyperspectral imaging. The learned features are classified by the softmax classifier.

Related work

Entropy rate superpixel segmentation

As the hyperspectral images have high dimensional data, it is necessary to adopt dimensionality reduction algorithm prior to superpixel segmentation. PCA is performed to reduce the spectral dimension, which gives an optimal representation of the hyperspectral image. The first PC is used to build the superpixel map as it contains most of the hyperspectral image information[20].

In ERS, the hyperspectral image is segmented using the graph optimization problem, so that it can preserve the boundaries. The image is mapped to the weighted graph $G = (V, E)$, where V is the vector set that represents the first image components and E is the edge set that represents the pairwise similarities between adjacent pixels. The similarity between adjacent pixels is measured with the weight function which is defined using Gaussian kernel as

$$w_{i,j} = \exp\left(-\frac{d(v_i, v_j)^2}{2\sigma^2}\right), \quad (1)$$

where $d(v_i, v_j)$ is the intensity difference multiplied by the spatial distance, and σ is the kernel parameter. The main aim of this algorithm is to select the subset of edges $A \subseteq E$, and the resulting graph $G_A = (V, A)$ has K connected subgraphs. Let us assume that every vertex in the graph has a self-loop. If the edge is not included in A, the edge weight of the self-loop of the associated vertices will increase. These subgraphs are considered as superpixels $S_{N_A} = \{S_1, S_2, \ldots, S_K\}$. The objective function is a combination of the entropy rate and balancing function which results in compact, homogeneous and balanced clusters. These clusters are obtained by optimizing the objective function corresponding to the edge set.

$$\max_A \mathcal{H}(A) + \lambda B(A), \quad (2)$$

subject to $A \subseteq E$ and $N_A = K$, where $\mathcal{H}(A)$ is the entropy rate, $B(A)$ the cluster distribution balancing term, K the number of superpixels and $\lambda \geq 0$ is the weight of the balancing term. The objective function is submodular and increases monotonically as it is a linear combination with non-negative coefficients. The objective function can be optimized using iterative greedy algorithm.

Edge preserving filter

In recent years, edge preserving filters (EPFs) are widely used in feature extraction which incorporates the spatial information of hyperspectral images. Here, we have used DTRF, which is a real-time filter[21]. It has low computational complexity compared to bilateral filtering[17], and shows significant improvement in classification accuracy[14]. In this filter, the input signal I is transformed to the domain transform Ω_ω. Then, the transformed coordinate $ct(x_\mathrm{m})$ is computed for each pixel to calculate the distance between two pixels. The function $ct(u)$ that defines the domain transform of the signal $I(x)$ is given as

$$ct(u) = \int_0^u 1 + \frac{\delta_s}{\delta_r} \,|\, I'(x) \,|\, \mathrm{d}x, \quad u \in \Omega_\omega, \quad (3)$$

where $I'(x)$ is the derivative of the input signal $I(x)$, δ_s and δ_r are the spatial and range parameters, and Ω_ω is the domain transform signal. The transformed signal is computed using recursive filter as follows

$$J[m] = (1 - c^d) I[m] + c^d J[m-1], \tag{4}$$

where $J[m]$ is the filter output result, c the feedback coefficient which is expressed as

$$c = \exp\left(\frac{-\sqrt{2}}{\delta_{\text{s}}}\right) \in [0, 1]$$

and d is the distance between the neighbouring samples x_m and x_{m-1} which is expressed as $d = ct(x_m) - ct(x_{m-1})$. The propagation chain will be stopped when d increases, which leads c^d to zero. In this way, the edges are preserved and pixels on the same side of the edge will have the almost same output. In 2D images, 1D filtering operations are computed for each dimension. In this study, we have adopted three iterations of 1D filtering, which give satisfactory filtering results.

Restricted Boltzmann machine

Restricted Boltzmann machine (RBM) is a layerwise training network which has only two layers, viz. visible and hidden. In this network, the binary units are defined with an energy function. There is no connection between visible and visible layers as well as between hidden and hidden layer. The visible layer is connected only to the hidden layer. Figure 1 shows the structure of RBM.

RBM is an energy-based probabilistic model in which the global energy of the configuration of visible and hidden layers is given by

$$E(v, h|\theta) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i W_{ij} h_j, \tag{5}$$

where v_i are the units of visible layer i, h_j the units of the hidden layer j, W_{ij} the weight connecting the visible and hidden layers, and a_i, b_j are the biases of the visible and hidden layer respectively. The term $\theta = \{W_{ij}, a_i, b_j\}$ defines the model parameters of RBM, and the total number of visible and hidden units is given as m and n respectively. The probability distribution of the layers is represented with an energy function as follows.

$$P(v, h|\theta) = \frac{1}{Z(\theta)} \exp(-E(v, h|\theta)), \tag{6}$$

$$Z(\theta) = \sum_{v}\sum_{h} E(v, h|\theta), \tag{7}$$

where Z is the normalizing constant which is the partition factor in correspondence with physical systems. In spite of the distinct structure of RBM, the visible and hidden layers are conditionally independent, and the distributions are given by logistic functions

$$P(h_j = 1|v, \theta) = \sigma\left(b_j + \sum_i v_i W_{ij}\right), \tag{8}$$

$$P(v_i = 1|h, \theta) = \sigma\left(a_i + \sum_j W_{ij} h_j\right), \tag{9}$$

where $\sigma(x)$ is the sigmoid function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}.$$

The weights can be trained by the method called contrastive divergence (CD)[22]. The updated weight is given as

$$w_{ij} = w_{ij} + \varepsilon(E_{\text{data}}[v_i h_j] - E_{\text{model}}[v_i h_j]), \tag{10}$$

where ε is a learning rate, $E_{\text{data}}[v_i h_j]$ represents the data distribution and $E_{\text{model}}[v_i h_j]$ represents the model distribution. The hidden layer holds information of the training data which represent the original data. This information is learnt as features from the input.

Gaussian–Bernoulli RBM

The classical RBM has been designed only for binary valued data. To solve this problem, the Gaussian–Bernoulli restricted Boltzmann machine (GBRBM) was introduced to process the real valued images[23]. The binary visible layer in classical RBM is replaced by the Gaussian

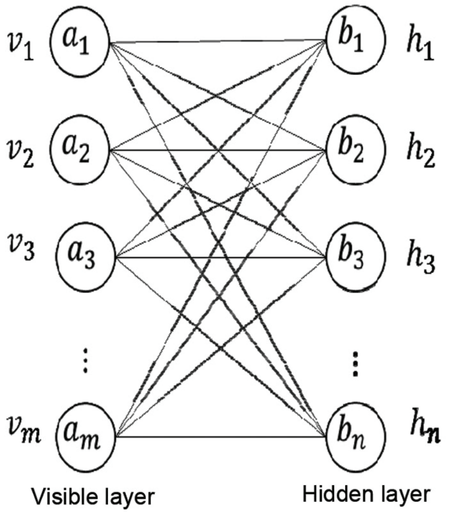

Figure 1. Structure of Restricted Boltzmann Machine.

visible layer, whereas the hidden layer can hold the binary-valued data. The energy function of GBRBM is given as

$$E_{\mathrm{GBRBM}}(v,h) = -\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{v_i}{\sigma_i}h_jW_{ij} - \sum_{i=1}^{m}\frac{v_i - a_i^2}{2\sigma_i^2} - \sum_{j=1}^{n}h_jb_j, \tag{11}$$

where σ_i^2 represents the variance of the visible layer. The probability distribution of the visible and hidden layers is computed as

$$p(h_j = 1|v) = \sigma\left(\sum_{i=1}^{m}\frac{v_i}{\sigma_i^2}W_{ij} + b_j\right), \tag{12}$$

$$p(v_i = v|h) = \mathcal{N}\left(\sum_{j=1}^{n}h_jW_{ij} + v_ia_i, \sigma_i^2\right), \tag{13}$$

where $\mathcal{N}(.\,|\,\mu,\sigma^2)$. presents the Gaussian probability density function with mean μ and variance σ^2. The learning process is carried out using the CD method which reduces the complexity of computing model distribution and also improves the learning performance. The updated weight of GBRBM after the training process is

$$w_{ij} = w_{ij} + \varepsilon\left(E_{\mathrm{data}}\left[\frac{v_ih_j}{\sigma^2}\right] - E_{\mathrm{model}}\left[\frac{v_ih_j}{\sigma^2}\right]\right). \tag{14}$$

To reduce the complexity of the model, σ is considered as a constant (=1).

Deep belief network

DBN is the layer-by-layer learning system which is formed by stacking multiple RBMs. In DBN, the learning of deep features can be done by pre-training. The input is mapped into the first RBM and the output of first RBM is given as the input of the second RBM. The process is continued till the last RBM, and the features of the whole network are obtained from the last RBM. Figure 2 shows the architecture of DBN.

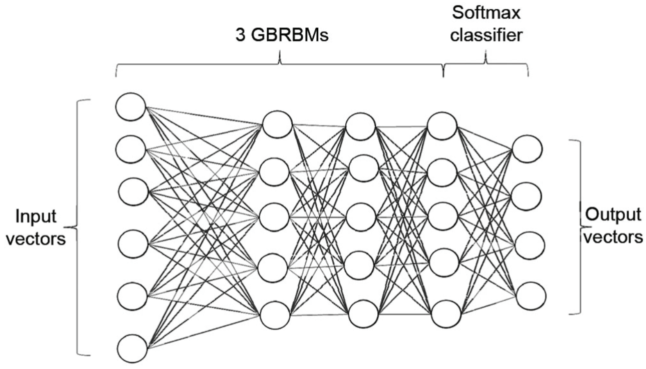

Figure 2. Structure of deep belief network.

DBN is a probabilistic graphical model that learns to extract a deep feature of the training data[24]. DBN is formed by stacking RBMs that train in a greedy manner. The training process involves two stages. The first is the unsupervised feature learning in which the RBM learns by the greedy learning algorithm. The second stage is supervised fine tuning, which is done by back propagation algorithm. Here, we have used GBRBM instead of the classical RBM.

Proposed approach

The proposed approach consists of three steps: (i) Developing superpixels using ERS. (ii) Filtering using DTRF. (iii) Classification based on DBN.

Step (i): Initially, PCA algorithm is applied to the original image I to generate the PCs which reduce the spectral bands of HSI, and is represented as

$$I_{\mathrm{p}} = \mathrm{PCA}(I), \tag{15}$$

where I_{p} is the dimension-reduced image and p is the reduced number of bands. Then applying ERS, 2D a superpixel map is generated with S_n superpixels. The pixels in each superpixel are linked to the same label.

$$I_{\mathrm{s}} = ERS(I_{\mathrm{p}}), \tag{16}$$

where I_{s} is the superpixel map of the band reduced image.

Step (ii): DTRF is applied to the superpixels to extract the features and is represented as

$$Q = \mathrm{DTRF}_{\delta_{\mathrm{s}},\delta_{\mathrm{r}}}(I_{\mathrm{s}}), \tag{17}$$

where δ_{s} and δ_{r} are the spatial and range standard deviations respectively, of the recursive filter and Q is the filtered image.

Step (iii): In the proposed framework, DBN stacked with GBRBM is used as a deep learning architecture. As there is limited number of training samples, the network should not be too deep because of the overfitting problem. So, in the proposed framework we have used three GBRBMs in our network. The spatial and spectral features from DTRF are given as 1D vectors to the DBN network as training samples. The training samples are given to the first layer and the learned features are retained in first hidden layer. Then the visible layer of the second GBRBM uses these features as input and the learned features are maintained in second hidden layer.

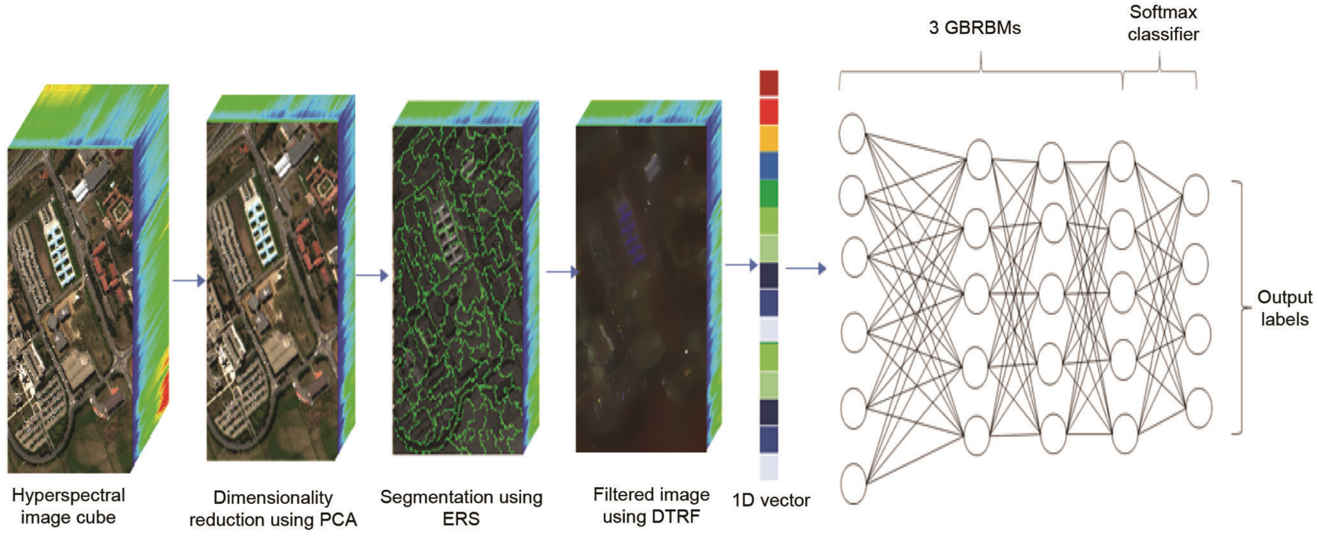

Figure 3. Proposed framework for spatial–spectral hyperspectral classification.

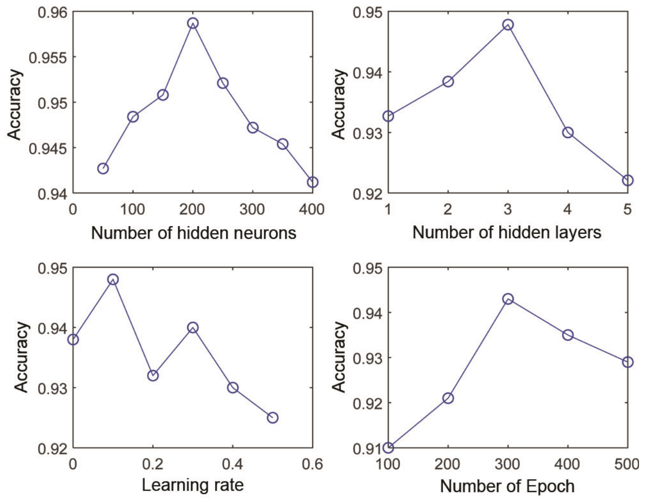

Figure 4. Relationship between the number of hidden neurons, number of hidden layers, learning rate, number of epochs and accuracy.

Table 1. Classes and samples in the Pavia University dataset

Class	Samples	Training samples	Testing samples
Asphalt	6,631	663	5,968
Meadows	18,649	1,865	16,784
Gravel	2,099	210	1,889
Trees	3,064	306	2,758
Painted metal sheets	1,345	135	1,211
Bare soil	5,029	503	4,526
Bitumen	1,330	133	1,197
Self-blocking bricks	3,682	368	3,314
Shadows	947	95	852
Total	42,776	4,278	38,498

This will be followed by the third GBRBM. This learning process is considered as the pre-training stage which utilizes the greedy learning procedure. In the fine-tuning stage, the classification is carried out along with the softmax layer. Figure 3 shows the proposed framework for hyperspectral image classification.

Results and discussion

In this study, we have used Pavia University dataset for validating the performance of the proposed framework. The experiments have been implemented using Matlab R2018a and the operating system is Win10 with 64-bit CPU Intel Core i5 processor.

Dataset description

The Pavia University image was collected using the reflective optics system imaging spectrometer (ROSIS) sensor. It has 103 spectral bands after removal of noise bands and consists of 610 × 340 pixels. The spatial resolution is 1.3 m with wavelength ranging from 0.43 to 0.86 μm. The nine classes of this image scene are asphalt, meadows, gravel, trees, painted metal sheets, bare soil, bitumen, self-blocking bricks and shadows (Table 1)[25].

Evaluation criteria

To evaluate the performance of the proposed framework, the quantitative metrics used are overall accuracy (OA), average accuracy (AA) and kappa coefficient. OA is defined as the ratio of the number of correctly classified pixels to the total number of samples. AA is defined as the ratio of the sum of accuracies per class to the total number of classes. Kappa coefficient compares the difference between the classification map and ground truth.

In the experimental analysis, the parameters that are tuned for better classification results are the number of hidden layers, number of hidden neurons, epoch and the

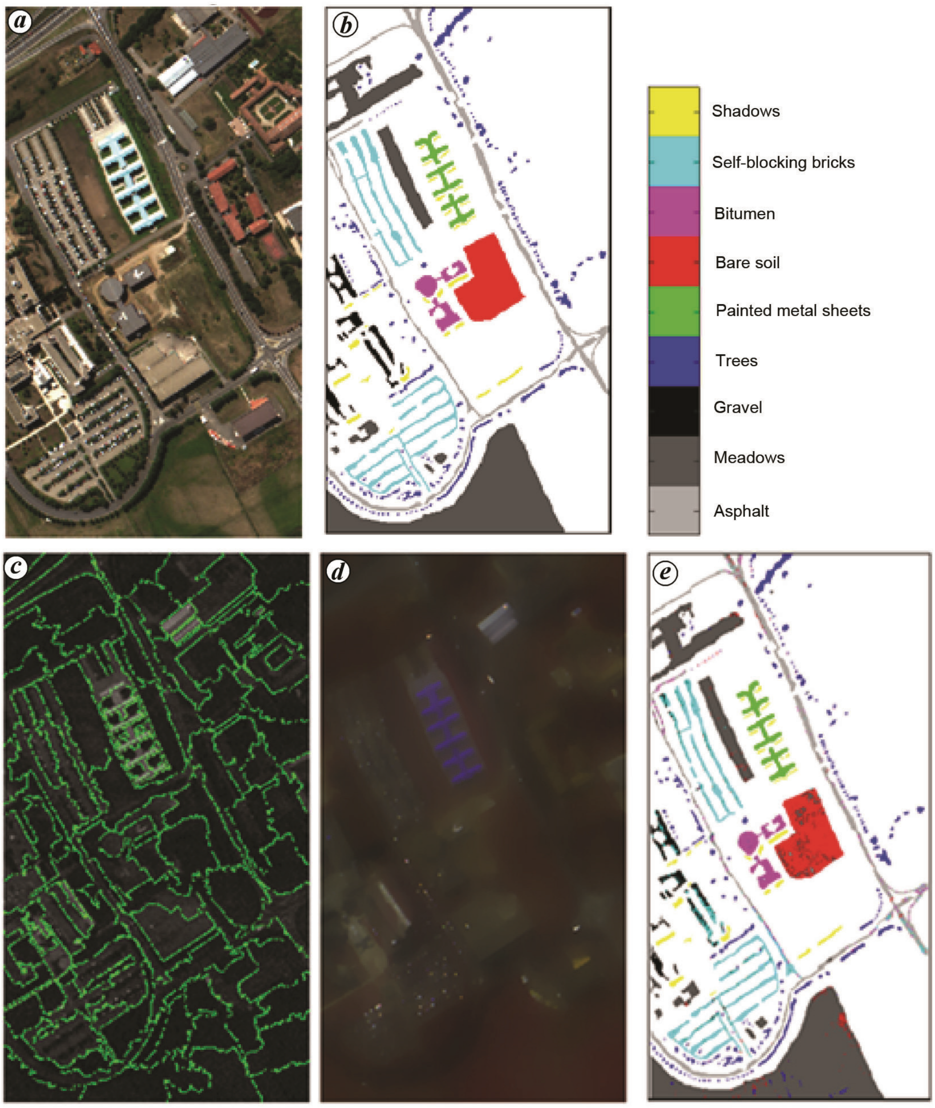

Figure 5. Results of the proposed framework: ***a***, RGB image of Pavia University. ***b***, Ground truth. ***c***, Result of entropy rate superpixel segmentation. ***d***, Results of domain transform recursive filter filtering. ***e***, Classification map of the proposed framework.

Table 2. Classification results using the Pavia University dataset

Model	Linear SVM	Super BF SVM	DBN	CNN	DBN network
Overall accuracy (%)	89.78	93.30	89.82	94.14	95.65
Average accuracy (%)	85.33	92.78	91.65	93.47	94.93
Kappa statistics	0.80	0.91	0.90	0.91	0.94
Time (sec)	2.43	12.17	0.75	54.72	0.17

learning rate. The number of hidden neurons and the number of layers are the key parameters to improve classification performance and are selected in such a way that the features are not over-trained. The parameter learning rate governs the learning process. If the learning rate is too small, the training process will take a long time and if it is too large, the training process will produce an unstable output. So, it is important to select a proper learning rate for the training process to improve the performance.

Figure 4 shows the relationship between the various parameters that are tuned in experimental analysis and accuracy. The number of hidden neurons chosen for the DBN network is 200, which gives a high accuracy value 0.958. The number of hidden layers in the proposed framework is three, which ensures that the features are not over-trained. The learning rate used in the DBN model is 0.1 and the number of epochs is 300.

Figure 5 *a* shows the RGB image of Pavia University. Figure 5 *b* shows the ground truth of Pavia University with nine classes. Figure 5 *c* shows the superpixels using ERS segmentation algorithm. Figure 5 *d* shows the filtered image using DTRF algorithm and Figure 5 *e* shows the classification map of the proposed framework.

Comparative analysis

To verify the performance of the proposed algorithm, linear SVM, Superpixel-based Bilateral Filtering-SVM, DBN and CNN were used for analysis. Table 2 shows the classification results of these algorithms. In the experimental analysis, we used Compute Unified Device Architecture to reduce the training time. From Table 2, it can be seen that the prediction time is less in the proposed framework when compared with other algorithms. The overall accuracy of the proposed framework is 95.65%, which is better than other algorithms.

Conclusion

Here, we propose a novel framework for extraction of spatial and spectral features based on DBN. The framework includes superpixel segmentation, which is a powerful tool to retrieve spatial and spectral information of hyperspectral images. The superpixels are filtered using EPF, which improves the classification accuracy. The proposed DBN model stacked with GBRBM learns the deep features by unsupervised pre-training and supervised fine-tuning. The softmax classifier is added to the framework for the classification of learned features. The proposed model is compared and analysed with the other popular classifiers. The proposed model performs better than these classification techniques in terms of accuracy. Moreover, the prediction time is less compared to the other algorithms. In future work, we will concentrate on the optimization of parameters used in deep learning framework.

1. Khan, M. J., Khan, H. S., Yousaf, A., Khurshid, K. and Abbas, A., Modern trends in hyperspectral image analysis: a review. *IEEE Access*, 2018, **6**, 14118–14129.
2. Massoudifar, P., Rangarajan, A. and Gader, P., Superpixel estimation for hyperspectral imagery. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 287–292; doi:10.1109/CVPRW.2014.51.
3. Felzenszwalb, P. F. and Huttenlocher, D. P., Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, 2004, **59**, 167–181.
4. Levinshtein, A. *et al.*, TurboPixels: fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, 2290–2297.
5. Achanta, R., Shaji, A., Smith, K. and Lucchi, A., SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, 2274–2282.
6. Liu, M. Y., Tuzel, O., Ramalingam, S. and Chellappa, R., Entropy rate superpixel segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2011, pp. 2097–2104; doi:10.1109/CVPR.2011.5995323.
7. Xie, F., Lei, C., Yang, J. and Jin, C., An effective classification scheme for hyperspectral image based on superpixel and discontinuity preserving relaxation. *Remote Sensing*, 2019, **11**, 1149.
8. Hughes, G. F., On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory*, 1968, **14**, 55–63.
9. Prasad, S. and Bruce, L. M., Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geosci. Remote Sensing Lett.*, 2008, **5**, 625–629.
10. Villa, A., Benediktsson, J. A., Chanussot, J. and Jutten, C., Hyperspectral image classification with independent component discriminant analysis. *IEEE Trans. Geosci. Remote Sensing*, 2011, **49**, 4865–4876.
11. Liao, W., Pižurica, A., Scheunders, P., Philips, W. and Pi, Y., Semisupervised local discriminant analysis for feature extraction in hyperspectral images. *IEEE Trans. Geosci. Remote Sensing*, 2013, **51**, 184–198.
12. Chen, Z., Jiang, J., Jiang, X., Fang, X. and Cai, Z., Spectral–spatial feature extraction of hyperspectral images based on propagation filter. *Sensors* (*Switzerland*), 2018, **18**, 1–15.
13. Tu, B., Yang, X., Li, N., Ou, X. and He, W., Hyperspectral image classification via superpixel correlation coefficient representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sensing*, 2018, **11**, 4113–4127.
14. Kang, X., Li, S. and Benediktsson, J. A., Feature extraction of hyperspectral images with image fusion and recursive filtering. *IEEE Trans. Geosci. Remote Sensing*, 2014, **52**, 3742–3752.
15. Chen, Z., Jiang, J., Zhou, C., Fu, S. and Cai, Z., SuperBF: superpixel-based bilateral filtering algorithm and its application in feature extraction of hyperspectral images. *IEEE Access*, 2019, **7**, 147796–147807.
16. Melgani, F. and Bruzzone, L., Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sensing*, 2004, **42**, 1778–1790.
17. Kang, X., Member, S., Li, S. and Benediktsson, J. A., Spectral – spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. Geosci. Remote Sensing*, 2014, **52**, 2666–2677.
18. Chen, Y., Jiang, H., Li, C., Jia, X. and Ghamisi, P., Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sensing*, 2016, **54**, 6232–6251.
19. Chen, Y., Zhao, X. and Jia, X., Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sensing*, 2015, **8**, 2381–2392.
20. Duan, W., Li, S. and Fang, L., Spectral-spatial hyperspectral image classification using superpixel and extreme learning machines. *Commun. Comput. Inf. Sci.*, 2014, **483**, 159–167.
21. Gastal, E. S. L. and Oliveira, M. M., Domain transform for edge-aware image and video processing. *ACM Trans. Graph.*, 2011, **30**, 1–12.
22. Hinton, G. E., Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 2002, **14**, 1771–1800.
23. Cho, K. H., Raiko, T. and Ilin, A., Gaussian–Bernoulli deep Boltzmann machine. In Proceedings of the International Joint Conference on Neural Networks, 2013; doi:10.1109/IJCNN.2013.6706831.
24. Liu, P., Zhang, H. and Eom, K. B., Active deep learning for classification of hyperspectral images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sensing*, 2017, **10**, 712–724.
25. Hyperspectral remote sensing scenes; http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 6 June 2018).

Received 26 May 2020; accepted 17 February 2021

doi: 10.18520/cs/v120/i11/1705-1711