

## Research Article

# Urban Land Use and Land Cover Classification Using Remotely Sensed SAR Data through Deep Belief Networks

Qi Lv,<sup>1,2</sup> Yong Dou,<sup>1,2</sup> Xin Niu,<sup>1,2</sup> Jiaqing Xu,<sup>2</sup> Jinbo Xu,<sup>2</sup> and Fei Xia<sup>3</sup>

<sup>1</sup>Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073, China

<sup>2</sup>School of Computer, National University of Defense Technology, Changsha 410073, China

<sup>3</sup>Electronic Engineering College, Naval University of Engineering, Wuhan 430033, China

Correspondence should be addressed to Qi Lv; lvqi@nudt.edu.cn

Received 13 November 2014; Accepted 29 January 2015

Academic Editor: Tianfu Wu

Copyright © 2015 Qi Lv et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Land use and land cover (LULC) mapping in urban areas is one of the core applications in remote sensing, and it plays an important role in modern urban planning and management. Deep learning is springing up in the field of machine learning recently. By mimicking the hierarchical structure of the human brain, deep learning can gradually extract features from lower level to higher level. The Deep Belief Networks (DBN) model is a widely investigated and deployed deep learning architecture. It combines the advantages of unsupervised and supervised learning and can archive good classification performance. This study proposes a classification approach based on the DBN model for detailed urban mapping using polarimetric synthetic aperture radar (PolSAR) data. Through the DBN model, effective contextual mapping features can be automatically extracted from the PolSAR data to improve the classification performance. Two-date high-resolution RADARSAT-2 PolSAR data over the Great Toronto Area were used for evaluation. Comparisons with the support vector machine (SVM), conventional neural networks (NN), and stochastic Expectation-Maximization (SEM) were conducted to assess the potential of the DBN-based classification approach. Experimental results show that the DBN-based method outperforms three other approaches and produces homogenous mapping results with preserved shape details.

## 1. Introduction

Urban land use and land cover (LULC) mapping is one of the core applications in remote sensing. Up-to-date LULC maps obtained by classifying remotely sensed data are essential to modern urban planning and management. In many remote sensing systems, the synthetic aperture radar (SAR) has long been recognized as an effective tool for urban analysis, as it is less influenced by solar illumination or weather conditions in contrast to optical or infrared sensors [1]. Since more scattering information can be collected in multipolarizations, polarimetric SAR (PolSAR) data have been increasingly used for urban LULC classification [2–4].

Nevertheless, most studies about urban mapping using SAR or PolSAR data are limited in identifying the urban extent or mapping very few urban classes. Few studies have focused on detailed urban mapping using SAR data. The

difficulty in detailed urban mapping using SAR data is mainly due to the complexity of the urban environment. The urban environment is comprised of various natural and man-made objects with several kinds of materials, different orientations, various shapes and sizes, and so forth, which complicates the interpretation of SAR images. Problems can also originate from the nature of polarimetric SAR imaging such as inherent SAR speckle or geometry distortions such as shadow and layover [1, 2]. As a consequence, detailed urban mapping using high resolution SAR data is still a challenging task.

Regarding the method of urban land cover mapping, approaches can be generally divided into pixel-based or object-based classification. Object-based methods, which directly explore the contextual information to improve the mapping accuracy, have been increasingly employed recently [5]. By using object-based approaches, shape characteristics and inner statistics of segmented objects can be used as

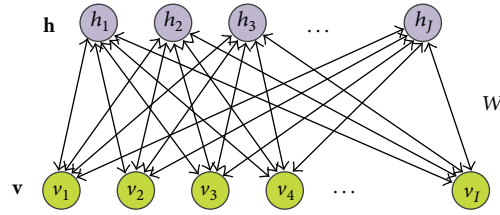


FIGURE 1: Schematic of an RBM with  $I$  visible units and  $J$  hidden units, where  $W$  is the weight matrix.

classification features [6–8]. However, the ideal segmentation on urban areas using SAR data is often difficult to achieve. Pixel-based approaches have been traditionally used for coarse-resolution SAR data with reasonable results. However, when dealing with high-resolution SAR data, the pixel-by-pixel approach is usually limited because of the speckles and increased interclass variance [9]. To cope with the problem of pixel-based approaches, some contextual analyses, such as Markov random field (MRF), have been employed [10–12]. Although contextual approaches [10–18] can learn the statistics within the local neighborhood, their capability to represent spatial patterns is limited. Moreover, although some texture indices can be used to describe certain spatial patterns, most of them are still limited in their relatively simple representation capabilities [19, 20].

From the perspective of data modeling, LULC classification methods can be grouped into parametric and nonparametric approaches. Parametric approaches, such as the minimum distance classifier, maximum likelihood classifier, and the expectation-maximization (EM) algorithm, often require proper assumptions of data distribution [21]. However, for multitemporal or multisource data, the class distributions are hard to model. On the other hand, nonparametric approaches, such as artificial neural networks, decision tree, and support vector machine (SVM), are widely used in land cover classification [22]. Nevertheless, the performance of nonparametric approaches strongly depends on the selected classification features.

As an advanced machine learning approach, deep learning has been successfully applied in the field of image recognition and classification in recent years [23–27]. By mimicking the hierarchical structure of the human brain, deep learning approaches, such as Deep Belief Networks (DBN), can exploit complex spatiotemporal statistical patterns implied in the studied data [28, 29]. For remotely sensed data, deep learning approaches can automatically extract more abstract, invariant features, thereby facilitating land cover mapping. However, to the best of our knowledge, no research has been reported using deep learning for detailed urban LULC mapping on SAR data.

The present study proposes a detailed urban LULC mapping approach based on the popular deep learning architecture DBN. This study is one of the first attempts to apply the deep learning approach to detailed urban classification. Two-date high-resolution RADARSAT-2 PolSAR data over the Great Toronto Area (GTA) have been used for evaluation.

The rest of this paper is organized as follows. Section 2 describes the proposed land cover classification approach

based on the DBN model. Section 3 introduces the data and the process of the experiment. Section 4 presents and discusses the experimental results. Finally, we conclude this paper in Section 5.

## 2. Methodology

The proposed approach is based on the DBN model. This section briefly reviews the principle of the DBN model and describes the proposed method for land cover classification.

**2.1. Deep Belief Networks.** The DBN model was introduced by Hinton et al. in 2006 [28] for learning complex data patterns. It has become one of the extensively investigated and deployed deep learning architectures [24, 25]. The DBN is a probabilistic multilayer neural network composed of several stacked Restricted Boltzmann Machines (RBMs) [28, 30]. In a DBN, every two sequential hidden neural layers form an RBM. The input of the current RBM is actually the output features of a previous one. A DBN is therefore expected to hierarchically explore the pattern features in several abstract levels, given that the features obtained by a higher-level RBM are more representative than those obtained by lower ones. The training of DBN can be divided into two steps: pretraining and fine-tuning. This training process is further discussed below.

**2.1.1. Restricted Boltzmann Machines.** As the basic component of a DBN, Restricted Boltzmann Machine (RBM) can be treated as an unsupervised energy-based generative model. An RBM consists of a layer of visible units  $\mathbf{v}$  and a layer of hidden units  $\mathbf{h}$ , connected by symmetrically weighted connections, as shown in Figure 1.

Assuming binary-valued units, the RBM defines the energy of the joint configuration of visible and hidden units  $(\mathbf{v}, \mathbf{h})$  as

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^I a_i v_i - \sum_{j=1}^J b_j h_j - \sum_{i=1}^I \sum_{j=1}^J w_{ij} v_i h_j, \quad (1)$$

where  $w_{ij}$  represents the weight associated with the connection between the visible unit  $v_i$  and the hidden unit  $h_j$ ,  $a_i$  and  $b_j$  are the bias terms, and  $I$  and  $J$  are the numbers of visible and hidden units, respectively. The RBM assigns a probability to each configuration  $(\mathbf{v}, \mathbf{h})$  using the energy function given by

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v}, \mathbf{h})}}{Z}, \quad (2)$$

where  $Z$  is a normalization factor obtained by summing up the energies of all the possible  $(\mathbf{v}, \mathbf{h})$  configurations:

$$Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}. \quad (3)$$

The conditional probabilities can be analytically computed as

$$p(h_j = 1 | \mathbf{v}) = \sigma \left( b_j + \sum_{i=1}^I v_i w_{ji} \right), \quad (4)$$

$$p(v_i = 1 | \mathbf{h}) = \sigma \left( a_i + \sum_{j=1}^J h_j w_{ji} \right), \quad (5)$$

where  $\sigma(x)$  is the sigmoid function; that is,  $\sigma(x) = 1/(1+e^{-x})$ .

The training process of the RBM can be described as follows. After the random initialization of the weights and biases, iterative training of the RBM on the training data is performed. Given the training data on the visible units  $\{v_i\}$ , the states of hidden units  $\{h_j\}$  are sampled according to (4). This step is called the positive phase of the RBM training. In the negative phase, the ‘‘reconstruction’’ of the visible units  $\{v'_i\}$  is obtained according to (5). The positive phase is once more conducted to generate  $\{h'_j\}$ . Afterwards, the RBM weights and biases can be updated by the contrastive-divergence (CD) algorithm [31] through gradient ascent, which can be formulated as

$$\begin{aligned} \Delta w_{ij} &= \varepsilon \left( \langle v_i h_j \rangle - \langle v'_i h'_j \rangle \right), \\ \Delta a_i &= \varepsilon \left( \langle v_i \rangle - \langle v'_i \rangle \right), \\ \Delta b_j &= \varepsilon \left( \langle h_j \rangle - \langle h'_j \rangle \right), \end{aligned} \quad (6)$$

where  $\varepsilon$  denotes the learning rate and  $\langle \cdot \rangle$  represents the mathematical expectation under the corresponding data distribution.

**2.1.2. Pretraining.** The DBN takes a layer-wise greedy learning strategy, in which RBMs are individually trained one after another and then stacked on the top of each other. When the first RBM has been trained, its parameters are fixed, and the hidden unit values are used as the visible unit values for the second RBM. The DBN repeats this process until the last RBM. Since pretraining is unsupervised, no label is needed. Unsupervised learning is believed to capture the crucial distribution of the data and can therefore help supervise learning when labels are provided. A batch-learning method is usually applied to accelerate the pretraining process; that is, the weights of the RBMs are updated every minibatch [32, 33].

**2.1.3. Fine-Tuning.** After the pretraining phase, the fine-tuning procedure is performed. A softmax output layer can be placed on top of the last RBM as a multiclass classifier, and the output-layer size is set to the same value as the total number of classes. To accomplish classification by utilizing the learned feature, we use the ordinary back-propagation technique through the whole pretrained network to fine-tune the weights for enhanced discriminative ability. Given

that the fine-tuning procedure is supervised learning, the corresponding labels for the training data are needed. After training, the predicted class label of a test sample can be obtained by forward propagation, in which the test data pass from the lowest-level visible layer through multi-RBM layers to the softmax output layer.

**2.2. LULC Classification Based on DBN.** To better understand the structure of the DBN-based LULC classification, a flowchart is given in Figure 2. To delineate the high variance and speckles of the PolSAR image, a neighbor window is used for local analysis, with the to-be-classified pixel placed at the center. Such neighbor window with size of  $winsize * winsize$  can be represented by a vector formed by the pixel values from the window. The original input feature for the DBN consists of the processed Pauli parameters, which are the diagonal elements  $(0.5|HH + VV|^2, 0.5|HH - VV|^2, \text{ and } 2|HV|^2)$  under the reciprocal assumption) of the coherency matrix with their logarithm form stretched by linear scaling [2]. One kind of Pauli feature in a window is reshaped in a vector by sequentially connecting each feature line. A Pauli vector of a day can then be formed by connecting the three Pauli feature vectors. For multitemporal analysis, the input to DBN can be formed by connecting the  $m$  dates’ Pauli vectors, with the dimension of  $winsize * winsize * 3 * m$ .

For the training of DBN, Pauli vectors of the training samples are assigned to the visible layer of the first RBM as input training features. With a layer-by-layer pretraining strategy, the spatiotemporal dependencies are successively encoded in the hidden layers  $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(n-1)}$ , and  $\mathbf{h}^{(n)}$ . In the output layer, the labels of the training samples are provided, and the weights of the DBN are fine-tuned in a supervised manner.

For the prediction, the input features of the test samples are prepared in the same way as that of the training samples. The classification labels for the test samples can be obtained from the forward propagation of the test features through the trained network.

### 3. Data and Experiment

The study area is located in northern Greater Toronto Area (GTA), Ontario, Canada. The ten major LULC classes in the study area are as follows: high-density residential areas (HD), low-density residential areas (LD), industrial and commercial areas (Ind.), construction sites (Cons.), Water, Forest, Pasture, golf courses (Golf), and two types of crops (Crop1 and Crop2).

Two fine-beam full polarimetric SAR images were acquired by the RADARSAT-2 SAR sensor on June 19, 2008, and July 5, 2008. The center frequency is 5.4 GHz, that is, C-band. The June 19 data were obtained from the descending orbit, whereas the July 5 data were obtained from the ascending orbit, as shown in Figures 3(a) and 3(b). The data from the ascending and descending orbits were expected to complement each other from two different look directions. A total of 4952065 pixels of the overlap between the two images were classified.

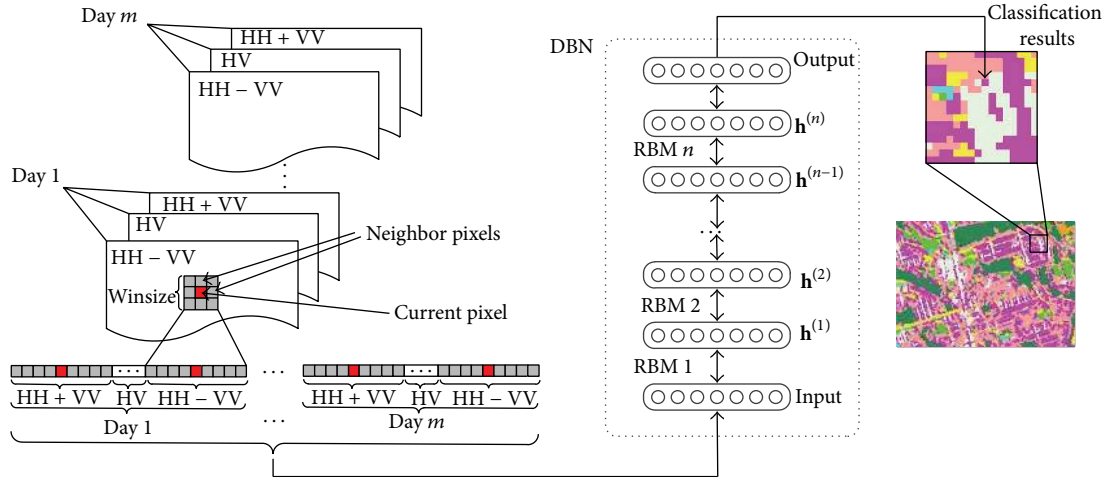


FIGURE 2: Flowchart of the proposed DBN-based classification approach.

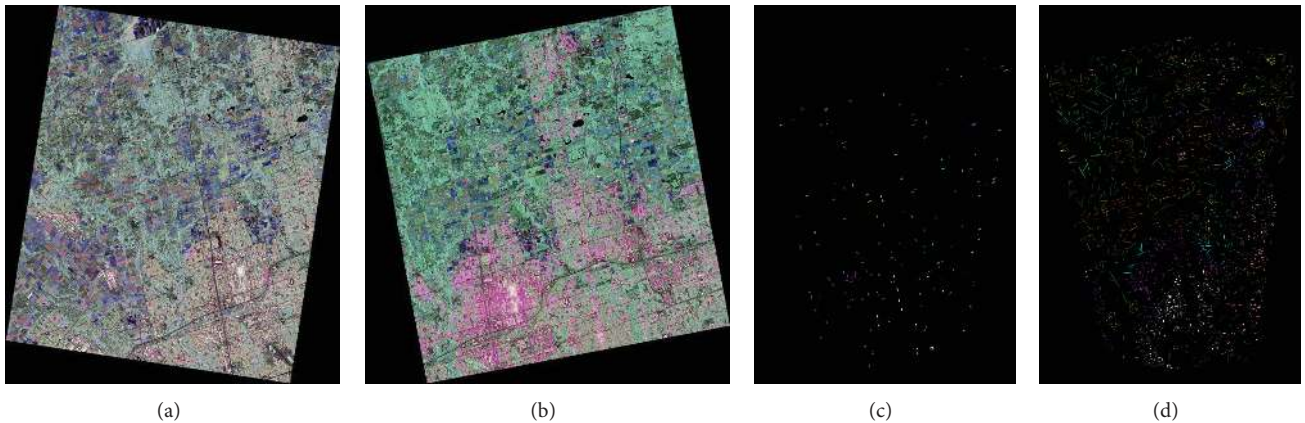


FIGURE 3: PolSAR images of northern Greater Toronto Area. (a) Pauli RGB image of RADARSAT-2 data on June 19, 2008. (b) Pauli RGB image of RADARSAT-2 data on July 5, 2008. (c) Training set. (d) Test set.

During the preprocessing, the multitemporal raw data were first orthorectified using the satellite orbital parameters and a 30 m resolution DEM. Then, they were registered to a vector file National Topographic Database (NTDB). A multilook process was further applied to generate the PolSAR features with the final spatial resolution of about 10 meters.

In the classification scheme, 19 subclasses were defined for the abovementioned 10 major land cover classes according to different scattering characteristics (e.g., the man-made structures have varying scattering appearance due to their distinctive shapes and directions). Approximately 1000 training pixels were assigned to each subclass. 120617 pixels evenly distributed over the classification area were randomly selected as the test samples. The training and test samples are visually shown in Figures 3(c) and 3(d), respectively.

The effective configurations of the DBN for detailed urban mapping were investigated. Comparisons with SVM, conventional neural networks (NN), and stochastic Expectation-Maximization (SEM) were conducted to assess the potential of our approach.

## 4. Results and Discussions

In this study, several experiments were conducted to validate the impact of different DBN configurations, including different network depths and hidden layer node numbers. To evaluate its classification efficiency, the DBN-based approach was compared with three other land cover methods: SVM, traditional neural networks (NN), and stochastic Expectation-Maximization (SEM). To quantitatively compare and estimate the capabilities of the proposed method, the overall accuracy (OA) and Kappa coefficient [34] were used as performance measurements.

The performance of the DBN-based classification method is sensitive to the neighbor window size. As the window size increases, more spatial dependencies could be captured by the DBN; thus, it is expected that better classification accuracy could be obtained with larger neighbor window size. Nevertheless, larger neighbor window size does not ensure better classification performance. Overly large window sizes could decrease the classification performance because bound areas tend to be confounded under an overlarge window. In

TABLE 1: DBN parameters setting.

Pretraining stage	
Learning rate	0.01
Number of epochs	50
Size of minibatch	100
Momentum	0.5 for the first 5 epochs, 0.9 thereafter
Weight decay rate	0.0002
Fine-tuning stage	
Learning rate	0.1
Number of epochs	20

the following experiments, the neighbor window size is set to  $11 * 11$ ; thus, the dimension of the input data would be  $11 * 11 * 3 * 2 = 726$ .

Several parameters of the DBN are listed in Table 1; some of these parameters are based on experimentation, while the others are based on the recommendation of Hinton [33]. All the hidden layers in the DBN have the same number of hidden units. For all the DBN depths mentioned below, only the hidden layers were counted.

**4.1. Effect of Network Depth.** We first examine how the DBN depth influences the classification performance. The number of hidden layers is one of the key factors to the deep learning strategy. On one hand, it is proved that additional RBM layer can yield improved modeling power [35]. A higher level of representation leads to potentially more abstract features [27]. On the other hand, Larochelle et al. [36] argue that unnecessary RBM layers may degenerate the generalization capability of the DBN because more layers engender a more complex network model with more parameters to fit. With relatively less training samples, complex models often cause the overfitting problem [35]. The best depth of the DBN is usually related to a specific application and dataset.

To find a proper network depth, DBN models with increased number of RBM layers (i.e., from one to four layers) were compared. Each DBN model had the same constant structure; that is, all the RBM layers had the same number of hidden neurons. Comparisons were also conducted by varying the number of hidden neurons from 100 to 600 per layer. The results in Figure 4 show that, regardless of the number of neurons, improved overall accuracies were all obtained by the two-layer DBN model. Although the comparisons were made only up to 4 layers, it is expected that, with more layers, the overfitting problem will become more serious, which will lead to worse results. As such, the depth of DBN was set to two layers in the following experiment.

**4.2. Comparison with Other Classification Methods.** To demonstrate the effectiveness of the proposed LULC classification method, a comparison was conducted with three other land cover classification approaches (i.e., SVM, conventional NN, and SEM). The same Pauli features as the DBN-based method were used in SVM and traditional NN. The SEM method [9] applied an adaptive Markov Random Field (MRF) to explore contextual information, and we used the same

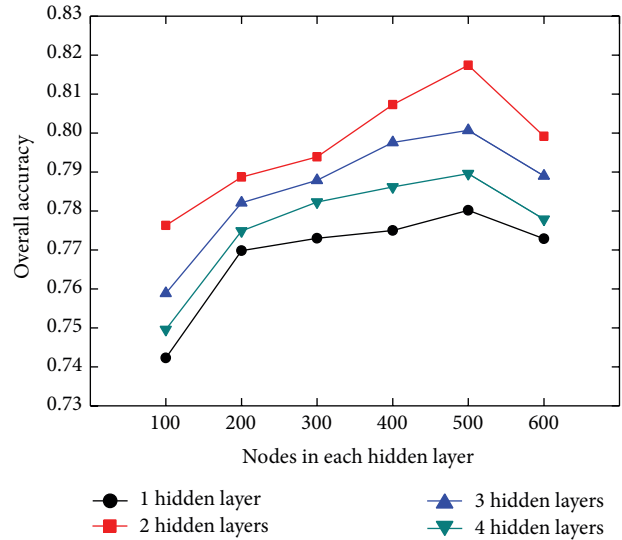


FIGURE 4: Impact of network depth.

settings reported there. The DBN contained two RBM layers, and each hidden layer had 500 units. Conventional NN had the same parameters as those of DBN; their only difference was that the weights of NN were not pretrained with unsupervised learning. The LIBSVM [37] toolkit was used as the implementation of SVM. SVM is a binary classifier, and the one-against-one strategy was used to convert the multiclass categorization problem to the binary classification problem. Experiments were performed using a radial basis function (RBF) kernel. The penalty term  $C$  and the RBF kernel width  $\sigma$  were selected using grid search within a given set  $\{2^{-10}, \dots, 2^{10}\}$ . The fivefold cross validation method indicated that the best validation rate was achieved when  $C = 32$  and  $\sigma = 2^{-7}$ . These parameters were then used to train the SVM model. The classification accuracies using different classification approaches are presented in Table 2, where  $P$  and  $U$  stand for the producer's accuracy and user's accuracy, respectively.

Table 2 shows that, among the four classification methods, the DBN method results in the best performance, with an overall accuracy (OA) of 81.74%. Tables 3, 4, 5, and 6 list the confusion matrices of the four classification methods in percent.

Obviously, SEM obtained the highest accuracies in most natural classes (Water, Golf, Pasture, Crop1, and Forest). However, it performed extremely badly in several man-made classes (LD, HD, and Ind.). Generally, SEM provided the lowest overall classification accuracy of 72.43%.

Although SVM attained higher producer's accuracies in Cons., LD, and Crop1, its overall accuracy was still below DBN by 5%. The improved classification accuracy by the DBN method mainly originated from the significant increase of Pasture and Crop2. Tables 3 and 6 show that the accuracy of Pasture was greatly improved owing to the decrease of the confusion with the Golf class. The improvement of the accuracy of Crop2 was mainly due to the decrease of the

TABLE 2: Comparison of different classification methods.

	SVM		NN		SEM		DBN	
	<i>P</i>	<i>U</i>	<i>P</i>	<i>U</i>	<i>P</i>	<i>U</i>	<i>P</i>	<i>U</i>
Water	0.8521	0.9169	0.7847	0.9560	<b>0.9668</b>	0.9733	0.8697	0.9052
Golf	0.8588	0.5364	0.8922	0.6048	<b>0.9245</b>	0.8346	0.8118	0.7727
Pasture	0.5776	0.8949	0.6095	0.9198	<b>0.8502</b>	0.8499	0.8139	0.8987
Cons.	<b>0.7639</b>	0.6879	0.6383	0.6657	0.7239	0.7750	0.7265	0.7899
LD	<b>0.6847</b>	0.8509	0.5771	0.8175	0.3160	0.7697	0.6703	0.8884
Crop1	0.9020	0.7548	0.7991	0.8971	<b>0.9617</b>	0.6497	0.8800	0.8804
Crop2	0.7965	0.8882	0.8615	0.7671	0.8306	0.8649	<b>0.8986</b>	0.8469
Forest	0.8703	0.9098	0.8908	0.9408	<b>0.9542</b>	0.7076	0.9095	0.9489
HD	0.7203	0.5830	0.7195	0.4570	0.6264	0.4898	<b>0.7824</b>	0.5867
Ind.	0.7593	0.7556	0.6817	0.7394	0.4135	0.5811	<b>0.7936</b>	0.7632
OA	0.7679		0.7437		0.7243		<b>0.8174</b>	
Kappa	0.7398		0.7119		0.6906		<b>0.7945</b>	

TABLE 3: Confusion matrix (in percent) of the SVM method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	85.21	3.70	0.07	0.96	0.00	0.02	0.00	0.00	0.01	0.18
Golf	11.85	85.88	26.41	5.24	1.36	3.27	2.51	2.46	0.05	0.48
Pasture	0.02	6.55	57.76	0.00	0.09	0.75	0.72	1.07	0.02	0.00
Cons.	2.92	2.00	0.34	76.39	0.00	0.03	10.87	0.00	0.47	0.20
LD	0.00	0.17	0.31	0.03	68.47	1.65	0.24	3.03	4.57	3.04
Crop1	0.00	0.55	6.80	0.83	5.34	90.20	1.53	0.20	4.31	1.12
Crop2	0.00	0.58	3.67	15.52	0.11	0.16	79.65	0.68	0.06	0.02
Forest	0.00	0.11	2.25	0.07	1.57	0.65	4.20	87.03	0.13	0.04
HD	0.00	0.07	2.27	0.09	15.65	3.12	0.21	3.71	72.03	18.98
Ind.	0.00	0.37	0.12	0.88	7.40	0.15	0.07	1.83	18.34	75.93

TABLE 4: Confusion matrix (in percent) of the NN method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	78.47	2.05	0.00	0.35	0.00	0.00	0.00	0.00	0.00	0.00
Golf	19.56	89.22	19.21	6.71	0.24	3.05	1.12	0.40	0.02	0.17
Pasture	0.00	1.93	60.95	0.77	0.13	2.68	1.18	0.13	0.01	0.01
Cons.	1.64	3.28	0.68	63.83	0.00	0.04	9.88	0.00	0.04	0.01
LD	0.00	0.14	0.01	0.00	57.71	5.25	0.02	2.34	5.61	2.07
Crop1	0.00	0.30	2.37	0.38	1.74	79.91	0.64	0.14	0.33	0.53
Crop2	0.15	2.45	14.48	26.70	0.71	3.13	86.15	0.94	0.19	0.02
Forest	0.00	0.47	1.96	0.18	2.06	0.64	0.98	89.08	0.16	0.09
HD	0.00	0.06	0.33	0.37	31.99	5.13	0.03	6.21	71.95	28.95
Ind.	0.17	0.10	0.03	0.71	5.41	0.18	0.00	0.77	21.69	68.17

TABLE 5: Confusion matrix (in percent) of the SEM method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	96.68	0.53	0.08	0.13	0.08	0.12	0.09	0.09	0.04	0.11
Golf	3.10	92.45	4.36	4.41	0.54	0.03	0.38	0.28	0.53	1.01
Pasture	0.08	5.40	85.02	1.15	2.11	1.32	4.82	0.75	0.69	0.07
Cons.	0.06	1.01	0.60	72.39	0.21	0.00	6.12	0.03	0.78	0.35
LD	0.00	0.00	0.01	0.03	31.60	0.16	0.03	0.36	6.54	3.18
Crop1	0.06	0.07	6.80	1.21	6.28	96.17	3.58	0.85	4.67	10.90
Crop2	0.04	0.35	2.55	20.16	0.94	0.22	83.06	1.21	0.73	0.43
Forest	0.00	0.15	0.53	0.32	34.84	1.69	1.89	95.42	2.43	0.53
HD	0.00	0.02	0.04	0.13	10.20	0.29	0.01	0.76	62.64	42.07
Ind.	0.00	0.01	0.00	0.06	13.20	0.00	0.01	0.24	20.95	41.35

TABLE 6: Confusion matrix (in percent) of the DBN method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	86.97	5.72	0.02	0.21	0.00	0.00	0.00	0.00	0.00	0.00
Golf	11.92	81.18	5.93	3.42	0.16	0.66	0.29	0.40	0.01	0.38
Pasture	0.11	10.55	81.39	0.31	0.03	2.28	0.55	0.17	0.09	0.00
Cons.	0.99	1.26	0.26	72.65	0.00	0.01	6.16	0.00	0.31	0.05
LD	0.00	0.09	0.27	0.03	67.03	2.22	0.23	1.95	3.35	1.46
Crop1	0.00	0.14	2.49	0.93	2.47	88.00	1.22	0.23	0.71	0.26
Crop2	0.00	0.66	7.20	21.05	0.23	0.63	89.86	1.70	0.12	0.00
Forest	0.00	0.22	1.11	0.19	1.70	0.69	1.39	90.95	0.15	0.08
HD	0.00	0.15	1.31	0.31	19.67	5.40	0.24	2.61	78.24	18.41
Ind.	0.00	0.04	0.02	0.90	8.70	0.11	0.05	2.00	17.04	79.36

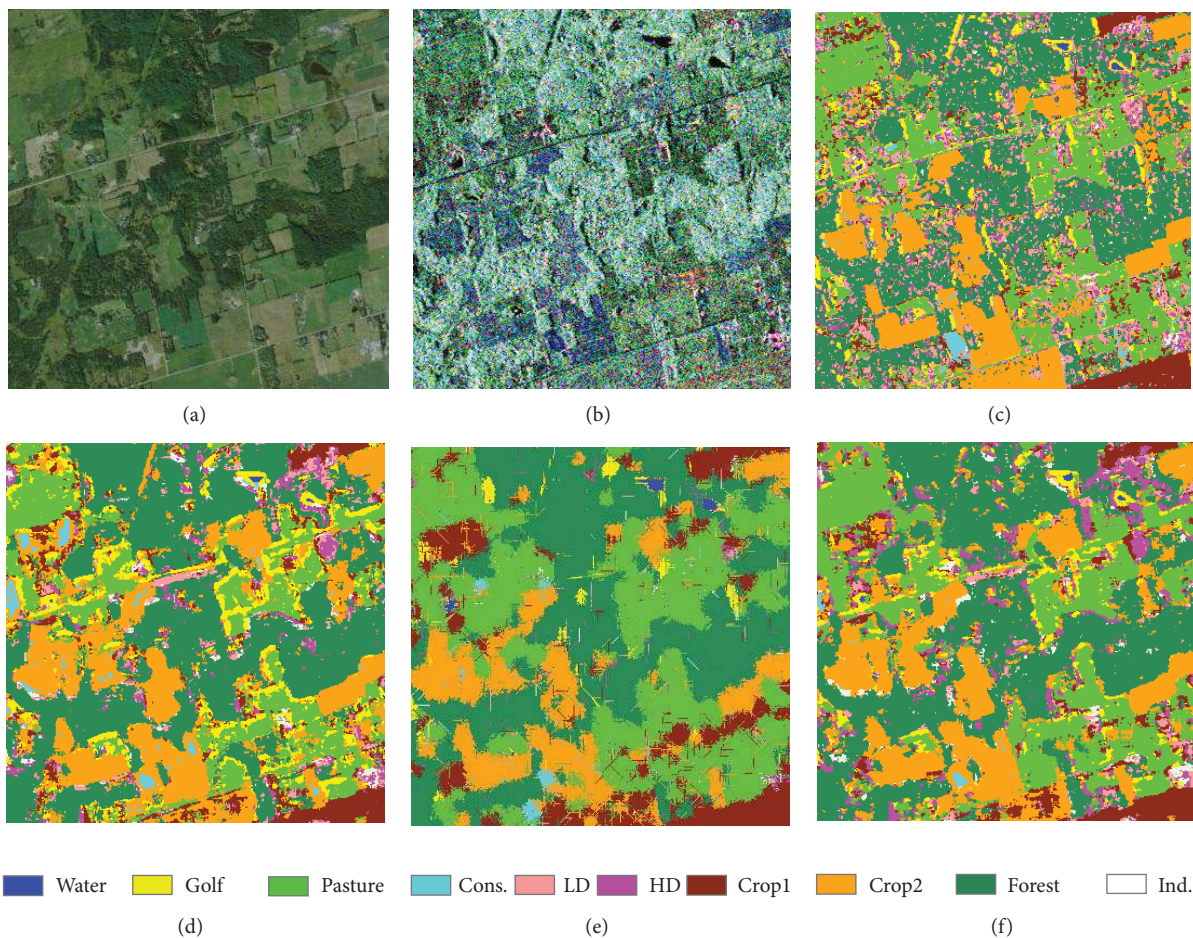


FIGURE 5: Zooming comparison of (a) Google Earth image and (b) PolSAR Pauli image and the classification results using (c) SVM, (d) NN, (e) SEM, and (f) DBN in a selected area.

commission to Cons. One plausible explanation for this improvement is that, with the effective features represented by the hidden layers, DBN could extract additional underlying dependencies and structures for the SAR data.

Compared with conventional NN, DBN obtained higher classification accuracies for almost all land cover types,

resulting in a notable increase in OA of 7%. The reason behind the superiority of DBN over NN is that, with an unsupervised pretraining process, more appropriate initial weights are assigned to the network, while the traditional neural network just sets random values for initial weights. The DBN-based method combines the advantages of both



FIGURE 6: Zooming comparison of (a) Google Earth image and (b) PolSAR Pauli image and the classification results using (c) SVM, (d) NN, (e) SEM, and (f) DBN in an Ind. area.

unsupervised and supervised learning; thus it can better distill spatiotemporal regularities from SAR data and improve classification performance.

The effects of different land cover classification methods are further illustrated in Figure 5. As can be observed in Figure 5, compared with SVM, the DBN method significantly reduces the misclassification of Forest. Compared with NN, DBN greatly decreases the misclassification of Pasture from Golf. Compared with SEM, DBN preserves the detail of residential areas. Figure 6 shows another example from an Ind. area. The figure shows that the DBN-based method provides classification map with more homogenous regions of the Ind. land cover type, which is more in line with reality.

## 5. Conclusion

A detailed urban LULC classification method based on the DBN model for PolSAR data is proposed. The effects of different network configurations are discussed. It is found that DBN with two hidden layers were appropriate for such detailed LULC mapping application. The experimental results demonstrate that the proposed method provides homogenous mapping results with preserved shape details and that it

outperforms other land cover classification approaches (i.e., SVM, NN, and SEM) in a complex urban environment. Our future work will focus on more deep learning models for SAR data to further improve the classification results.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work is mainly supported by the National Natural Science Foundation of China under Grants U1435219, 61125201, 61202126, 61202127, and 61402507.

## References

- [1] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 1, pp. 6–43, 2013.



- [2] X. Niu and Y. Ban, "Multi-temporal RADARSAT-2 polarimetric SAR data for urban land-cover classification using an object-based support vector machine and a rule-based approach," *International Journal of Remote Sensing*, vol. 34, no. 1, pp. 1–26, 2013.
- [3] P. Lombardo, M. Sciotti, T. M. Pellizzeri, and M. Meloni, "Optimum model-based segmentation techniques for multi-frequency polarimetric SAR images of urban areas," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1959–1975, 2003.
- [4] T. Macri Pellizzeri, "Classification of polarimetric SAR images of suburban areas using joint annealed segmentation and "H/A/ $\alpha$ " polarimetric decomposition," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 1-2, pp. 55–70, 2003.
- [5] Y. Ban and A. Jacob, "Object-based fusion of multitemporal multiangle ENVISAT ASAR and HJ-1B multispectral data for urban land-cover mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 1998–2006, 2013.
- [6] S. W. Myint, P. Gober, A. Brazel, S. Grossman-Clarke, and Q. Weng, "Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery," *Remote Sensing of Environment*, vol. 115, no. 5, pp. 1145–1161, 2011.
- [7] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 1, pp. 2–16, 2010.
- [8] U. C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen, "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 3-4, pp. 239–258, 2004.
- [9] X. Niu and Y. Ban, "An adaptive contextual SEM algorithm for urban land cover mapping using multitemporal high-resolution polarimetric SAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 4, pp. 1129–1139, 2012.
- [10] Y. Wu, K. Ji, W. Yu, and Y. Su, "Region-based classification of polarimetric SAR images using wishart MRF," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, pp. 668–672, 2008.
- [11] D. Gleich, "Markov random field models for non-quadratic regularization of complex SAR images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 3, pp. 952–961, 2012.
- [12] A. Voisin, V. A. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Classification of very high resolution SAR images of urban areas using copulas and texture in a hierarchical Markov random field model," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 1, pp. 96–100, 2013.
- [13] P. Yu, A. K. Qin, and D. A. Clausi, "Unsupervised polarimetric SAR image segmentation and classification using region growing with edge penalty," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1302–1317, 2012.
- [14] D. H. Hoekman, M. A. M. Vissers, and T. N. Tran, "Unsupervised full-polarimetric SAR data segmentation as a tool for classification of agricultural areas," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 2, pp. 402–411, 2011.
- [15] K. Ersahin, I. G. Cumming, and R. K. Ward, "Segmentation and classification of polarimetric SAR data using spectral graph partitioning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 1, pp. 164–174, 2010.
- [16] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 907–924, 2013.
- [17] L. Bombrun, G. Vasile, M. Gay, and F. Totir, "Hierarchical segmentation of polarimetric SAR images using heterogeneous clutter models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 726–737, 2011.
- [18] V. Akbari, A. P. Doulgeris, G. Moser, T. Eltoft, S. N. Anfinsen, and S. B. Serpico, "A textural-contextual model for unsupervised segmentation of multipolarization synthetic aperture radar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 2442–2453, 2013.
- [19] R. J. Dekker, "Texture analysis and classification of ERS SAR images for map updating of urban areas in the Netherlands," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1950–1958, 2003.
- [20] V. V. Chamundeeswari, D. Singh, and K. Singh, "An analysis of texture measures in PCA-based unsupervised classification of SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 2, pp. 214–218, 2009.
- [21] X. Niu and Y. Ban, "Multitemporal polarimetric RADARSAT-2 SAR data for urban land cover mapping through a dictionary-based and a rule-based model selection in a contextual SEM algorithm," *Canadian Journal of Remote Sensing*, vol. 39, no. 2, pp. 138–151, 2013.
- [22] B. Waske and J. A. Benediktsson, "Fusion of support vector machines for classification of multisensor data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 3858–3866, 2007.
- [23] N. Jones, "Computer science: the learning machines," *Nature*, vol. 505, no. 7482, pp. 146–148, 2014.
- [24] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.
- [25] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.
- [26] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.
- [27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [28] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [29] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [30] N. Lopes and B. Ribeiro, "Towards adaptive learning with improved convergence of deep belief networks on graphics processing units," *Pattern Recognition*, vol. 47, no. 1, pp. 114–127, 2014.
- [31] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [32] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, vol. 7700 of *Lecture Notes in Computer Science*, pp. 437–478, Springer, Berlin, Germany, 2nd edition, 2012.

- [33] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Tech. Rep., Department of Computer Science, University of Toronto, 2010.
- [34] W. D. Thompson and S. D. Walter, "A reappraisal of the kappa coefficient," *Journal of Clinical Epidemiology*, vol. 41, no. 10, pp. 949–958, 1988.
- [35] N. le Roux and Y. Bengio, "Representational power of restricted Boltzmann machines and deep belief networks," *Neural Computation*, vol. 20, no. 6, pp. 1631–1649, 2008.
- [36] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.
- [37] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

