International Journal of Computer Applications (0975 – 8887)
Volume 134 – No.11, January 2016

Neural Network Approach for Automatic Landuse Classification of Satellite Images: One-Against-Rest and Multi-Class Classifiers

Anil Kumar Goswami
DTRL, DRDO
Delhi, India

Heena Joshi
Banasthali Vidhyapith
Rajasthan, India

S. P. Mishra
DTRL, DRDO
Delhi, India

ABSTRACT
Artificial Neural Network (ANN) is an important Artificial Intelligence (AI) and Machine Learning (ML) method used in various remote sensing applications such as image classification, pattern recognition etc.One of important remote sensing applications is the landuse classification i.e. classification of satellite data into various landuse classes such as forest, waterbody, snowcover etc. Landuse classification from satellite data can take place in manual, semi-automatic or automatic mode. Automatic landuse classification is necessary to reduce manual efforts,which can be achieved by making use of machine learning techniques. This paper uses neural network approach for automatic landuse classification from satellite data by providing two classification approaches using multi layer perceptron (MLP) namely one against rest classification (OARC) and multi class classification (MCC), and then provides the comparison between these two approaches.

Keywords
Artificial Neural Network (ANN), Multi Layer Perceptron (MLP), Error Back Propagation (EBP), Landuse Classification, One-Against-Rest Classification (ORAC), Multi-Class Classification (MCC), Landuse Classification, Remote Sensing

1. INTRODUCTION
Classification of satellite image into landuse classes is an important component for various remote sensing applications and decision support systems. Satellite images are very useful resource for extraction of various landuse classes in manual, semi-automatic or automatic mode. Automatic classification of satellite images is an area of classification that exploits the capability and computational power of the machine and makes use of artificial intelligence approach to emulate the human visual interpretation process. There are various machine learning (ML) methods available for this purpose. Artificial Neural Network (ANN) is one of the ML methods based on the working of human brain, and makes the machine to learn to perform the classification of satellite images in an automatic mode. There are various types and variation of ANN classification algorithms such as Multi Layer Perceptron (MLP), Self Organizing Feature Map (SOFM), Radial Basis Functions (RBF), Probabilistic Neural Networks (PNN), Cellular Neural Networks (CNN) etc. MLP is an ANN used very frequently for various applications such as pattern classification, function approximation or prediction and it is universal in the sense that they can approximate any continuous nonlinear function [1].

This paper provides a framework for an automatic classification of landuse using MLP classifier with two different approaches namely one against rest classification (OARC) and multi class classification (MCC) and then compares these two approaches. The next section of the paper presents the theoretical framework with respect to MLP. This section is followed by methodology that explains in detail the designing and developing of the system along with implementation of experimental setup. Result and discussion section discusses the outcome of the study. The final section presents the conclusion of the paper.

2. THEORETICAL FRAMEWORK
2.1 Multi Layer Perceptron (MLP) and Error Back Propagation (EBP)
ANN is based on the biological structure of the neurons and their connections in living organisms [2]. It is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge in inter-neuron connection strengths known as synaptic weights by a learning process and making it available for use for prediction [1]. Neurons are arranged in various layers that include an input layer, hidden layers, and an output layer [3]. MLP invented by Rosenblatt in 1958, is a type of ANN that consists of one input layer, one output layer and one or more hidden layers as shown in Figure 1. The role of hidden layers is to compute a weighted sum of their activation inputs and pass the result to neurons in succeeding layers [3].

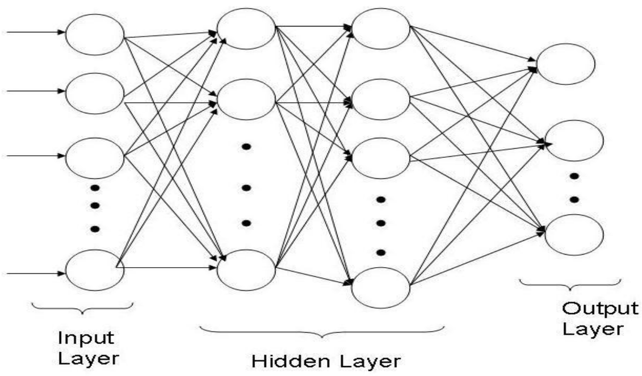

Figure 1: MLP Layers

MLP is trained using training data before it is applied for prediction [3] and this training is supervised learning where prior information of desired response is known [1]. There are various algorithms proposed for the training of MLP but EBP algorithm is being used widely [4]. It is based on gradient descent method for teacher-based supervised learning and developed in the 1960s and 1970s, and then applied to neural networks in 1981 [5]. In EBP algorithm, weights of various connections between nodes are updated by feeding back the differences between network output and desired output by making use of gradient decent method [4]. EBP algorithm consists of two processes namely feed forward propagation and error backward propagation. Error backward propagation consists of two key processes namely computing gradient and updating of weights. Feed forward propagation computes network responses from the input layer to the output layer to provide computed output whereas error back propagation transmits the error (between computed output and desired output) from the output layer to the input layer and then modifying the connection weights between the neurons thus adjusting network for getting the desired output for any input [6]. Feed forward process of MLP has been presented below.

Start FEEDFORWARD
For *layer* = 1 to L do
For node = 1 toN_{layer} do

$$u_{layer,node} = f(\sum_{i=0}^{N_{layer-1}} w_{layer,node,i} * u_{layer-1,i})$$

End {FEED_FORWARD}

Here, L represents the number of hidden layers plus output layer, N represents the nodes in a layer, u represents the output matrix and fis an activation function. The process for computing of gradient of MLP has been presented below.

Start GRADIENT COMPUTATION
For *layer* = L-1 to 1 do
For node = 1 to $N_{layer+1}$ do
If *layer* = *L* - 1 then

$$e_{node-1} = d_{node-1} - u_{L,node}$$

$$G_{layer-1,node-1} = a * u_{layer+1,node} * (1 - u_{layer+1,node}) * e_{node-1}$$

Else

$$G_{layer,node-1} = a * u_{layer+1,node} * (1 - u_{layer+1,node}) * \sum_{k=1}^{N_{layer+2}} (G_{layer+1,k-1} * w_{layer+1,node,k})$$

End {GRADIENT COMPUTATION}

Here, e represents the error component and G represents the gradient matrix. The standard BP algorithm is based on the Widrow–Hoff delta learning rule and makes use gradient descent method for correcting and adjusting each connection weight along the opposite direction of the gradient of error performance function [6]. The weight updation mechanism has been presented below.

Start WEIGHT UPDATION
For *layer* = 0 to L do
For node = 1 to $N_{layer+1}$ do
For count = 0 to N_{layer} do

$$w_{layer,count,node} = w_{layer,count,node} + \eta * u_{layer,count} * G_{layer,node-1}$$

End {WEIGHT UPDATION}

Here, W represents the weight matrix.

2.2 Satellite Images

Satellite images contain data of area of interest captured by various remote sensing satellites. The satellite captures the data of an area of interest on earth in form of reflectance values that are then converted into pixel values based on certain well established computation. A satellite image is a collection of pixels arranged in matrix form i.e. rows and column, and each pixel is represented by a vector of the size of number of bands in that image. In case of multispectral data, there are few bands corresponding to each pixel whereas in hyper-spectral data there are many more bands. Satellite images are significant sources of information with respect to various natural and man-made objects and used for various remote sensingapplications for civilian and defence sector.

2.3 Landuse

Landuse or landcover refers to the use of the land either naturally or by the human being. There are various examples of natural landuse such as forest, waterbody, snowcover, vegetations etc. and man-made landuse such as urban area, ploughing land, man-made waterbody etc.

3. METHODOLOGY

This section provides the detailed methodology adopted in this research work and explains the experiment conducted. Two approaches have been followed for landuse classification in automatic mode namely one-against-rest classification and multi-class classification using MLP.

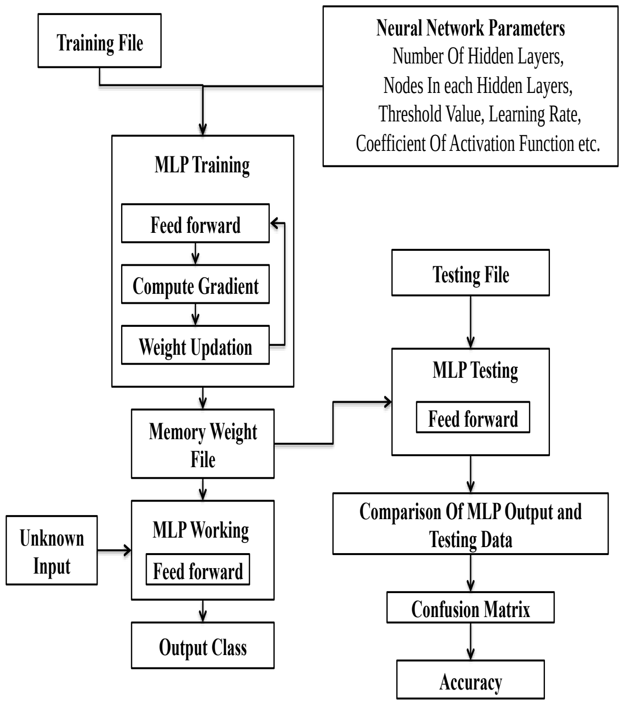

Figure 2: MLP Block Diagram [1]

The block diagram of the MLP classifier used for this work has been shown in Figure 2. The MLP consists of three modules namely MLP training, MLP testing and MLP working. The first module MLP training provides the training to the network using EBP learning algorithm and thus creating memory or weight matrix with respect to training data. MLP testing module is used to test the accuracy of the network using test data. The trained network is then used for solving the problem for which it has been designed, trained and tested and this is done by MLP working module. The flow chart for the training process using MLP is shown in the Figure 3.

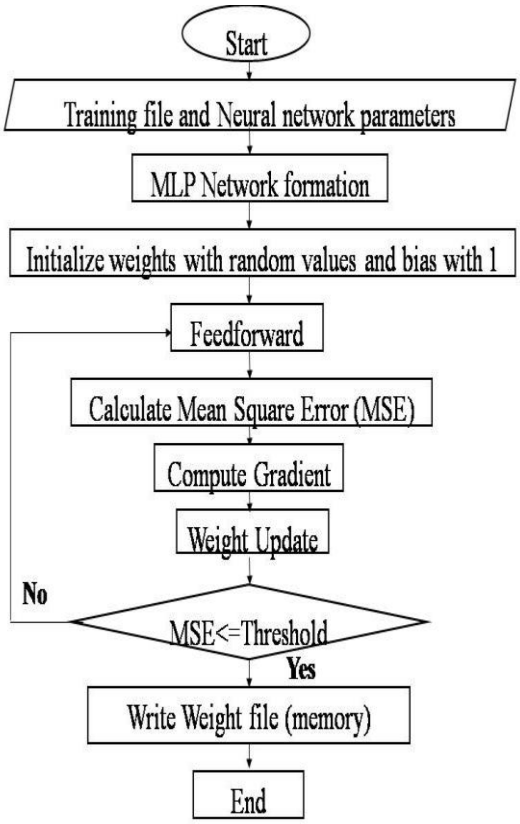

Figure 3: Flow Chart of MLP training process using EBP [1]

3.1 Approach 1: One-Against-Rest Classification (OARC)

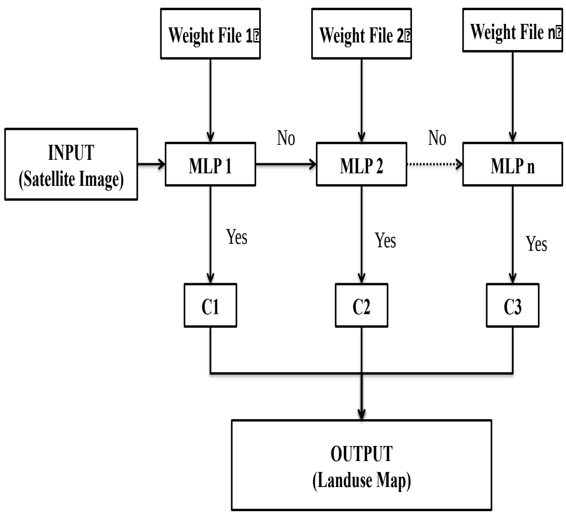

Figure 4: OARC Block Diagram

In ORAC, there is one classifier for each class, which means that there are N classifiers for an N class classification problem. It creates N binary classifiers and combines their results to determine the class label of a pattern. This method decomposes an N-class problem into a series of N two-class problems where each problem discriminates a given class from the other N−1 classes and each classifier is trained to distinguish one class from the remaining N-1 classes. The training dataset of N class problem is decomposed into a series of N training files. Each classifier is trained individually with the training file of the class for which it has been designed. After the completion of N classifiers, N weight files are created. All the classifiers are then cascaded to get the output of an unknown dataset. The block diagram of OARC is shown in Figure 4. Figure 5 shows the flow chart of testing process in which testing file and weight file are given as input.

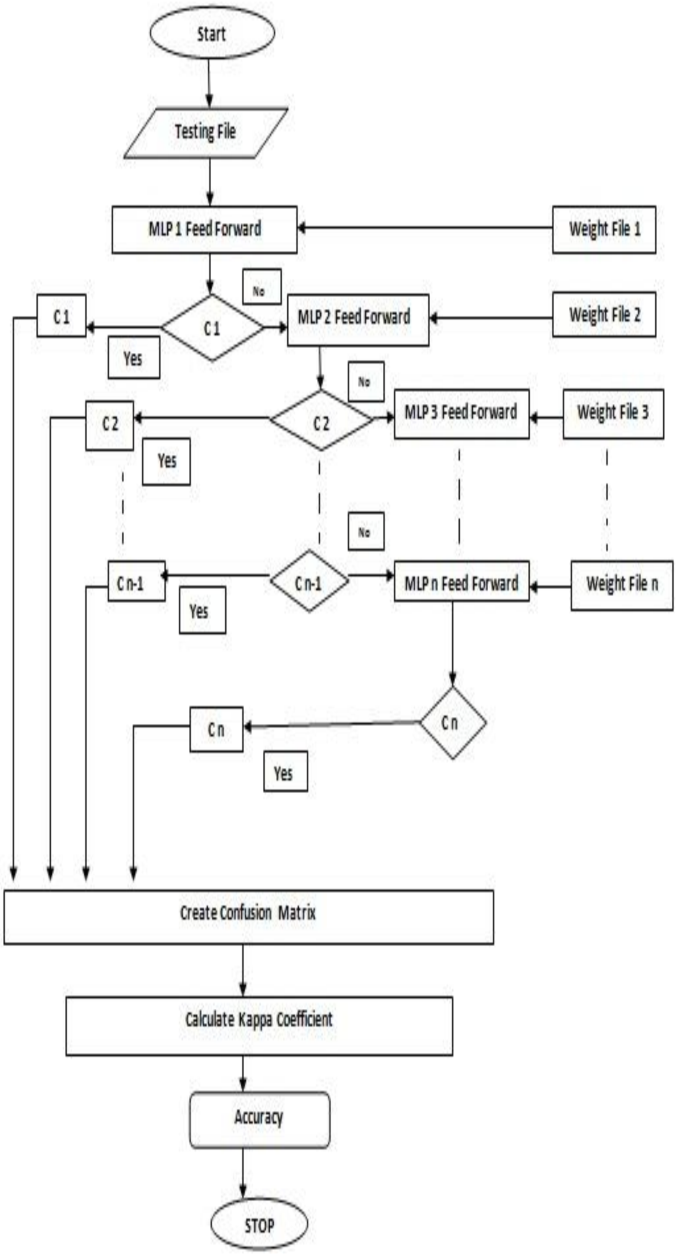

Figure 5: Flow Chart of OARC testing process

3.2 Approach 2: Multi-Class Classification (MCC)

In MCC, there is only one classifier for an N class classification. There is only one training file consisting of patterns of all N classes and the classifier is trained with this training file and only a single weight file is created. The N class classification problem is not broken down in various binary classifiers as in the case of OARC but it is considered as a whole. The block diagram of MCC has been shown in the

Figure 6. The flow chart for the testing process using MLP is shown in the Figure 7.

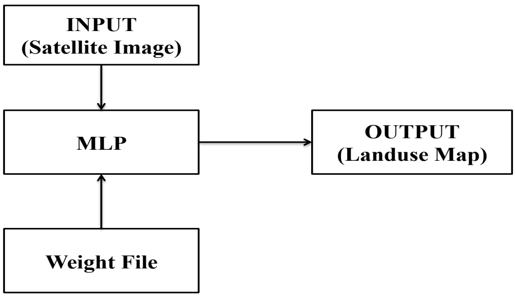

Figure 6: MCC Block Diagram

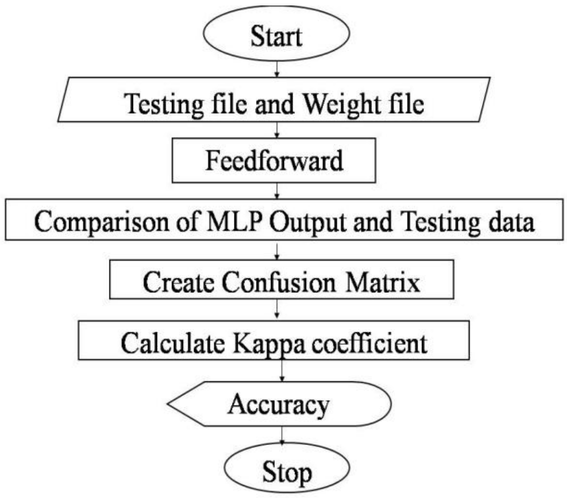

Figure 7: Flow Chart of MCC testing process

3.3 Training and Testing Dataset

Training and testing data have been created with the help of domain expert for using in this research work. Training data consist of patterns used to train the network whereas testing data consist of patterns used to test the network that whether it has achieved the desired result or not. Training and testing data have been created from satellite images by means of taking samples of various landuse classes existing in these images. For the robustness of the training and testing data, the samples have been drawn from different images. Each pattern in the training and testing data consists three dimensional feature vector having Red, Green and Blue values of pixels along with class label of landuse class to which it belongs. The number of landuse classes taken in this research work is seven. The ratio of number of patterns in training and testing data is 30:70 respectively. Three such sets have been created and used for training and testing purpose. Table 1 shows total number of patterns of training and testing data in three data sets. Three datasets as shown in Table 1 have been used for training and testing of MLP in MCC approach. For OARC, the data have been further partitioned into various training and testing data as shown in Table 2, Table 3 and Table 4.

Table 1: Training and Testing data

Set	1	2	3
Total Training Patterns	2532	2527	2531
Total Testing Patterns	5918	5909	5917
Total Number of classes	7	7	7

Table 2: Training and Testing data for OARC (Set 1)

	SET 1					
	Training File			**Testing File**		
OARC	**Class**	**Others**	**Total**	**Class**	**Others**	**Total**
Class 1	552	1980	**2532**	1288	4630	**5918**
Class 2	600	1932	**2532**	1402	4516	**5918**
Class 3	512	2020	**2532**	1196	4722	**5918**
Class 4	204	2328	**2532**	478	5440	**5918**
Class 5	155	2377	**2532**	364	5554	**5918**
Class 6	259	2273	**2532**	606	5312	**5918**
Class 7	250	2282	**2532**	584	5334	**5918**

Table 3: Training and Testing data for OARC (Set2)

	SET 2					
	Training File			**Testing File**		
OARC	**Class**	**Others**	**Total**	**Class**	**Others**	**Total**
Class 1	546	1981	**2527**	1276	4633	**5909**
Class 2	606	1921	**2527**	1414	4495	**5909**
Class 3	507	2020	**2527**	1186	4723	**5909**
Class 4	204	2323	**2527**	478	5431	**5909**
Class 5	154	2373	**2527**	361	5548	**5909**
Class 6	265	2262	**2527**	621	5288	**5909**
Class 7	546	1981	**2527**	1276	4633	**5909**

Table 4: Training and Testing data for OARC (Set3)

	SET 2					
	Training File			**Testing File**		
OARC	**Class**	**Others**	**Total**	**Class**	**Others**	**Total**
Class 1	552	1979	**2531**	1290	4627	**5917**
Class 2	605	1926	**2531**	1413	4504	**5917**
Class 3	511	2020	**2531**	1194	4723	**5917**
Class 4	201	2330	**2531**	469	5448	**5917**
Class 5	154	2377	**2531**	361	5556	**5917**
Class 6	259	2272	**2531**	607	5310	**5917**
Class 7	249	2282	**2531**	583	5334	**5917**

4. RESULTS AND DISCUSSION

Table 5: Various parameters for OARC training and testing corresponding to each classifier (Set 1)

	SET 1						
Classifiers	**C 1**	**C 2**	**C 3**	**C 4**	**C 5**	**C 6**	**C 7**
MLP Architectu	1(8)	1(8)	1(8)	1(8)	1(8)	1(8)	1(8)
Learning Rate (η)	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Number of Classes	2	2	2	2	2	2	2
Activation Rate	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Training MSE	1.85E-6	1.14E-5	4.67E-6	1.26E-10	2.04E-7	1.04E-7	5.10E-8
Training Accuracy	**97.11%**	**95.85%**	**98.22%**	**100%**	**99.92%**	**100%**	**100%**
Testing Accuracy	**95.67%**	**95.13%**	**98.07%**	**100%**	**99.62%**	**99.81%**	**99.79%**

Generally, the final network is found through a trial-and-error procedure [6][7] and depends on user experience as well as needs intensive human interaction and computational time [7]. The same approach has been applied in this research work to find the number of hidden layers and number of nodes in each of them. No ubiquitous principle for the relationship has been found between the number of hidden neurons and the accuracy of a model, however, in the case of same accuracy by many models, the model with less number of hidden neurons is considered because of having less computation time [4] and less prone to overfitting.

Table 6: Various parameters for OARC training and testing corresponding to each classifier (Set2)

	SET 2						
Classifiers	**C 1**	**C 2**	**C 3**	**C 4**	**C 5**	**C 6**	**C 7**
MLP Architectu	1(5)	1(5)	1(5)	1(5)	1(5)	1(5)	1(5)
Learning Rate (η)	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Number of Classes	2	2	2	2	2	2	2
Activation Rate	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Training MSE	1.83 E-6	1.46 E-5	6.93 E-6	9.10 E-11	9.73 E-7	1.18 E-7	8.01 E-8
Training Accuracy	**96.56%**	**97.55%**	**98.61%**	**100%**	**99.72%**	**99.96%**	**100%**
Testing Accuracy	**97.14%**	**97.63%**	**98.96%**	**100%**	**99.00%**	**99.08%**	**99.88%**

Table 7: Various parameters for OARC training and testing corresponding to each classifier (Set 3)

	SET 3						
Classifiers	**C 1**	**C 2**	**C 3**	**C 4**	**C 5**	**C 6**	**C 7**
MLP Architectu	1(8)	1(8)	1(8)	1(8)	1(8)	1(8)	1(8)
Learning Rate (η)	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Number of Classes	2	2	2	2	2	2	2
Activation Rate	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Training MSE	2.69 E-6	4.39 E-6	3.84 E-6	1.05 E-10	4.69 E-7	1.09 E-7	6.53 E-8
Training Accuracy	**96.79%**	**96.99%**	**98.77%**	**100%**	**99.84%**	**100%**	**100%**
Testing Accuracy	**97.24%**	**97.61%**	**98.42%**	**100%**	**99.59%**	**99.27%**	**99.69%**

Various combinations of hidden layers and nodes in them have been experimented for MLP architecture for this research work. The network architectures along with network parameters suited best for OARC have been shown in the Table 5, Table 6 and Table 7 for set 1, set 2 and set 3 respectively. The architecture giving good result in OARC consists of one hidden layer with either 5 nodes or 8 nodes. The training and testing accuracy is more than 98% consistently across all the sets and all the classifiers except few exceptions. The Mean Squared Error (MSE) for all the classifiers is having reasonable low value showing the fitness of network architecture. The consistency of having very high testing accuracy shows that network is not over-fitted with the training data.

Table 8: Various parameters for OARC overall testing after cascading the classifiers

SET	1	2	3
Number of Classes	7	7	7
Total Training Patterns	2532	2531	2527
Total Testing Patterns	5918	5917	5909
Testing Accuracy	**93.05%**	**94.22%**	**93.51%**

Table 8 has shown the overall accuracy of testing of OARC after cascading the classifiers as shown in Figure 5. The results are encouraging with overall accuracy of 93.05%, 94.32% and 94.55% for set 1, set 2 and set 3 respectively. This is reasonable good accuracy for a classifier.

Table 9: Various parameters for MCC training and testing

SET	1	2	3
MLP Architecture	2(50,50)	2(50,50)	2(50,50)
Learning Rate (η)	0.2	0.2	0.2
Activation Rate	0.5	0.5	0.5
Number of Classes	7	7	7
Training MSE	1.86E-6	5.28E-6	5.37E-6
Total Training Patterns	2532	2531	2527
Training Accuracy	**99.64%**	**98.73%**	**97.82%**
Total Testing Patterns	5918	5917	5909
Testing Accuracy	**96.63%**	**95.06%**	**95.51%**

The network architectures along with network parameters suited best for MCC has been shown in the Table 9 for all sets. The architecture giving good result with MCC consists of two hidden layers with 50 nodes in each hidden layer. The training accuracy is good having values 99.64%, 98.73% and 97.82% for set 1, set 2 and set 3 respectively showing the consistency for all the sets. The testing accuracy is also good having values 99.63%, 95.06% and 95.51% for set 1, set 2 and set 3 respectively and it is also showing the consistency for all the sets. The Mean Squared Error (MSE) for all sets of MCC is having reasonably low value showing the fitness of network architecture. The consistency of having very high testing accuracy shows that network is not over-fitted with the training data in MCC.Confusion matrix and kappa coefficient of all sets of OARC and MCC have been shown in Table 10-15.

The results of the experiment in terms of training and testing accuracy, and kappa coefficient with two approaches namely OARC and MCC signify the importance of MLP as a good classifier for automatic landuse classification from satellite data with significantly good accuracy. The comparison between OARC and MCC approaches has shown that the results are better in MCC than OARC. However the size of the network is significantly less in OARC, which results into less computation time in OARC against MCC.

Satellite image is given as input to MLP, which then classify it into landuse map based on learning provided by training data during training phase of the network. The landuse maps generated from automatic landuse classification using MLP, have been shown in Figure 9, 10 and 11 using both the approaches namely OARC and MCC. Different color codes in the output symbolize the different landuse classes.

Table 10: Confusion Matrix for OARC (Set 1)

	C1	C2	C3	C4	C5	C6	C7
C1	1033	188	0	0	0	0	67
C2	1	1366	11	0	0	0	24
C3	0	0	1099	0	4	0	93
C4	0	0	0	478	0	0	0
C5	0	0	0	0	357	2	5
C6	0	0	0	0	8	596	2
C7	0	0	6	0	0	0	578
	KAPPA:-0.9161						

Table 11: Confusion Matrix for OARC (Set 2)

	C1	C2	C3	C4	C5	C6	C7
C1	1121	52	0	0	0	0	117
C2	0	1326	15	0	0	0	72
C3	0	0	1152	0	8	0	34
C4	0	0	0	469	0	0	0
C5	0	0	0	0	326	32	3
C6	0	0	0	0	7	600	0
C7	0	0	2	0	0	0	581
	KAPPA:-0.9302						

Table 12: Confusion Matrix for OARC (Set3)

	C1	C2	C3	C4	C5	C6	C7
C1	1113	12	0	0	0	0	151
C2	0	1285	15	0	0	0	114
C3	0	0	1125	0	10	0	51
C4	0	0	0	478	0	0	0
C5	0	0	1	0	348	4	8
C6	0	0	0	0	0	621	0
C7	0	0	16	0	1	0	556
	KAPPA:-0.9220						

Table 13: Confusion Matrix for MCC (Set 1)

	C1	C2	C3	C4	C5	C6	C7
C1	1214	74	0	0	0	0	0
C2	31	1356	15	0	0	0	0
C3	0	38	1155	0	3	0	0
C4	0	0	0	478	0	0	0
C5	0	0	2	0	358	4	0
C6	0	0	0	0	20	586	0
C7	0	0	12	0	0	0	572
	KAPPA:-0.9592						

Table 14: Confusion Matrix for MCC (Set 2)

	C1	C2	C3	C4	C5	C6	C7
C1	1242	48	0	0	0	0	0
C2	86	1292	34	0	1	0	0
C3	2	1	1165	0	20	0	6
C4	0	55	0	414	0	0	0
C5	0	0	0	0	335	26	0
C6	0	0	0	0	7	600	0
C7	0	0	6	0	0	0	577
	KAPPA:-0.9400						

Table 15: Confusion Matrix for MCC (Set 3)

	C1	C2	C3	C4	C5	C6	C7
C1	1167	109	0	0	0	0	0
C2	24	1363	24	0	3	0	0
C3	0	8	1140	0	38	0	0
C4	0	0	0	478	0	0	0
C5	1	5	8	0	313	34	0
C6	0	0	0	0	2	619	0
C7	0	0	9	0	0	0	564
	KAPPA:-0.9456						

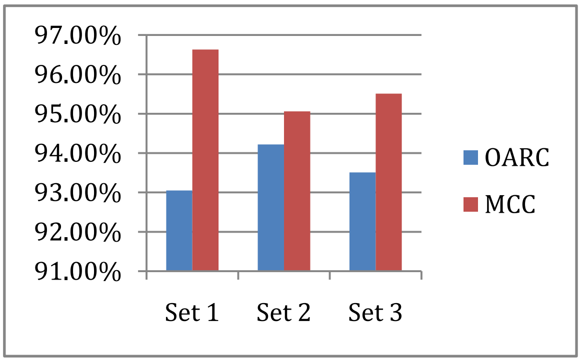

Figure 8: Comparison of Testing Accuracy of OARC and MCC

5. CONCLUSION

ANN is an important method for automatic landuse classification from satellite data by using EBP as a learning algorithm. Accuracy of results of training and testing data in this experiment has shown the strength of the MLP as a good classifier for image classification and various other remote sensing applications.The experiment has made use of two different approaches for automatic generation of landuse map from satellite data. The result has shown the better accuracy in case of MCC as compared to OARC, however OARC classifiers are having less number of hidden layers and hidden nodes compared to MCC. Hence, it signifies the trade-off between OARC and MCC with respect to computational cost and accuracy. Mostly, MCC should be used because it has significantly more accuracy than OARC. However, in certain applications such as real time applications, where there is need of fast computation and if the accuracy given by OARC is acceptable, the OARC may be used in place of MCC. It means that choice of a method depends on the problem in hand. One more thing to be kept in mind that accuracy of a method heavily depends on the robustness of the training and testing data. The consistency in training and testing accuracy in this experiment shows the robustness of training and testing data used here.

Overall the use of ANN for automatic extraction of landuse map from satellite image is very promising and should be incorporated in related applications. For making use of ANN in such applications, training and testing data should be created carefully to make them robust, reliable and consistent. Further, designing of architecture of ANN in terms of various parameters for a problem is also a challenging area which need to be explored further as till now no suitable methods are available to find the free parameters of the network which works well with all the possible dataset.

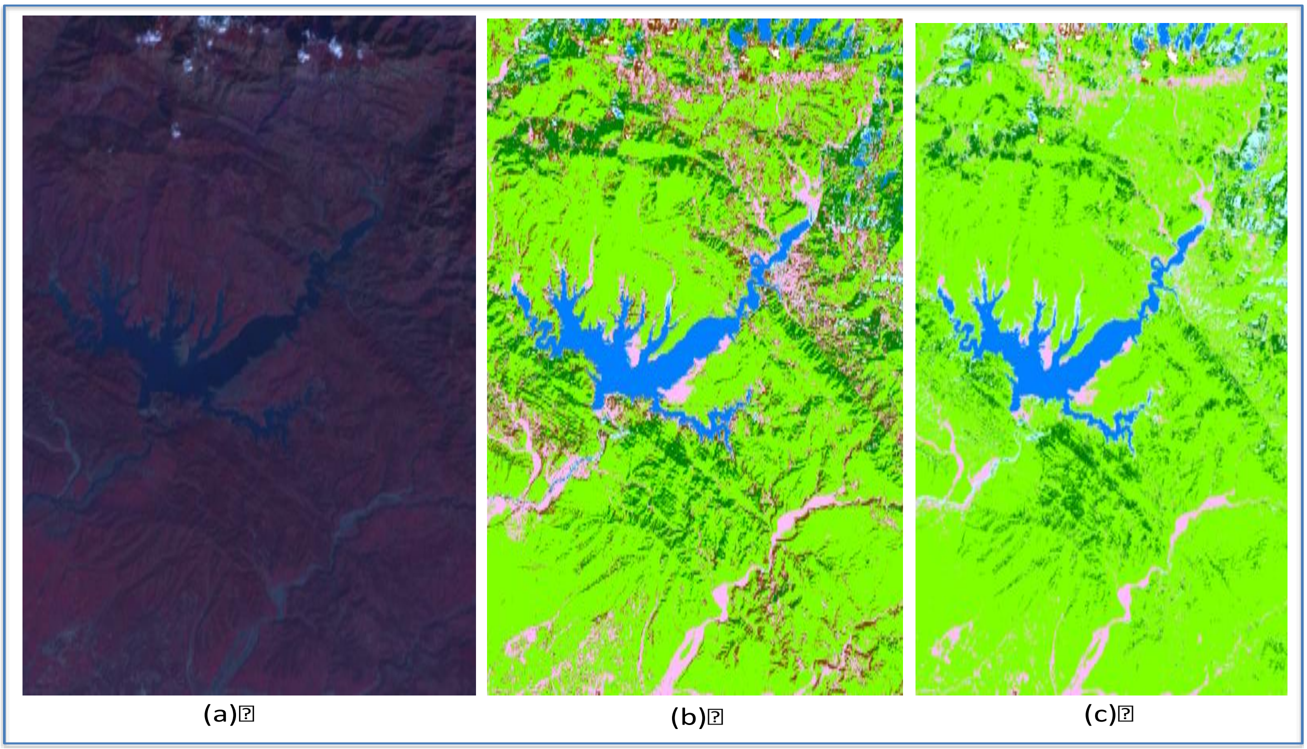

Figure 9: Landuse Map (a) Satellite image as input (b) OARC(c) MCC

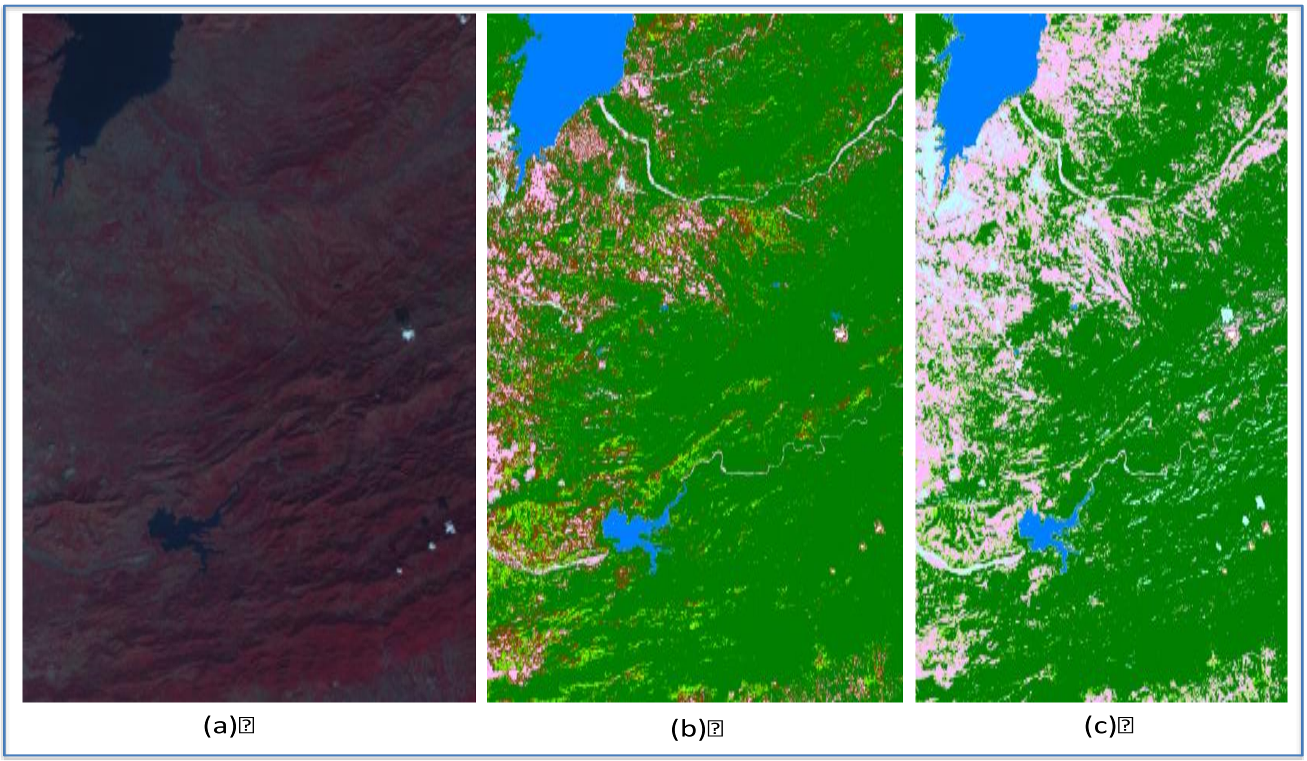

Figure 10: Landuse Map (a) Satellite image as input (b) OARC (c) MCC

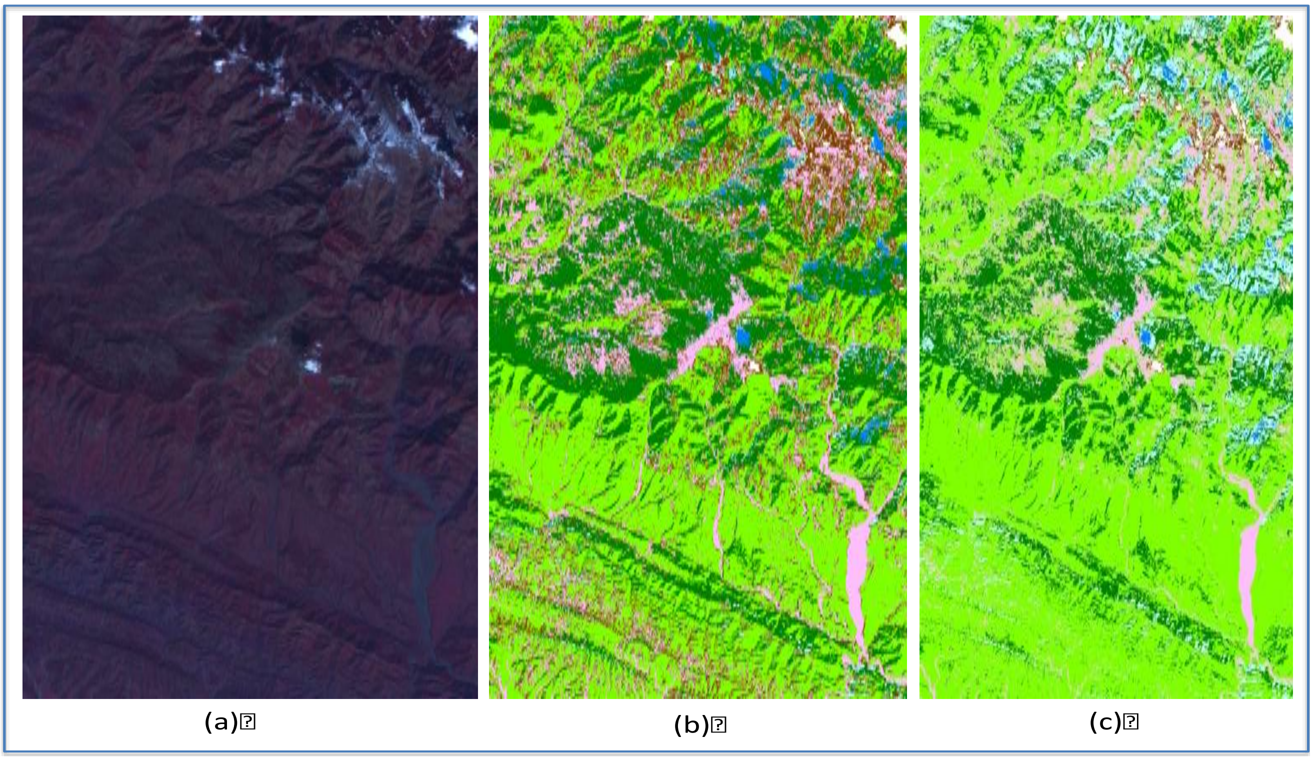

Figure 11: Landuse Map (a) Satellite image as input (b) OARC (c) MCC

6. REFERENCES

[1] Goswami, A.K., Gakhar, S. and Kaur, H., 2014. Automatic object recognition from satellite images using artificial neural network. International Journal of Computer Applications, 95(10), pp. 33-39.

[2] Bangalore, P. and Tjernberg, L.B., 2015. An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings. Smart Grid, IEEE Transactions on, 6(2), pp.980-987.

[3] Agarwal, D., Tamir, D.E., Last, M. and Kandel, A., 2012. A comparative study of artificial neural networks and info-fuzzy networks as automated oracles in software testing. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 42(5), pp.1183-1193.

[4] Zhao, Z., Xu, S., Kang, B.H., Kabir, M.M.J., Liu, Y. and Wasinger, R., 2015. Investigation and improvement of multi-layer perception neural networks for credit scoring. Expert Systems with Applications, 42(7), pp.3508-3516.

[5] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks, 61, pp.85-117.

[6] Hou, J. and Huang, C., 2014. Improving mountainous snow cover fraction mapping via artificial neural networks combined with MODIS and ancillary topographic data. Geoscience and Remote Sensing, IEEE Transactions on, 52(9), pp.5601-5611.

[7] Lu, T.C., Yu, G.R. and Juang, J.C., 2013. Quantum-based algorithm for optimizing artificial neural networks. Neural Networks and Learning Systems, IEEE Transactions on, 24(8), pp.1266-1278.