

## Object-based Urban Land Use Classification using Deep Belief Network

Su Wai Tun, Khin Mo Mo Tun  
University of Information Technology  
Yangon, Myanmar

suwaitun@uit.edu.mm, suwaidun@gmail.com  
khinmomotun@uit.edu.mm, khinmo2htun@gmail.com

### Abstract

*Urban land use information is very important for urban planning, regional administration and management. Classification of urban land use from high resolution images remains a challenging task, due to the extreme difficulties in differentiating complex spatial patterns to derive high-level semantic labels. Deep learning is a powerful state-of-the-art technique for image processing including remote sensing images. 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. In this paper, deep belief network model is used to improve the performance of object-based land use classification. First, to achieve an object-based image representation, the original image is segmented into objects by graph-based minimal-spanning-tree segmentation algorithm. Second, spectral, spatial and texture features for each object are extracted. Then all features are put into deep belief network and the parameters of the network using training samples are trained. Finally, all objects are classified by network.*

**Keywords-** Classification, Deep Belief Network, Land Use

### 1. Introduction

Land use information is essential for urban planning and management and also provides a key input to urban and transportation models, and is essential to understanding the complex interactions between human activities and environmental change. In recent years, high resolution remote sensed images from satellites, planes, and unmanned aerial vehicle (UAV) have been widely used for classification. Information on urban land use within high resolution images is presented implicitly as patterns or high level semantic functions, in which some identical low-level ground features or object classes are frequently shared amongst different land use categories. This complexity and diversity of spatial and structural patterns in urban areas makes its classification into land

use classes a challenging task. It is very important to develop robust and accurate urban land use classification techniques by effectively representing the spatial patterns or structures lying in high resolution remotely sensed data. Pixel based image analysis (PBIA) has been a popular method to classify remote sensed images given its simplicity and high efficiency. The PBIA method cannot take full advantage of the texture/contextual information found in high resolution images thus the results display a salt and pepper effect after classification. Because of this problem, object-based image analysis (OBIA) has become a main method in land-use/land-cover (LULC) applications over the last decade. OBIA was presented to overcome the drawbacks of PBIA when classifying high resolution image.

Recently, deep learning has become the new hot topic in machine learning and pattern recognition, where the most representative and discriminative features are learnt end-to-end, hierarchically [6]. The DBN employs a hierarchical structure with multiple stacked restricted Boltzmann machines and works through a layer by layer successive learning process. This paper presents object-based land use classification based on deep belief network (DBN) to improve classification result.

### 2. Related Works

In literature, [1] land use classification method based on stack autoencoder has been proposed by Anzi Ding, Xinmin Zhou. This method is tested in GF-1 images with 4 spectral bands and spatial resolution of 8 m. They show that the method based on SAE is more accurate in classification result than support vector machine and back propagation neural network.

[2] Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel have proposed land cover classification method based on Deep recurrent neural networks. This proposed model has validated on two different data set showing that this framework efficiently deals with both pixel- and object-based classifications.[4]Deep convolutional neural network for land-cover classification method has proposed by Grant J. Scott, R. England, William A. Starms, Richard A. Marcum and Curt H. Davis.

[5] Chen and Guestrin published a new, regularized implementation of gradient boosting machines (GBM), called extreme gradient boosting classifier (Xgboost). It has made a very strong impact on the machine learning community, being the winning solution of most machine learning competitions. Then, Stefanos Georganos, Tais Grippa, Sabine Vahuysse, Moritz Lennert, Michal Shimoni evaluated the implementation of Xgboost for very high resolution object-based land use land cover urban classification. The results demonstrated that optimized Xgboost with a Bayesian model consistently outperforms random forest (RF) and support vector machines (SVMs) in different very high resolution data sets and classification schemes but at the cost of increased computational time.

[6]Ce Zhang, Isabel Sargent, Xin Pan, Huapeng Li, Andy Gardiner, Jonathon Hare proposed object-based convolutional neural network (OCNN) for urban land use classification. Their proposed method starts with an initial image segmentation to achieve an object-based image representation. They used Mean-shift segmentation ,as a nonparametric clustering approach, to partition the image into objects with homogeneous spectral and spatial information. Then they developed two CNN networks with different model structures and window sizes to predict linearly shaped objects (e.g. Highway, Canal) and general (other non-linearly shaped) objects. Then a rule-based decision fusion was performed to integrate the class-specific classification results. Their proposed OCNN method was tested on aerial photography of two large urban scenes in Southampton and Manchester in Great Britain. The classification accuracy and computational efficiency of their method outperformed the Pixel-wise CNN, contextual-based MRF and object-based image analysis SVM methods.

[10]Qi Lv, Yong Dou, Xin Niu, Jiaqing Xu, Jinbo Xu, and Fei Xia proposed 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. Their DBN-based method outperformed support vector machine (SVM), conventional neural networks (NN), and stochastic Expectation-Maximization (SEM) and produces homogenous mapping results with preserved shape details.

### 3. Theory Background

In machine learning, a deep belief network (DBN) is a generative graphical model, or alternatively a class of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer. When trained on a set of examples without supervision, a DBN can learn to probabilistically reconstruct its inputs. The layers then act as feature detectors. After this learning step, a DBN can be further trained with supervision to perform classification.

Restricted Boltzmann Machine, unsupervised learning, has the advantage of fitting the feature of the samples. So when we have an output of the hidden layer in a RBM, we can use it as the visible layer's input of another RBM. This process can be regarded as further feature extraction from the extracted feature of our samples. With this kind of thought, Hinton raised Deep Belief Network (DBN) in 2006, which is based on RBM. As the Figure 1 shows, by using the output of the upper RBM's hidden layer as the input of the lower RBM's visible layer, we get a Deep Belief Network. This DBN is stacked by three RBMs.

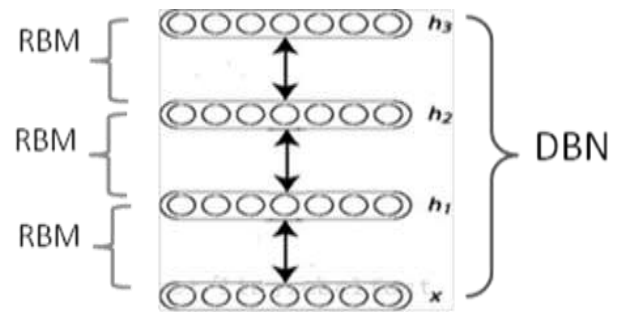


Figure 1. A DBN stacked by three RBMs

## 4. Object-based classification based on Deep Belief Network

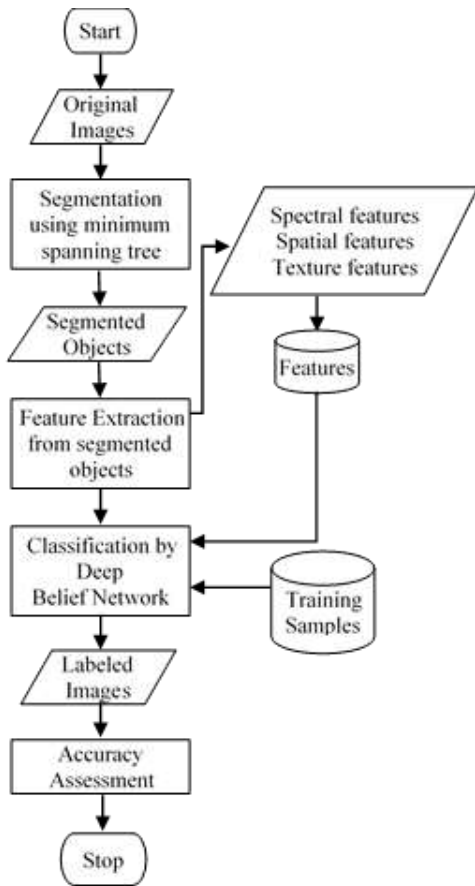


Figure 2. System Overview

### 4.1. Graph-based Minimal Spanning Tree Segmentation

Image segmentation is a process of partitioning a raster image into multiple segments. Many segmentation algorithms have been proposed, such as watershed, level set, etc. In this paper, a graph-based minimal-spanning-tree segmentation (GBS) method is used. The main concept in GBS is that each pixel of an image is considered as a vertex of an undirected graph, and a four neighborhood adjacent pixel-pair as an edge. The weight of each edge is calculated by the dissimilarity between two vertexes using the dissimilarity function. Then, all edges must be inserted into a minimal-spanning tree (or minimum-spanning-forest) in ascending order of its weight. The whole insertion progress is the merge-process and each tree represents an object. When the weight of the trees to be merged is smaller than the threshold, the merge-process will stop. At the end, some post segmentation procedures are implemented. Once the minimal-spanning-tree (or minimum-

spanningforest) is formed, there always exist small regions (objects) in it. These are called “silver objects,” and could be eliminated by forcibly merging them into their largest neighboring object. Another significant operation after segmentation is to convert results from a labeled raster image into vector data format (such as ESRI Shapefile format), referred to “Polygonization” (or “Vectorization”). The scale of the segmentation result is decided by a threshold, but it is hard to select it robustly. Therefore a trial and error method is used to get an appropriate segmentation scale.

### 4.2. Feature Extraction

To identify the category of an object automatically by supervised classification method, the features of object should be extracted. Features of an object are calculated by all the pixels and shape (contour) of an object. Three types of features are taken into consideration: spectral feature, spatial features, and texture features. They will be separately described in three tables.

1) Spectral features: Spectral features are the statistical attributes of an object found in the spectral bands of an image. A spectral feature can be an attribute of a single band (such as mean-value of the band) or all bands (such as brightness). All spectral features taken into consideration are listed in Table 1.

2) Spatial features: Spatial features depict the position and geometry information for an object, and calculated from the contour of the object polygonized from the pixels. All spatial features are listed in Table 2.

3) Texture features: Texture features include texture information of all pixels within the object. Image-texture refers to particular frequencies of change in tones and their resulting spatial arrangements. Haralick features for gray level co-occurrence matrix (GLCM) and gray level difference vector (GLDV) is used as the texture feature as listed in Table 3.

**Table 1. Spectral Features**

Name	Target	Description
Brightness	All bands	Mean value of all an object's pixels in all of a digital numbers spectral bands.
Mean	Single band	Mean value of an object's all pixels' digital numbers
Standard Deviation	Single band	Standard deviation of an object's all pixels' digital numbers
Max	Single band	Max value of an object's all pixels' digital numbers
Min	Single band	Min value of an object's all pixels' digital numbers
Mean of inner border	Single band	Mean value of pixels' digital numbers on the inner border of an object
Skewness	Single band	The Skewness feature describes the distribution of all the image layer intensity values of all pixels that form an object. A normal distribution has a skewness of zero

**Table 2. Spatial Features**

Name	Target	Description
Area	Contour	The area of the object's region
Asymmetry	Contour	The Asymmetry feature describes the relative length of an image object, compared to a regular polygon
Border Index	Contour	The border index feature describes how jagged an image object is; the more jagged, the higher its border index
Border Length	Contour	Border Length of an image object's region
Compactness	Contour	The Compactness feature describes how compact an image object is
Elliptic Fit	Contour	The elliptic fit feature describes how well an image object fits into an ellipse
Elongation	Contour	Elongation describes the ratio of the long side and the short side of the minimum-bounding-rectangle of an object
Radius of largest enclosed ellipse	Contour	The radius of largest enclosed ellipse of the object
Radius of smallest enclosing ellipse	Contour	The radius of smallest enclosing ellipse of the object
Rectangular Fit	Contour	The similarity of an image object to a rectangle
Roundness	Contour	The similarity of an image object to a circle
Shape Index	Contour	The smoothness of an image object border
X Center	Contour	X-coordinate of the center of an object
X Max	Contour	Maximum x-coordinate of an object
X Min	Contour	Minimum x-coordinate of an object
Y Center	Contour	Y-coordinate of the center of an object
Y Max	Contour	Maximum y-coordinate of an object

	r	
Y Min	Contour	Minimum y-coordinate of an object

**Table 3. Texture Features**

Name	Target	Description
Homogeneity	GLCM	The GLCM is a tabulation of how often different combinations of pixel gray levels occur in a scene. A different co-occurrence matrix exist for each spatial relationship
Contrast		
Dissimilarity		
Entropy		
Ang. 2nd Moment		
Mean		
Standard Deviation		
Correlation		
Ang. 2nd Moment	GLDV	The GLDV is the sum of the diagonals of the GLCM. It counts the occurrence of references to the neighbor pixels' absolute differences.
Entropy		
Mean		
Contrast		

### 4.3 Classification of Deep Belief Network

The DBN architecture and the general methodology included in DBN machine learning are described in this section. DBN is a multilayered architecture that consists of one visible layer and multiple hidden layers. The visible layer of a DBN accepts the input data and transfers the data to the hidden layers to complete the learning process [9].

---

#### Algorithm 1 Deep Belief Network Model

---

**Input :** Input Data  $D$ , Maximum number of layers  $ML$ ,  
 Number of Neuron for Each Layer  $N$ ,  
 Maximum number of Epochs  $ME$

Initialize  $D$ ,  $ML$ ,  $N$ ,  $ME$

**For** Layer = 1:  $ME$  **do**

Train Network using RBN learning rule  
 Save weights of hidden-visible connections  
 and biases

**End For**

Back Propagation Classification

**Output:** Labeled Data

---

The overall learning process of DBN model is described in algorithm 1. As shown in algorithm 1, the input data, total number of hidden neurons in each hidden layer and maximum number of epochs for the model training process are required and initialized before the start of the DBN training process. Each layer of DBN is trained using the restricted Boltzmann machines (RBM)

learning rule with two learning steps: positive and negative phases. In positive learning phase, data are transferred from bottom visible layer to hidden layer and the probability of generating hidden units are determined as  $p(h \setminus v, W)$ . In negative phase, a reconstruction of the data from previous visible layer are operated and the probability of generating visible units are determined as  $p(v \setminus h, W)$ . The data vector is used as an input to the visible units. After a maximum number of training epochs, the repetitive positive and negative phases in the training of RBM layers will result in trained weights and generated visible units. The overall system function of DBN learning procedure can be expressed as the probability of generating a visible vector ( $v$ ) as a function of weights and hidden vectors based on RBM learning rule [8]. The probability of generating a visible vector  $p(v)$  by DBN learning process can be formulated using the probability of generating visible units in the reconstruction phase of the previous epoch and the probability of generating hidden units in the positive phase of the current epoch as

$$p(v) = \sum_h p(h \setminus v, W) p(v \setminus h, W) \quad (1)$$

An iterative process from a lower layer to a higher layer continues till the maximum number of layers is trained. Each RBM is individually trained and the weights and biases are saved during the DBN training process. At the end of the training process, the data is transferred from bottom visible layer (data layer) to higher invisible layers throughout the DBN architecture [9]. The DBN layer by layer training is an unsupervised learning process that cannot provide class labels of the training data. The label information of the training data will be used during the back-propagation training.

**4.3.1. Stacked RBM learning.** DBN is constructed with stacked RBMs. Training of the DBN model is completed through training of each RBM structure using RBM learning rule. Each RBM unit consists of two layers. There are a number of neurons in each layer and there is no synaptic weight connection between neurons within the same layer [7].

---

**Algorithm 2** Stacked RBM Model

---

**Input:** Input Data  $D$ , Maximum Number of Epochs  $ME$ ,  
Number of Hidden Layers and Batches  $Numb$   
Initialize Symmetric Weight and Biases  
**For** Batch =1:  $Numb$  do  
**For** Epoch =1:  $ME$  do

Learn Positive Phase using Eq : (2)  
 $P(h_j = 1 \setminus v) = \text{sigm}(-b_j - \sum_k v_k w_{jk})$

Learn Negative Phase using Eq: (3)  
 $P(v_k = 1 \setminus h) = \text{sigm}(-b_k - \sum_j h_j w_{jk})$

**If** Epoch < 5, **then**

Momentum = Initial momentum

**Else**

Momentum = Final momentum

**End If**

Update Weights and Biases

**End For**

**End For**

---

The iterative learning process for one RBM unit is described in algorithm 2. In this algorithm, the synaptic weights and biases of all neurons in each RBM layer are initialized at the beginning of the training process. After that, the RBM unit will be trained repeatedly with input training data. The training dataset is often divided into mini-batches with a small number of data vectors and weights are updated after treating each mini-batch. Each training epoch consists of two phases, the positive phase and negative phase. The positive phase transforms the input data from visible layer to the hidden layer. In negative phase, a reconstruction of the neurons of the previous visible layer is operated. The positive phase of RBM learning can be denoted mathematically as

$$P(h_j = 1 \setminus v) = \text{sigm}(-b_j - \sum_k v_k w_{jk}) \quad (2)$$

During the states of neurons in visible layer neurons in visible layer are reconstructed, the negative phase can be denoted mathematically as

$$P(v_k = 1 \setminus h) = \text{sigm}(-b_k - \sum_j h_j w_{jk}) \quad (3)$$

where  $h_j$  and  $v_k$  are the states for the  $j$ th neuron in hidden layer and the  $k$ th neuron in the visible layer respectively. The visible and hidden layer neurons are binary stochastic neurons with binary states 0 or 1, which representing on and off conditions of the neurons in the learning process.

After learning process for both positive and negative phases, synaptic weights and biases can be updated based on state vectors of neurons in both hidden layer and visible layers [8]. The update of synaptic weight,  $w_{jk}$ , can be denoted as

$$\Delta w_{jk} = \delta((v_k h_j)_{data} (v_k h_j)_{recon}) \quad (4)$$

where  $\delta$  is a value between 0 and 1, denoting the learning rate;  $(v_k h_j)_{data}$  is the pairwise product of the state vectors for the  $j$ th neuron in the hidden layer and the  $k$ th neuron in the visible layer after positive phase learning process whereas  $(v_k h_j)_{recon}$  denotes the pairwise product after the negative phase learning process for reconstruction of the visible layer. The same learning rule is utilized for bias updating, but individual hidden and visible units are used instead of pairwise products [8]. To stabilize the RBM learning process, a momentum is usually used in updating the synaptic weights and biases. With momentum, the weight update,  $\Delta w_{jk}$ , at the current epoch and formulated as

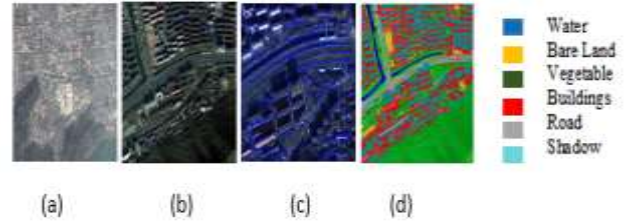
$$[\Delta w_{jk}]_n = (m[\Delta w_{jk}]_{n-1}) + \delta((v_k h_j)_{data} - (v_k h_j)_{recon}) \quad (5)$$

The initial and final momentums utilized in the RBM training process are 0.5 and 0.9 respectively [8]. The learning parameters such as weights and biases of each RBM in the DBN model will be continuously optimized until a maximum number of training epochs are reached. This completes the training of one RBM and the process will be continued until all RBMs in the DBN structure are trained.

**4.3.2. Back-propagation learning.** After layer by layer learning process, the next step of the DBN training is the supervised learning process that will be completed by the back-propagation training algorithm. The supervised learning uses labeled data for the training of the DBN model. Unlike the unsupervised DBN training process, which considers all DBN layers simultaneously, the back-propagation training is continued until the network output reaches the maximum number of epochs. After the supervised back-propagation training process, the trained DBN model can be further fine-tuned to improve classification accuracy through fine-tuning algorithms.

## 5. Datasets and Experimental Result

Experiment is conducted on World View-2 dataset including multispectral image which was obtained on January 13, 2010. The spatial resolution of multispectral image is 1.8 meter. The area is located in the Xihu District Hangzhou, China at  $30^{\circ}14'05.234''N$ ,  $120^{\circ}6'01.677''E$ , covering an area of approximately 312.43 km. The dataset contains three spectral bands, which represent the red, green, blue band separately. The task in the experiments was to classify all pixels in image into six categories: water, bare land, vegetable, buildings, road, shadow. Part of the training datasets are shown in Fig 3.

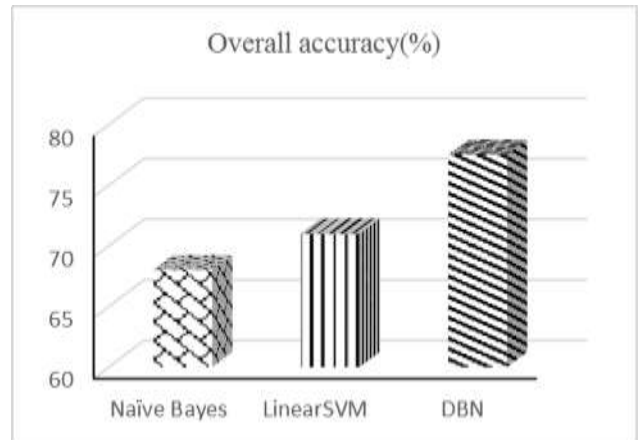


**Figure 3. (a) The original image of World View-2 (b) Preprocessed image (c) Objects generated by GBS algorithm (d) Training samples with the original image (part of the sample)**

In the experiments, preprocessing and segmentation of the image are implemented sequentially as shown in Fig 3. Then 181 features (19 spectral features, 18 spatial features and 144 texture features) are generated. Finally, objects are classified by the deep belief network, which was trained using training dataset and testing dataset. In this experiment, the proposed object-based classification approach based on DBN is compared with other approaches: Bayes (Naïve Bayes) and linear support vector machine (linearSVM). Most procedures are the same to object-based classification approach using DBN except classification part. The overall accuracy of different methods are shown in Fig 4.

**Table 4. overall accuracy of Naïve Bayes, Linear SVM, DBN**

Overall accuracy (%)	
Naïve Bayes	68
Linear SVM	71
DBN	77.5



**Figure 4. Object-based classification method based on DBN compared to other classification methods**

To evaluate the effectiveness and generalization of the



proposed classification approach based on DBN, the same experiment on the World View-2 dataset. The overall accuracy of each classifier as shown in table and figure which suggest that proposed classification approach based on DBN is more accurate than Bayes (Naïve Bayes) and linear support vector machine(linearSVM) on the World View-2 dataset.

## 6. Conclusion

This paper presented a object-based land use classification approach based on DBN. The results demonstrate that classification approach based on DBN outperforms Bayes and LinearSVM. In the existing object-based land use classification approach, the DBN model has not been used yet. Therefore, to improve the accuracy of the object-based land use classification approach, the DBN is used.

## 7. References

- [1]Anzi Ding, Xinmin Zhou, “Land-use Classification with Remote Sensing Image Based on Stacked Autoencoder”, International Conference on Industrial Informatics, 2016.
- [2]Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel, “Land Cover Classification via Multitemporal Spatial Data by Deep Recurrent Neural Networks”, IEEE Geoscience and Remote Sensing Letters, 2017.
- [3]Grant J.Scott, Matthew R. England, William A. Strams, Richard A.Marcum and Curt H. Davis, “ Training Deep Convolutional Neural Networks for Land-Cover Classification of High-Resolution Imagery”, IEEE.
- [4]Liangpei Zhang, Lefei Zhang, Bo Du, “ Deep Learning for Remote Sensing Data” , IEEE Geoscience and Remote Sensing Magazine” , 2016.
- [5] Stefanos Georganos, Tais Grippa, Sabine Vahuyse, Moritz Lennert, Michal Shimoni , “ Very High Resolution Object-Based Land Use-Land Cover Urban Classification Using Extreme Gradient Boosting”, IEEE Geoscience and Remote Sensing Letters, 2018.
- [6] Ce Zhang, Isabel Sargent, Xin Pan, Huapeng Li, Andy Gardiner, Jonathon Hare, “ An Object-based Convolutional Neural Network (OCNN) for Urban Land Use Classification” , 2018.
- [7] Lee H, Grosse R, Ranganath R, Ng AY, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations “ International conference on machine learning. Montreal, Canada, 2009.
- [8] Hinton GE. “A practical guide to training restricted Boltzmann machines. Momentum”, 2010.

[9]Hinton GE, Osindero S, Teh YW.” A Fast learning algorithm for deep belief nets”, Neural Computation, 2006.

[10]Qi Lv, Yong Dou, XIn Niu, Jiaqing Xu, Jinbo Xu and Fei Xia, “ Urban Land Use and Land Cover Classification Using Remotely Sensed SAR Data through Deep Belief Networks ”, Journal of Sensor, 2015.