Deep Learning Based Feature Selection for Remote Sensing Scene Classification

Qin Zou (邹勤), *Member, IEEE*, Lihao Ni (倪立昊), Tong Zhang (张彤), and Qian Wang (王骞)

***Abstract*—With the popular use of high-resolution satellite images, more and more research efforts have been placed on remote sensing scene classification/recognition. In scene classification, effective feature selection can significantly boost the final performance. In this letter, a novel deep-learning-based feature-selection method is proposed, which formulates the feature-selection problem as a feature reconstruction problem. Note that the popular deep-learning technique, i.e., the deep belief network (DBN), achieves feature abstraction by minimizing the reconstruction error over the whole feature set, and features with smaller reconstruction errors would hold more feature intrinsics for image representation. Therefore, the proposed method selects features that are more reconstructible as the discriminative features. Specifically, an iterative algorithm is developed to adapt the DBN to produce the inquired reconstruction weights. In the experiments, 2800 remote sensing scene images of seven categories are collected for performance evaluation. Experimental results demonstrate the effectiveness of the proposed method.**

***Index Terms*—Deep belief network (DBN), feature learning, iterative deep learning, scene recognition, scene understanding.**

I. Introduction

IN THE past decade, with the development of high-resolution and high-speed imaging sensors, high-resolution satellite images have become easier and cheaper to obtain. The high-resolution images, on the one hand, provide accurate information for people to understand the world, while, on the other hand, bring challenges to the remote-sensing community in smart image interpretation.

In fact, people already have made a lot of efforts to interpret remote sensing images, in which image classification has been a hot research topic for decades [1]. One decade ago, pixel-level image classification, which may be referred to as image segmentation in computer vision, domains the research efforts and makes numerous successes. This is because satellite images were mostly in low resolution at that period, e.g., Thematic Mapper (TM) series images, Systeme Probatoire d'Observation dela Terre (SPOT) series images, etc. In a low-resolution satellite image, pixel-level classification is useful and adequate for some geographical applications, e.g., the evaluation of the forest coverage rate, the rice planting area, the urban expanding speed, etc.

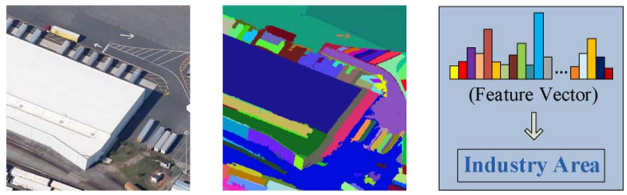

Fig. 1. Difference between pixel-level image classification and scene classification of remote sensing images. (a) Remote sensing scene image. (b) Pixel-level image classification result. (c) Feature vector and scene-classification result.

However, when high-resolution satellite images are in popular use, e.g., the IKONOS and QuickBird images, pixel-level classification methods gradually show their limitations. This is because one single pixel does not make as much sense in a high-resolution image as in a low-resolution image, particularly in representing an object. An example is shown in Fig. 1(b), where the pixel-level classification result is not informative enough to interpret the high-resolution scene shown in Fig. 1(a). To better interpret high-resolution satellite images, features of image patches or subimages of even larger size are inquired, and the context information should be taken into account. From this perspective, it is necessary to study feature-based remote sensing scene classification/recognition.

Generally, a remote sensing scene can be classified into a specific scene theme, e.g., a part of a forest, a parking lot, a farmland, etc. In such a classification, supervised learning techniques are usually employed [2], where a scene image is first represented as a feature vector and then fed into a learning machine for training and testing, as illustrated in Fig. 1(c). In the feature-based image representation, in addition to feature extraction and feature coding, feature selection is another important issue. An effective feature-selection method can significantly improve the final performance [3]–[6]. Therefore, the research and development of effective feature-selection methods hold great importance.

In this letter, we develop a novel feature-selection method based on the deep-learning technique. Note that the deep learning implemented by the deep belief network (DBN) [7] has attracted wide attention owing to its effectiveness in feature abstraction. In [8], a feature-selection method working on DBN features is proposed and successfully used for gene feature selection. In this method, supervised active learning is utilized

Manuscript received March 7, 2015; revised June 30, 2015 and August 7, 2015; accepted August 28, 2015. Date of publication September 18, 2015; date of current version October 27, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61301277 and 41371431, by the National Basic Research Program of China under Grants 2012CB719906 and 2012CB725303, and by the 3551 Optics Valley Talents Scheme of Wuhan East Lake High-Tech Zone. *(Corresponding author: Tong Zhang.)*

Q. Zou, L. Ni, and Q. Wang are with the School of Computer Science, Wuhan University, Hubei 430072, China (e-mail: qzou@whu.edu.cn; lhni@whu.edu.cn; qianwang@whu.edu.cn).

T. Zhang is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Hubei 430079, China (e-mail: zhangt@whu.edu.cn).

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/LGRS.2015.2475299

to remove the less informative DBN features at first, and then, t-test is applied on the remaining features for discriminative feature selection. While in our work, feature selection is performed in an unsupervised way. In fact, the DBN achieves feature abstraction by minimizing the feature reconstruction error over all the input features. With the fine-tuned reconstruction weights from the DBN, features with small reconstruction errors generally show a good intrinsic, which can be transformed into the feature discriminative power for classification. To this end, we formulate the feature-selection problem as a feature reconstruction problem. Specifically, we select the discriminative features by filtering off the features with higher reconstruction errors. In addition, we develop an iterative algorithm to adapt the DBN to produce the inquired reconstruction weights. In the experiments, we show the effectiveness of the proposed feature-selection method on a benchmark data set containing 2800 remote sensing scene images.

The contributions of this work are threefold. First, to the best of our knowledge, it is the first to use deep learning for feature selection in the field of remote sensing. Second, it formulates the problem of feature selection into a problem of feature reconstruction in the computation of the DBN. Third, it contributes a new benchmark data set to the remote sensing community to promote the research on remote sensing scene classification.

The remainder of this letter is organized as follows. Section II reviews the DBN, a popularly used deep-learning technique. Section III describes the proposed feature-selection method. Section IV reports the experiments and results, and Section V concludes our work.

II. DBN

As a subfield of machine learning, deep learning [7], [9], [10] has attracted wide attention since the beginning of this decade. In visual cortex study, the brain is found to have a deep architecture consisting of several layers. In visual activities, signals flow from one brain layer to the next to obtain different levels of abstraction. By simulating the function of the deep architecture of the brain, deep learning uses a set of algorithms in machine learning to attempt to model high-level abstractions in data by using model architectures composed of multiple nonlinear transformation learning [11]. In this letter, we use the popular DBN [7] to investigate feature abstraction and reconstruction, based on which we develop a feature-selection method.

The deep-learning procedure of the DBN consists of two stages: layer-wise feature abstracting and reconstruction weight fine-tuning [7]. At the first stage, the DBN uses a family of restricted Boltzmann machines (RBMs) [12] to calculate the reconstruction weights layer-wisely. At the second stage, the DBN performs a backpropagation to fine-tune the weights obtained from the first stage [7], [13].

Let v be a visible layer and h be a hidden layer, and in both layers, each node represents a feature dimension. In the DBN, all nodes are binary random variables, i.e., 0 or 1, which satisfy the Boltzmann distribution, and then, the bipartite graph formed by fully connecting the nodes across the two layers is an RBM.

In an RBM, a joint configuration of the visible and hidden units has an energy

$$E(v,h;\theta) = -\sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j \qquad (1)$$

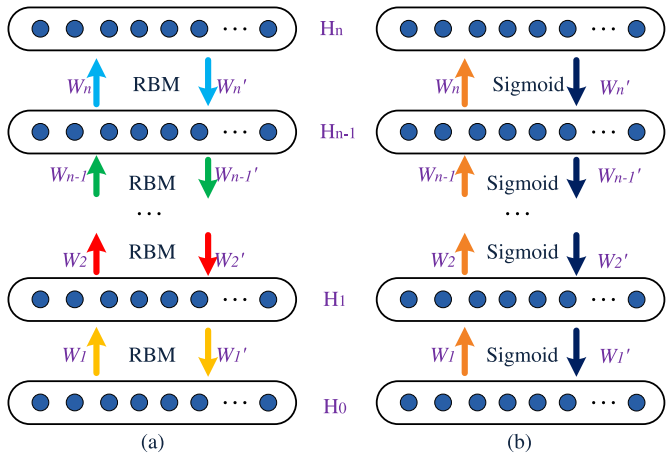

Fig. 2. Feature learning procedure of the DBN. (a) n-layer feature learning with RBMs. (b) Backpropagation with the sigmoid kernel to fine-tune all the weights. W and W' represent weights between the neighboring layers, down–up and up–down, respectively. H_0 is the input visible layer, and H_1 to H_n are the hidden layers.

where θ denotes the parameters (i.e., W, a, b); W denotes the weights between visible and hidden nodes (units); and a and b denote the bias of the hidden and visible layers, respectively. Then, the joint probability of the joint configuration can be determined by the Boltzmann distribution, i.e.,

$$P_\theta(v,h) = \frac{1}{Z(\theta)} \exp\left(-E(v,h;\theta)\right) \qquad (2)$$

where $Z(\theta) = \sum_{v,h}(\exp(-E(v,h;\theta))$ is the normalization factor. Combining (1) and (2), we have

$$P_\theta(v,h) = \frac{1}{Z(\theta)} \exp\left(\sum_{i,j} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\right). \qquad (3)$$

Because of the bipartite structure of RBMs, the visible and hidden units are conditionally independent to each other. Thus, the marginal distribution of v with respect to h can be written as

$$P_\theta(v) = \frac{1}{Z(\theta)} \exp[v^T W h + a^T h + b^T v]. \qquad (4)$$

As θ can be obtained by maximizing the likelihood of $\log(P_\theta(v))$, the hidden layer h becomes a visible layer. We can repeat the same algorithm to learn a multiple-layer deep Boltzmann machine, as shown in Fig. 2(a).

At the second stage of feature learning, a backpropagation is applied as a whole over all the layers, as illustrated in Fig. 2(b), where the weights obtained from the first stage are fine-tuned.

In this letter, we use the DBN implementation [14][1] for feature abstraction and reconstruction. Specifically, we try up to four hidden layers in the DBN with the decreasing number of nodes. In the next section, we will investigate how to use the DBN for feature selection.

III. Proposed Method

As discussed earlier, feature reconstruction in the DBN can be achieved by applying the layer-wise weights W and

[1] http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html

W' on the input features. Thus, the layer-wise weights can be treated as a kind of "reconstruction weights." Given all layer-wise reconstruction weights, a reconstruction error can be calculated for each input feature. The variation of features would commonly make the difference of reconstruction errors. Intuitively, a feature with a lower reconstruction error is more reconstructible. In the feature learning procedure of the DBN, we suppose that more reconstructible features are more prone to hold the feature intrinsic. As feature intrinsic is vital in image representation, we propose to select the features that are more reconstructible as the discriminative features, which leads to a new feature-selection method.

A. Iterative Feature Learning

To obtain reliable reconstruction weights, we iteratively apply the feature learning procedure of the DBN to remove the feature outliers. The feature outliers are those with large reconstruction errors, which can be identified by analyzing the distribution of the reconstruction errors.

Let $V = \{v_i | i = 1, 2, \ldots, n\}$ be the input features, v_i' be the reconstructed feature corresponding to v_i, then the reconstruction error of feature v_i can be calculated as $e_i = \|v_i' - v_i\|$, and an average error can be calculated by

$$\hat{e} = \frac{1}{n} \sum_{i}^{n} e_i. \tag{5}$$

Algorithm 1 Iterative Feature Reconstruction

procedure IFR
  input:
    $V_1 = \{v_i | i = 1, 2, \ldots, n\}$: the original input features,
    η: the ratio of feature outliers to the whole inputs,
    ε: the stop criteria,
    nIt: the max number of iterations.
  output:
    M: the final reconstruction weight matrix.
  for ($j = 1$ to nIt) **do**
    % Get the weight matrix and the average error:
    $\{M_j, \hat{e}_j\} \leftarrow \mathbf{DBN}(V_j)$
    $M = M_j$;
    % Check the stop option:
    if $\hat{e}_{j-1} - \hat{e}_j < \varepsilon$ **then** break;
    end if
    % Calculate the threshold according to η:
    $\tau \leftarrow \mathbf{Bipart}(V_j, \eta)$;
    % Filter the features with τ:
    $V_{j+1} \leftarrow \mathbf{Filter}(V_j, \tau)$;
  end for
end procedure

Details of the proposed iterative algorithm for feature reconstruction are given in Algorithm 1. In each iteration of feature learning, a threshold τ is calculated according to η. The parameter η defines the ratio of outliers to the whole input features, which is empirically to $\eta = 5 \times 10^{-2}$. The features after filtering can be described as

$$V' = \{v_i | e_i < \tau, v_i \in V\}. \tag{6}$$

It should be noted that, apart from a stop criteria set by the maximum iteration times, there is another parameter ε in Algorithm 1 to terminate the feature learning procedure. A low difference between the average reconstruction error of the current iteration and that of the last will stop the iteration.

B. Feature Selection

As the feature outliers have been eliminated during the iterative feature learning, the final reconstruction weight matrix is more reliable for feature reconstruction. Here, we introduce how to use the weight matrix obtained from training images to achieve feature selection for the testing images.

Suppose M is the reconstruction weight matrix obtained by Algorithm 1, I is an image in the testing data set, $V_N^I = \{v_i^I | i = 1, 2, \ldots, N\}$ is a number of N features extracted from I, then the reconstructed features $V_N'^I$ can be written as

$$V_N'^I = \mathbf{Reconstruct}\left(V_N^I, M\right) \tag{7}$$

and the reconstruction error E_N^I with respect to to V_N^I can be calculated by

$$E_N^I = \left\| V_N'^I - V_N^I \right\|. \tag{8}$$

As discussed in the beginning of Section III, we propose to select the features with smaller reconstruction errors as the discriminative features. Thus, we proportionally filter features with higher reconstruction errors. To achieve this, a parameter δ is defined by

$$\delta = (1 - \eta)^t \tag{9}$$

where η is the parameter that defines the ratio of outliers in Algorithm 1, and t is the real number of iterations of feature learning. As η and t can be obtained from Section III-A, δ is then a known variable.

After sorting the error values in E_N^I in ascending order, we take the error value at order $\delta \cdot N$ as an error threshold, which is denoted by τ^I for convenience. Then, the feature selection on image I can be performed by

$$V^I = \left\{ v_i^I | e_i^I < \tau^I, e_i^I \in E_N^I \right\}. \tag{10}$$

IV. Experiments and Results

Here, we validate the proposed feature-selection method. First, a remote sensing scene data set is introduced. Then, the parameter settings for feature selection are described. Finally, experimental results are given for the proposed method, as well as the comparison methods.

A. Dataset

To evaluate the proposed feature-selection method, a remote sensing scene data set is collected, namely, the RSSCN7.[2] The data set RSSCN7 contains 2800 remote sensing scene images, which are from seven typical scene categories, namely, the grassland, forest, farmland, parking lot, residential region, industrial region, and river and lake. For each category, there

[2]RSSCN7 is available at https://sites.google.com/site/qinzoucn/documents.

Fig. 3. Sample images from the RSSCN7 data set used in this letter. From left to right (in columns): grassland, farmland, industrial and commercial regions, river and lake, forest field, residential region, and parking lot. There are four scales, from top to bottom (in rows): 1:700, 1:1300, 1:2600, and 1:5200.

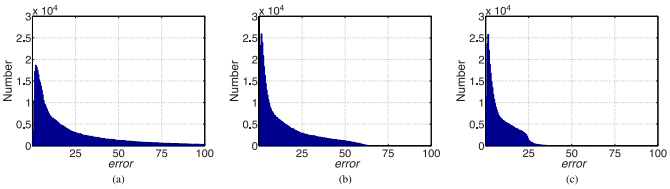

Fig. 4. Error distributions at different iterations. (a) First iteration. (b) Third iteration, (c) Ninth iteration. Note that the x-axis denotes the value of the reconstruction error, and the y-axis denotes the number of features.

are 400 images collected from the Google Earth, which are sampled on four different scales with 100 images per scale. Each image has a size of 400 $\times$ 400 pixels. This data set is rather challenging due to the wide diversity of the scene images that are captured under changing seasons and varying weathers and sampled on different scales. Some sample images from RSSCN7 are shown in Fig. 3.

B. Feature Selection

Iterative Feature Learning: Half of the RSSCN7 data set is used to compute the reconstruction weight matrix. We randomly select 1400 images from the total to run the iterative feature reconstruction algorithm, as described in Section III-A.

Considering that deep learning can well handle the primitive visual features, we simply sample a number of image patches from each image to construct the input features for the DBN. The patch size is empirically set to 28 $\times$ 28. For each image, we randomly sample 400 patches, since 400 patches are adequate to cover the whole image.

For algorithm 1, the stop criterion ε is defined as $\varepsilon = \mathbf{min}\{(1/50)\hat{e}_1, (1/2)\}$, where $\hat{e}_1$ is the average reconstruction error at the first iteration of feature learning. The number of iterations is set to $n\mathrm{It} = 15$. In practice, it finishes in nine iterations.

Fig. 4 shows the error distribution at different iterations of feature learning. From Fig. 4, we can see that the reconstruction errors centralize to the axis of $x = 0$ under the ongoing iteration. Table I lists the average reconstruction errors of nine iterations, as well as $\Delta\hat{e}$, i.e., the difference between the current $\hat{e}$ and the last $\hat{e}$. It can be seen that the average reconstruction error becomes smaller during the iterations.

TABLE I
AVERAGE RECONSTRUCTION ERRORS AT DIFFERENT ITERATIONS, AS WELL AS THEIR DIFFERENCE ($\Delta\hat{e}_i = \hat{e}_{i-1} - \hat{e}_i$)

iteration	1	2	3	4	5
$\hat{e}$	24.1855	18.4612	14.6720	13.4748	11.4922
$\Delta\hat{e}$	-	5.7243	3.7892	1.1972	1.9826
iteration	6	7	8	9	
$\hat{e}$	10.8186	9.1555	8.6058	8.3430	
$\Delta\hat{e}$	0.6736	1.6631	0.5497	0.2628	

Feature Selection: A number of 2000 patches are randomly sampled on each image for feature selection. Features after feature selection will then be fed into a bag-of-words framework for classification.

C. Scene Classification

A standard bag-of-words framework is employed, and a nonlinear support vector machine using the χ^2 kernel is applied. A vocabulary containing 1024 words is produced by the k-means technique. In feature quantization, hard assignment is used.

Comparative Study: To validate the effectiveness of the proposed feature-selection method, we compare it with four other methods. To make a fair comparison, we conduct experiments by varying the number of training data from 10% to 90% of all data, at an interval of 10%. In each experiment, we randomly select the according percentage of images for each category for training and the rest for testing. The procedure is repeated 30 times, and the results are averaged.

First, we make comparisons by using and not using feature selection in classification, under the same experimental settings. The "Subimage" denotes the method without feature selection, which uses raw patches $(28 \times 28 \times 3)$ as the input. Second, we compare the proposed method with three other methods that also use deep-learning features for classification. Among them, one is the "DBN features," using DBN features in a bag-of-words framework, which has a similar feature-coding strategy with that in [13]. The DBN is configured with four hidden layers, with 1024, 512, 256, and 128 nodes for the first to the fourth layer, respectively. Another one is the "dynamic convolutional neural network (DCNN) features," which conducts deep learning with an eight-layer convolutional neural network [15]. The DCNN is equipped with a fixed learning-rate policy, with 1×10^{-4} as the base learning rate, 9×10^{-1} as the momentum, and 4×10^{-3} as the weight decay. The third one is "DBN feature selection [8]," which iteratively executes active learning, with 200 iterations, to collect an informative feature set from the DBN features, and then performs t-test for feature selection.

The results are plotted in Fig. 5. It can be seen from Fig. 5 that the proposed method obtains much higher performance than the baseline method—"Subimage." It simply indicates that the proposed feature-selection method is effective to select the discriminative features for this scene-classification task. In Fig. 5, "DBN features" shows a slightly lower performance than "Subimage." It indicates that the feature outliers would undermine the performance, which, in contrary, demonstrates the necessity of feature selection. With a feature-selection step, "DBN feature selection [8]" also earns a significant improvement over "DBN features." In Fig. 5, "DCNN features" is observed to outperform "DBN features" and even outperform "DBN feature

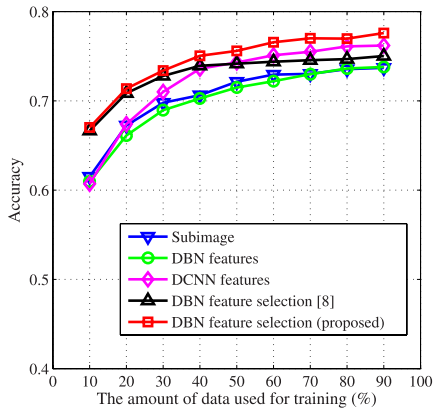

Fig. 5. Comparison of the classification results. In "DBN features," features are encoded by the DBN. In "DCNN features," features are encoded by the DCNN.

TABLE II
TIME COST IN THE TRAINING PROCESS (UNIT: HOUR)

method	DCNN	DBN	DBN feat. sel. [8]	proposed
running time	33	16	25	26

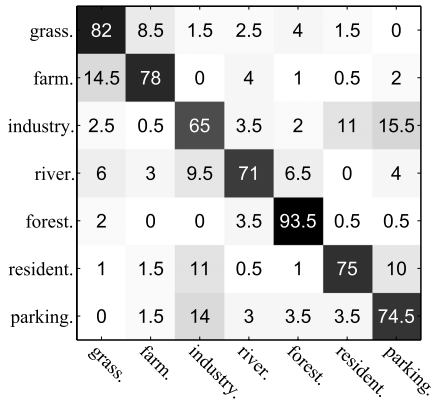

Fig. 6. Confusion matrix of the classification results achieved by the proposed feature-selection method using the fixed 50% data samples for training.

selection [8]" when the training data is over 50%. However, the proposed method still holds a clearly higher performance than "DCNN features." Moreover, the training time of DCNN is the longest, as shown in Table II.

Performance Analysis: To further evaluate the performance of the proposed feature-selection method, we fix half of the images in each category for training and the rest for testing. The information of file division is also included in RSSCN7. Average accuracy of 77.0% is achieved. Fig. 6 shows the confusion matrix of the classification results. As can be seen from the diagonal of the matrix, accuracy of 93.5% is gained on "forest"—the highest score among all scene themes. This is because the appearance of a forest scene is relatively more distinctive than that of other scene themes, which makes "forest" the least confusable. We can also see that the lowest accuracy is observed on "industry," who encounters heavy confusion with the "parking" and the "resident." This is because scenes from the industry area are easily to be tangled with scenes from the parking lots and the residential area.

V. CONCLUSION

In this letter, a novel deep-learning-based feature-selection method has been proposed, which formulates the feature-selection problem as a feature reconstruction problem in a DBN. An iterative feature-learning algorithm was developed to get reliable reconstruction weights, and features with relatively small reconstruction errors were taken as the discriminative features. A benchmark data set containing 2800 remote sensing images of seven categories was created for evaluation. Experimental results validated the effectiveness of the proposed method. In the future work, we will use graphics processing units [16], [17] to accelerate the feature learning process.

ACKNOWLEDGMENT

The authors would like to thank S. T. Ouyang from Wuhan University for help in collecting the RSSCN7 data set, as well as the anonymous reviewers for their constructive comments on this letter.

REFERENCES

[1] L. Gómez-Chova, G. Camps-Valls, J. Munoz-Mari, and J. Calpe, "Semisupervised image classification with Laplacian support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 336–340, Jul. 2008.
[2] M. Ferecatu and N. Boujemaa, "Interactive remote-sensing image retrieval using active relevance feedback," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 818–826, Apr. 2007.
[3] G. Camps-Valls, J. Mooij, and B. Scholkopf, "Remote sensing feature selection by kernel dependence measures," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 3, pp. 587–591, Jul. 2010.
[4] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.
[5] E. Aptoula, "Remote sensing image retrieval with global morphological texture descriptors," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 3023–3034, May 2014.
[6] J. A. Piedra-Fernández, M. Cantón-Garbín, and J. Z. Wang, "Feature selection in AVHRR ocean satellite images by means of filter methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 12, pp. 4193–4203, Dec. 2010.
[7] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[8] R. Ibrahim, N. A. Yousri, M. A. Ismail, and N. M. El-Makky, "Multi-level gene/MiRNA feature selection using deep belief nets and active learning," in *Proc. IEEE Int. Conf. Eng. Med. Biol. Soc.*, 2014, pp. 3957–3960.
[9] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Annu. Conf. NIPS*, 2006, pp. 153–160.
[10] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Proc. Annu. Conf. NIPS*, 2006, pp. 1137–1144.
[11] L. Deng and D. Yu, "Deep learning: Methods and applications," *Trends Signal Process.*, vol. 7, no. 3/4, pp. 197–387, Jun. 2013.
[12] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* vol. 1*, Foundations*. Cambridge, MA, USA: MIT Press, 1986.
[13] Q. Zou, Y. Cao, Q. Li, C. Huang, and S. Wang, "Chronological classification of ancient paintings using appearance and shape features," *Pattern Recog. Lett.*, vol. 49, pp. 146–154, Nov. 2014.
[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Annu. Conf. NIPS*, 2012, pp. 1–9.
[16] H. Yang, Q. Du, and G. Chen, "Unsupervised hyperspectral band selection using graphics processing units," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 660–668, Sep. 2011.
[17] W. Wei, Q. Du, and N. H. Younan, "Fast supervised hyperspectral band selection using graphics processing unit," *J. Appl. Remote Sens.*, vol. 6, no. 1, 2012, Art. ID. 061504.