American Journal of Environmental Sciences 5 (6): 706-713, 2009
ISSN 1553-345X
© 2009 Science Publications

Change Detection Using Neural Network with Improvement Factor in Satellite Images

[1]M.A. Fkirin, [1]S.M. Badwai and [2]Sayed A. Mohamed
[1]Department of Industrial Electronic Engineer and Control,
Faculty of Electronic Engineering, Menoufia University, Cairo, Egypt
[2]Department of Data Reception Analysis and Receiving Station Affairs,
National Authority for Remote Sensing and Space Sciences 23 Joseph Broz Tito St.,
El-Nozha EL-Gedida, Cairo, Egypt

Abstract: Problem statement: The aim of this study is to investigate the applicability of using the neural network techniques in change detection of remotely sensed data. **Approach:** In addition, the tuning parameters of the network, such as encoding the output classes, adding the momentum term and learning rate, are investigated in order to achieve best network performance. **Results:** Neural network-based change detection system in this study is implemented using back propagation-training algorithm. This trained network is designed to be able to detect efficiently any variation between two images and provide adequate information about the type of changes. In an effort to meet these requirements, neural network scheme with improvement factor, leaning rate and momentum term is proposed to monitor environmental changes in Toshka area, Egypt. Two sets of satellite images with different dates are used, the first set contains of two sample satellite images, the second set of images acquired on 1984, 2000 and 2003. **Conclusion/Recommendations:** Comparing the output of the proposed model with the mostly used change detection techniques; ratio and classification, results show a great potential as the proposed scheme was able to identify not only the changed and non-changed area but also it was capable to identify the nature of these changes.

Key words: Change detection, neural network, back propagation, Toshka area, Improvement factor, LANDSAT-7 data

INTRODUCTION

Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times[11]. It involves the ability to quantify temporal effects using multi-temporal data sets[15]. Some of the major applications of change detection, in images obtained from earth orbiting satellites, are land-use change analysis, monitoring of shifting cultivation, assessment of deforestation, damage assessment, disaster monitoring and the acquisition of military geographic information. Besides, many other applications can use change detection such as security systems and quality control.

A number of problems can be posed as change detection problems. This study presents an overview and assessment of the conducted digital change detection techniques using remotely sensed data, giving examples of their use in both environmental and military application. It cites also a number of comparative studies. Development of an efficient algorithm for detecting changes in satellite images suitable for all land-use is required. In an effort to meet this requirement, this study was directed toward design, develop and implement an algorithm for spatial change information extraction system based on artificial neural networks.

Many papers, classified from different viewpoint, Among the various ANN models developed and implemented for image classification, the multilayer perceptron is the most widely used network[3,7] have summarized the current methods of using Neural network.

This study proposes a change detection technique of remotely sensed data using neural networks. (Abstract, introduction, results and discussion, conclusion, and references)

Neural network model: There are many different types of neural networks; from relatively simple to very complex. Among them feed-forward, multi-layer neural networks which are one of the most popular designs for

Corresponding Author: M.A. Fkirin, Department of Industrial Electronic Engineer and Control, Faculty of Electronic Engineering, Menoufia University, Cairo, Egypt

neural networks. Network learning algorithms play an important role for the neural network in change detection schemes. There is no unique learning algorithm for the design of neural networks. Rather, we have a kit of tool represented by a variety of learning algorithms, each of which offers advantages of its own. Basically, learning algorithms differ from each other in the way in which the adjustment to a synoptic weight of a neuron is formulated using Back-Propagation learning Network (BPN)[2,4].

The back-propagation learning algorithm is used to train the weights in the network and updating the weights of a multilayered network which undergo supervised training[5]. has developed a technique for adapting the weights and then in 1986, Rumelhart *et al.*[14] had improved it into neural network[6]. This improvement of weight adaptation rule is known as back-propagation.

Back-propagation algorithm[18] used supervised learning approach due to the fact that in BPN, the target output vectors are defined earlier in the system. The learning process begins with the presentation of an input pattern to the BPN. In which, the net total input is found using the standard summation of products as defined in Eq. 1 below:

$$\Delta w_{ij}\ (n+1) = \mu\delta_j o_i + \alpha\Delta w_{ij}\ (n) \qquad (1)$$

Where:

(W_{ij}) = The weight associated with the connection between neuron j and I
(o_i) = The output of neuron i in output layer, (μ) Learning rate
(δj) = Output error from last cycle
(α) = Momentum
(n) = Number of cycles

The delta rule algorithm[16] will adjust the weights leading to the output units. It is affected by the variation of the η learning rate and α momentum. Each factor can be changed independently of the others to obtain the best results. In some experiments we changed two factors simulations to investigate their effects on the results.

In the Eq. 1, the term α is introduced to add in a proportion of the previous weight change, this is due to reduce the likelihood of the weight changes to oscillate. Therefore, the weight change for pattern n+1 is dependent on the weight change for pattern n. Those steps discussed above were repeated for each pattern defined in the systems. The patterns were continually presented to the network, epoch after epoch. This step is conducted until during one epoch, all outputs for each pattern are within tolerance[16].

Variable learning rate: The back-propagation error surface usually consists of a large amount of flat regions as well as extremely steep regions (Hush and Horne, 1993), so there is difficulty how to choose an appropriate value of learning rate. As such, the back-propagation algorithm with a variable learning is desirable[1].

Momentum: This is a modification based on the observation that convergence might be improved if we could smooth out the oscillations in the trajectory. Among the used techniques a low-pass filter shows a promising result[9,10].

Improvement factor: The response of the neural network to noise is an important factor in determining its suitability to a given application. A data set used to train a neural network may have random speckles in it. In the process of training, a matrix is applied to the neural network to see how well the network has learned the training data. The performance is compared with and without the adaptation of the Improvement Factor (IF) to investigate its importance[13].

MATERIALA AND METHODS

Proposed scheme: The proposed scheme is designed to accept any number of digital images (two images in our case) and produce changed image where changed and non-changed areas are assigned to different codes. Different types of changes and non-changes are included as prescribed in the learning phase. The system can be divided by functionality into the following three modules as illustrated in Fig. 1.

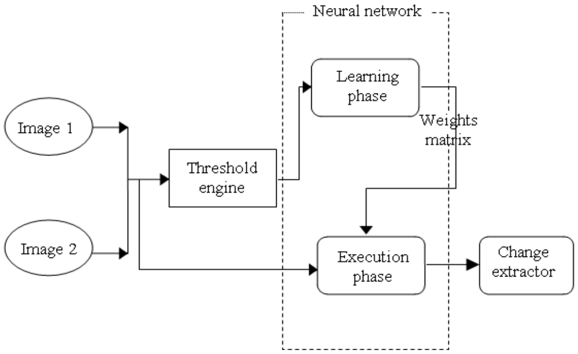


Fig. 1: This block diagram of proposed change detection system

Table 1: Ample of the training file used for testing data set 1

First sample image			Second sample image			Desired output		
0.42	0.56	0.67	0.44	0.50	0.60	1	0	0
0.45	0.56	0.53	0.16	0.08	0.06	0	1	0
0.44	0.59	0.68	0.44	0.53	0.49	0	0	1

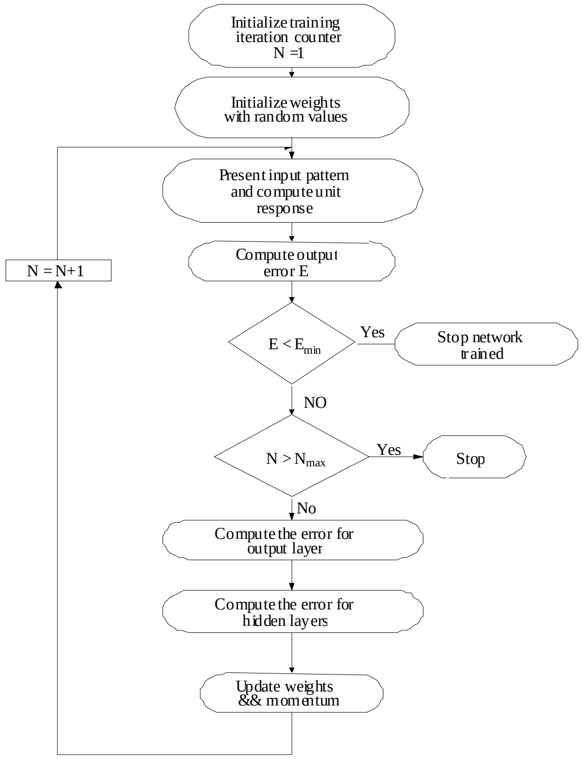

Initialize training iteration counter N =1
Initialize weights with random values
Present input pattern and compute unit response
Compute output error E
N = N+1
E < E_{min}
Yes
Stop network trained
NO
N > N_{max}
Yes
Stop
No
Compute the error for output layer
Compute the error for hidden layers
Update weights && momentum

Fig. 2: This flow chart for learning phase

Sample extractor (threshold engine): The two images are inputs to this part of the system and arbitrary distributed training sets are extracted from both images and compared with each other. Different types of changed and non-changed area are evaluated and assigned to different binary codes. The normalized pixel values of the training sets represent the input patterns and the output code represents the expected output. Table 1 shows the arrangement construction of entering process assuming the two inputs represented by satellite images and the desired output are set arbitrary to binary codes.

Neural network engine: The network will adapt the matrix of random weights in order to assign the input pattern to the expected output in forward feed-back direction[18] as shown in Fig. 2 and then the set of computed weights are used in forward path to classify the rest of images. These are done in two phases:

- Learning phase
- Execute phase

Learning phase: In this stage, the network weights are randomly generated and the training process will try to adjust the weights so that the actual output comes out closer to the expected output[8] according to the following procedures:

One of the patterns to be learned will be put on the input units. Values of the output of hidden layers units and output layer units are calculated.

The errors on the output layer units are calculated.

The delta rule algorithm[17] will adjust the weights leading to the output units. The errors on the hidden units are calculated. The weights leading into the hidden layers will be adjusted.

Steps from 1-6 are repeated for all the input patterns.

Steps from 1-7 are repeated until the actual output is closer to desired output.

These procedures of learning are summarized in Fig. 2.

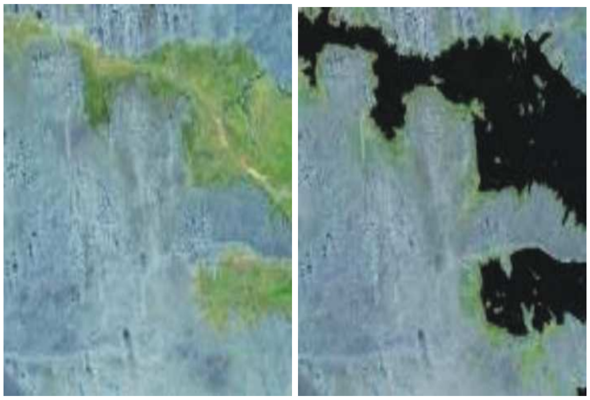
Execution phase: The two images are executed by the network. One pixel from the first image and the corresponding one from the other image at a time and a set of precalculated weights matrix, in training phase, is used to calculate the output values in forward propagation[12,14].

Change extractor: In this part the values of output neurons for each input vector are compared to each other and the order of neuron with the highest value with respect to the other neurons (winner neuron) represents the output class for the given input vector.

RESULTS

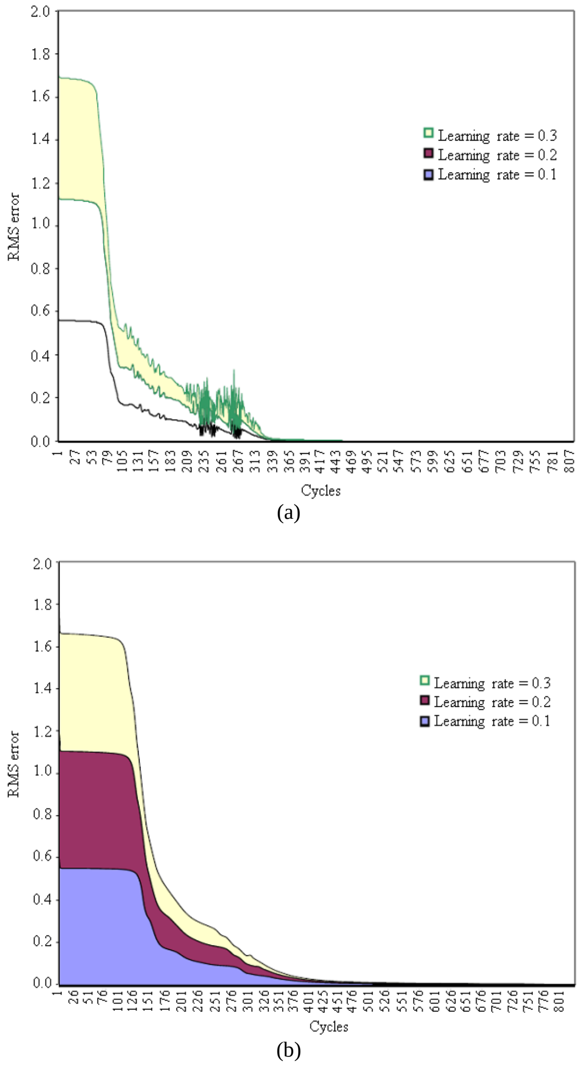
One of the most important aspects of any learning paradigm in neural network is how it scales according to problem size and complexity. To carry out this task with known training error and pre-specified maximum number of training updates, the convergence of the back-propagation algorithm is investigated, for both sample sets of satellite data, with respect to:

- Criticality of the initial weight set
- Number of hidden neurons
- Number of hidden layers
- Learning rate and momentum
- Improvement factor



(a) (b)

Fig. 3: (a) Sample image (1); (b) sample image (2)



(a)

(b)

Fig. 4: (a) Momentum = 0.0; (b) momentum = 0.1

Table 2: Represents the changes by using IF (data set 1)

	Color	Area (Km^2)	Area (Km^2)
Land cover changes		with IF	without IF
Class 1 desert to desert (no-change)	Black	158.680	156.052
Class 2 vegetation to water (change)	Blue	67.1724	67.2606
Class 3 desert to vegetation (change)	Green	29.0718	31.6116

A C++ program is implemented to scale the neural network parameters to obtain the best possible results for the data sets. The output of the program is the scaled neural network though adjusting the following parameters:

- Variable learning rate: Control the speed of the learning process
- Momentum: Increase the learning stability and reduce error
- Improvement factor: Increase the quality of the images by smoothing the range of pixel values

For each parameter, a careful designing phase was carried out in order to assess the best performances provided by neural network. The trials are summarized as follows:

- For the weights parameter, we carried out different trials with different randomized weight values. For the network structure, over 100 different topologies from one to three hidden layers are tested
- For both the learning rate and momentum parameters, we carried out different trials over the whole range from 0-0.9
- For the improvement factor parameter, different values were added to the training set

The Accuracy (AC) of any class is defined as the ratio between the positively true (pixels classified to be in a class and are truly in that class) and all pixels that are used as ground truth of this class.

Data set 1: This data set consists of two samples of the used satellite images of the Toshka area. Each image consists of 515 column and 550 row with 256 gray level assigned to the three main colors (red, green and blue) as shown in Fig. 3.

Data set 2: Three satellite images represent Toshka area are used. The first image was acquired on June 1984 from the LANDSAT-7 TM data. The second image on December 2000 and third image on March 2003 from LANDSAT-7 ETM+ data. The pixel size of the three images represents 30 m in the ground. Each image consists of 2962 column and 3220 row with 256 Gray level assigned to the three main colors red, green and blue. The training file used to train the network consists of 3266 input.

Experiment 1: The two samples of satellite images (data set1) are normalized to the range from zero to one, by dividing the pixels values by 255. The training file used to train the network consists of 3266 input vector samples, representing changed and non-changed areas, approximately 0.002% of the inputs and the output is assigned to three different classes namely (Table 2):

- Non-changed area (desert-desert)
- Changed area (vegetation-water)
- Changed area (desert-vegetation)

Figure 4a shows the Root Mean Square (RMS) error versus number of iterations and Momentum constant equal 0.0, while Fig. 4b shows the RMS error versus number of iterations (Cycles) and Momentum equal 0.1(Table 3).

Figure 5 shows the difference of the system performance by using IF.

Experiment 2: The three satellite images (data set 2) are normalized to the range from zero to one as shown in Fig. 6 and 8. There are nine and eleven different classes representing the different types of changed and non-changed areas for the two modules respectively.

Module 1: The first module represents changes between two satellite images (1984-2000).

Table 3: Represents optimal parameter used to produce the output

Initial weight set	-1, 1
Number of hidden neuron	7
Number of hidden layer	3
Learning rate	0.2
Momentum	0.1
Improvement factor	0.01

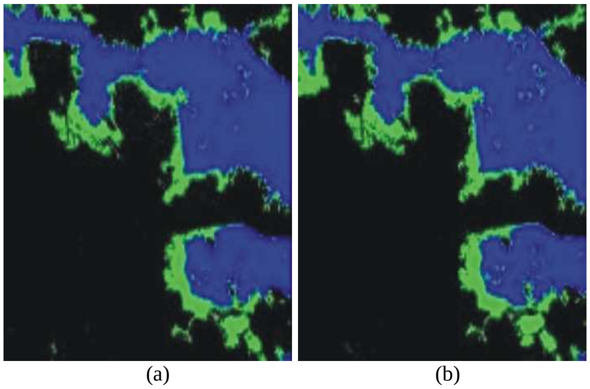

(a) (b)

Fig. 5: (a) Output without IF; (b) output with IF

Figure 7 shows the difference of the system performance by using IF.

No changes from water to water, changed from desert to water, changed from vegetation to water, no

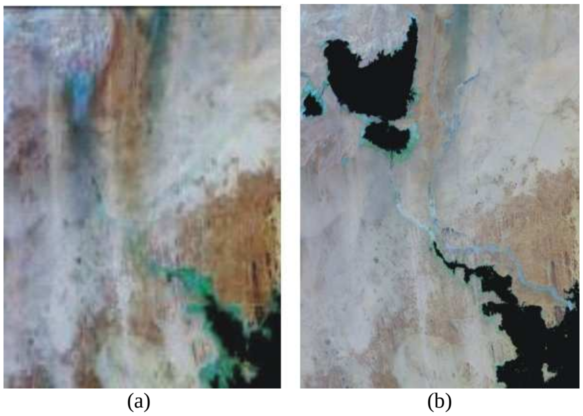

(a) (b)

Fig. 6: (a) First image (1984); (b) second image (2000)

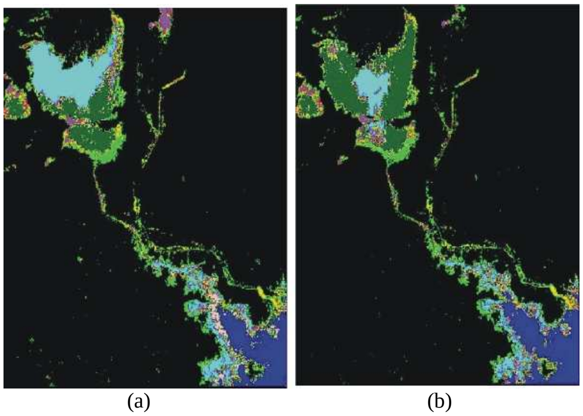

(a) (b)

Fig. 7: (a) Output without IF; (a) output with IF

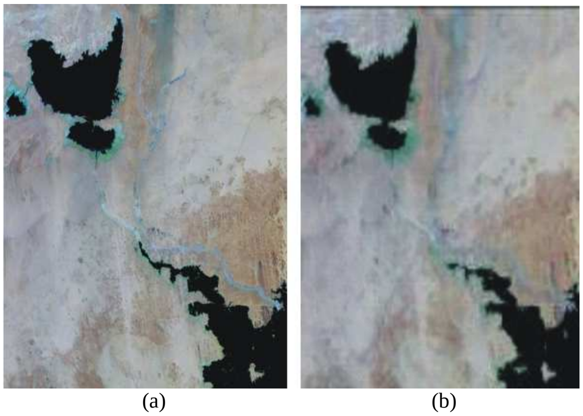

(a) (b)

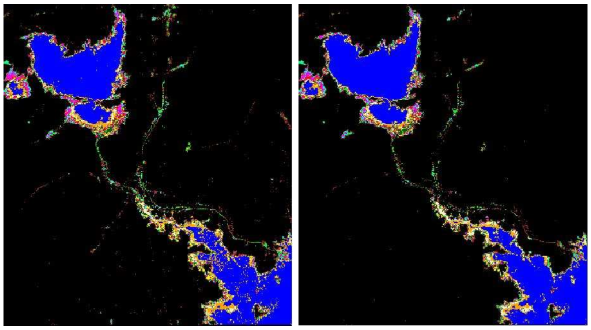
Fig. 8: (a) First image (2000); (b) second image (2003)

Table 4: Represents the change by using IF (data set 2 module: 1)

Ground cover changes	Color	Area (Km^2) with IF	Area (sq. Km.) without IF
Group water to water (No-Change)	Blue	276.2586	241.1127
Class2 desert to water (Change)	D-green	441.2637	266.6763
Class3 vegetation to water (Change)	LightPink	78.1164	112.3470
Class4 desert to desert (No-Change)	Black	7188.4746	7184.3841
Class5 desert to vegetation (Change)	green	179.7309	155.2041
Class6 desert to unknown1 (Change)	Yellow	40.4271	49.7421
Class7 desert to unknown 2 (Change)	Brown	34.5384	45.3681
Class8 sabkha to sabkha (No Change)	purple	58.1256	85.8033
Class9 sabkha to water (Change)	Cyan	290.0691	446.3667

Table 5: Represents the change by using IF (data set 2 –module: 2)

Land cover changes	Color	Area (Km^2) with IF	Area (Km^2) without IF
Class1 water to water (No-Change)	Blue	903.7980	917.8272
Class2 water to vegetation (Change)	Bisque	106.2639	87.3018
Class3 water to sabkha (Change)	Beige	51.0624	69.8544
Class4vegetation to vegetation (No Change)	D-green	69.6114	79.1532
Class5 vegetation to desert (Change)	Yellow	90.2340	46.8909
Class6 sabkha to water (Change)	Red	7.8894	0.8829
Class7 sabkha to vegetation (Change)	Magenta	80.0523	58.4820
Class8 sabkha to sabkha (No-Change)	Cyan	51.6294	51.1839
Class9 desert to vegetation (Change)	green	29.2086	14.8149
Class10 desert to Sabkha (Change)	Brown	9.8982	9.3393
Class11 desert to desert (No-Change)	Black	7187.3560	7251.2730



(a) (b)

Fig. 9: (a) output without IF; (b) output with IF

changes from desert to desert, changed from desert to vegetation, changed from desert to unkown1, changed from desert to unkown2, No changes sabkha to Sabkha and changed from Sabkha to water (Table 4).

Module 2: The second module represents changes between two satellite images (2000-2003).

No changes from water to water, changed from water to vegetation, changed from water to sabkha, no changes from vegetation to vegetation, changed from vegetation to desert, changed from sabkha to water, changed from sabkha to vegetation, no changes sabkha to sabkha, changed from desert to vegetation, changed from desert to sabkha and no changes from desert to desert (Table 5).

Figure 9 shows the difference of the system performance by using IF.

DISCUSSION

From the above it is clear that with the use of this design led to the separation between the changes and put it in a separate layer and specialists to discuss this area in the National Authority for Remote Sensing and Space Sciences and the Ministry of Agriculture acknowledged the credibility of the output of this design .and after comparing the output field trip in 2003, assessed the output of the design by 78%.and after the study recommended that the recommendations of the three:

- The hunting process in a very healthy low Toshka and lakes where the fish breed in this region
- Agriculture in this region is necessary that the spaces away from the lakes enough
- Not be built next to the areas that are leakage of water from which the lakes

CONCLUSION

Using the neural network system for digital change detection shows a great potential compared with other techniques (ratio, classification). Especially, if the two images are not completely radiometricaly balanced and there is no previous information about the value of the threshold, since the other techniques can provide spurious effect on change and non-change classes. Classification of the output can be easily added to the system just by adding additional nodes to the output layer to specify different types of changes. Therefore we can conclude that:

- Adding momentum term in learning phase will reduce the time needed and reduced oscillation of output network
- Using learning rate optimization will reduce the time and effort needed to select an appropriate value of learning rate
- Using improvement factor optimization will reduce superiors data in training phase and improve output results of neural network

There is no universally optimal change detection technique. The choice is dependent upon the application. Even the question of, which is best for any particular application, remains generally open.

ACKNOWLEDGMENT

We would like to thank the National Authority for Remote Sensing and Space science (NARSS) for providing us the data and their staff for the assessment of the results.

REFERENCES

1. Attoh-Okine, N.O., 1999. Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. J. Adv. Eng. Software, 30: 291-302. DOI: 10.1016/S0965-9978(98)00071-4
2. Bao, M., 1999. Backscattering change detection in SAR images using wavelet techniques. Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, June 28-July 2, IEEE Xplore Press, Hamburg, Germany, pp: 1561-1563. DOI: 10.1109/IGARSS.1999.772019
3. Benediktsson, J.A., P.H. Swain and O.K. Ersoy, 1990. Neural network approaches versus statistical methods in classification of multisource remote sensing data. IEEE Trans. Geosci. Remote Sens., 28: 540-552. http://notendur.hi.is/benedikt/neuralvsstat.pdf
4. Cannady, J., 1998. Artificial neural networks for misuse detection. Proceedings of the National Information Systems Security Conference Oct. 1998, Arlington, VA., USA., pp: 443-456.
5. Cavalieri, S. and O. Mirabella, 1999. A novel learning algorithm which improves the partial fault tolerance of multilayer neural networks. Neural Networks, 12: 91-106. DOI: 10.1016/S0893-6080(98)00094-X
6. Haykin, S., 1994. Neural Networks, a Comprehensive Foundation. 1st Edn., Macmillan College Publishing Company, New York, ISBN: 0-02-352761-7, pp: 696.
7. Heerman, P.D. and N. Khazenie, 1992. Classification of multi-spectral remote sensing data using a back-propagation neural network. IEEE Trans. Geosci. Remote Sens., 30: 81-88. DOI: 10.1109/36.124218
8. Hush, D.R. and B.G. Horne, 1993. Progress in supervised neural networks. IEEE Sign. Process. Mag., 10: 8-39. DOI: 10.1109/79.180705
9. Martin, T.H., H.B. Demuth and M.H. Beale, 1995. Neural Network Design. PWS Publishing Company, Boston, Chapter 12, ISBN: 0-9717321-0-8, pp: 9.
10. Riedmiller, M. and H. Braun, 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Proceedings of the IEEE International Conference on Neural Networks, Mar. 28-Apr. 4, IEEE Xplore Press, San Francisco, CA., USA., pp: 586-591. DOI: 10.1109/ICNN.1993.298623
11. Ross, S.L. and C.D. Elividge, 1999. Remote sensing change detection environmental monitoring methods and application. Taylor and Francis, UK., ISBN: 0-415-23795-5, pp: 248.
12. Quinlan, J.R., 1983. Learning Efficient Classification Procedure and Their Application to Chess Endgames. In: Machine Learning, Michalski, R.S., J. Carbonell and T.M. Mitchell (Eds.). Tioga Press, Palo Alto, pp: 463-482.
13. Rao, V.B. and H. Rao, 1995. C++ Neural Network and Fuzzy Logic. 2nd Edn., M and T Press, ISBN: 10: 1558515526, pp: 549.
14. Rumelhart, D., G.R. Hinton and R. Williams, 1986. Learning representation by back-propagating errors. Nature, 323: 533-536. http://www.nature.com/nature/journal/v323/n6088/abs/323533a0.html
15. Singh, A., 1989. Digital change detection techniques using remotely sensed data. IEEE INT. J. Remote Sens., 10: 989-1003. http://www.citeulike.org/user/carueda/article/347201
16. Thaahirah, S., M. Rasied, Othman O. Khalifa and Y.B. Kamarudin, 2005. Human face recognition based on singular valued decomposition and neural network. Proceeding of the GVIP 05 Conference, Dec. 19-21, CICC, Cairo, Egypt, pp: 1-6. http://www.icgst.com/GVIP05/papers/P1150540101.pdf

17. Werntges, H., 1993. Partitions of unity improve neural network function approximation. Proceeding of the IEEE International Conference on Neural Networks, Mar. 28-Apr. 4, IEEE Xplore Press, San Francisco, CA., USA., pp: 914-918. DOI: 10.1109/ICNN.1993.298679
18. Yu, X.H., 1992. Can back propagation error surface not have local minima. IEEE Trans. Neural Networks, 3: 1019-1021. DOI: 10.1109/72.165604