

# The Composition of the Output Layer Architecture in a Back-Propagation (BP) Neural Network for Remote Sensing Image Classification

Guobin Zhu<sup>†</sup> and Dan G. Blumberg<sup>‡</sup>

Department of Geography and Environmental Development, Ben-Gurion University of the Negev, P.O.B 653, Beer-Sheva, 84105, Israel

<sup>†</sup>E-Mail: [zhu@bgumail.bgu.ac.il](mailto:zhu@bgumail.bgu.ac.il)

<sup>‡</sup>E-Mail: [blumberg@bgumail.bgu.ac.il](mailto:blumberg@bgumail.bgu.ac.il)

**KEY WORDS:** Neural Network, Remote Sensing, Classification, Output Layer Architecture

## ABSTRACT

A Neural Network is treated as a data transformer when used for mapping purposes. The objective in this case, is to associate the elements in one set of data with the elements in a second set. According to this principle, three encoding methods, namely, single output layer, binary encoding, and ortho-encoding, have been designed for the output layer of a Neural Network based on five criteria, and put into experiments for Remote Sensing classification by means of a series of images coordinated with incremental noise level, from 1% to 10%. At last, the experiment results are assessed from different perspectives such as, accuracy, convergence, mixture detection, and confidence of classification, comparing to three encoding methods respectively.

## 1. Introduction

Over the past decades there has been considerable increase in the use of Neural Networks for image classification in Remote Sensing. Most studies of Neural Networks in this area can be subdivided into various aspects, including the structure of the Networks (Lippmann, 1987; Caudill, 1988; Pao, 1989; Widrow and Lehr, 1990; Paola and Schowengerdt, 1995; Fischer and Staufer, 1999), application of Networks (Miller, Kaminsky and Rana, 1994; Yang, Meer and Bakker, 1999; Atkinson and Tatnall, 1997), and improvements to Neural Networks (Chen, Yen and Tsay, 1997; Kaminsky, Barad and Brown, 1997; Mather, Tso and Koch, 1998; Kavzoglu and Mather, 1999).

There are many different types of neural Networks (Pao, 1989). Rather than describe each type, this paper focuses on one of the most commonly used Neural Networks in Remote Sensing, Back-Propagation network (BP).

The discussion about the output layout encoding is described by Benediktsson *et al.* (1990), Heermann and Khazenie (1992), and Civco (1993). Usually, they used different encoding methods in their classification work, and derived some conclusions. With great difference from their work, in which the real classification performance has been done using some particular encoding methodologies, individually, our work focuses on the effectiveness of neural networks based on these encoding methods, using a series of test images mixed with noise. Comparing different encoding systems, this work can provide technical guides to the real classification in Remote Sensing in various aspects, such as accuracy, confidence, mixture detection, and training convergence.

## 2. The encoding methods of the output layer in BP networks

### 2.1 The mapping perspective on classification understanding

Neural networks, in the simplest sense, may be seen as data transformers (Pao 1989), where the objective is to associate the elements in one set of data with the elements in a second set. When applied to classification, for example, they are concerned with the transformation of data from feature space to class space (Atkinson & Tatnall, 1997). As for a typical multi-layer perceptron architecture, each node in it can be viewed as a system which combines inputs in a “quasi-linear”

way and in so doing defines a hyper-surface in a feature space which, when combined with a decision rule or process can be used to separate hyper-regions and, thus, classes (Kanellopoulos & Wilkinson, 1997). Therefore, expressing the class space and forming the highest transformation efficiency with precision between feature space and class space will be a worthy issue for discussing.

Since the input space is unchangeable (in spite of the situation one uses different data sources or auxiliary data sets), the ways to improve the effectiveness of classification will be a) design of Neural Network architectures, including hidden layer and an output layer composition, and, b) use of different Neural Networks. Here we choose output layer composition (or encoding) design for the former solution.

## 2.2 The geometric understanding of the composition of output layer

Based on geometric viewpoints, the best way to design the output layer is to scatter the class nodes (which are treated as M-dimensional distribution of one class, M means output dimensions) as much as possible on a M-dimensional hyperspace. To design such an output layer, there are two ways to go: a) a way based on single dimension, b) a way based on multiple dimensions. Fig 1,2 and 3 show the basic structure from one dimension to three dimensions.

To design an output layer for a neural classifier, at least one of the following criteria should be considered:

- **Speed of training.** The encoding system is designed to use the shortest time performing a succeed network training.
- **The maximum scatter degree for output vectors.** There are several factors to assess the maximum scatter degree. One is based on Euclidean distance; another is based on Hamming distance.
- **The length of the codeword.** The length of the codeword will affect the dimension of class space, and affect the effectiveness of training procedure at last. Therefore, the short length of codeword is always welcome.
- **The characteristics of ortho-intersection.** The ortho-intersection characters existing among output vectors from one to each other will make the output space redundancy minimized, thus facilitating the training process.
- **The ability to detect mixture features.** The best result of classification is extracting relevant features from remote sensing images with near-zero error, including omission error and commission error. In the real situation, however, there always exists mis-classification. Therefore, the capability to detect mixture features becomes one factor to design the output layer composition.

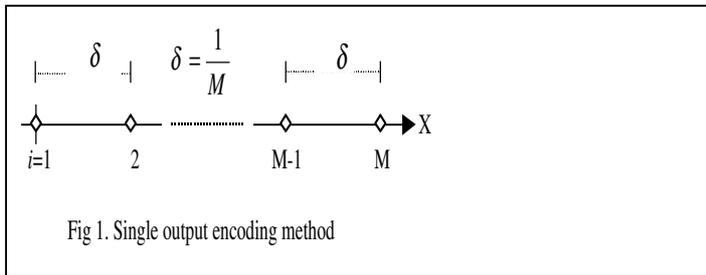
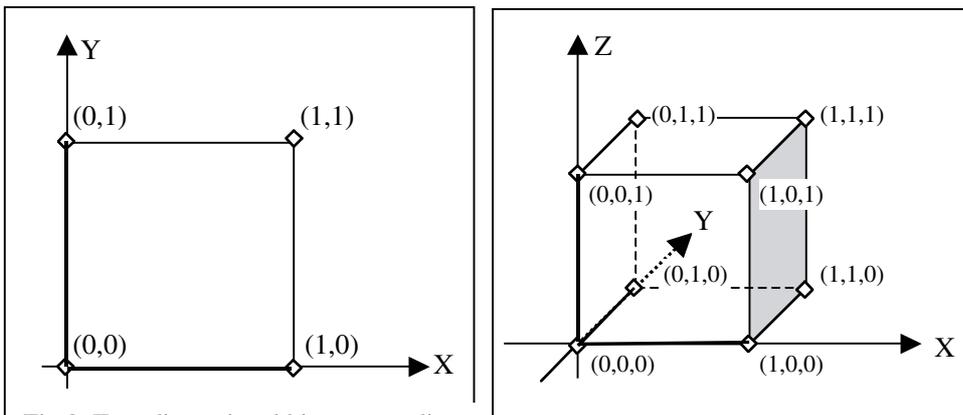


Fig 1. Single output encoding method

It is worthy to note that it is almost impossible to design an encoding system to meet all of the requirements from the criteria mentioned above. For example, speed and codeword length are a pair of paradox factors, so are speed and ortho-intersection character. Consequently, there should be some compromise between the factors. Actually, there are three basic encoding methods with respect to part of the criteria mentioned above: single output encoding, binary encoding, and ortho-encoding (Fig1, Fig 2, and Fig 3).



The first one called single output encoding is based on speed criterion. There is only one node to represent the output layer, the class space. Due to normalization required in Neural Networks, the

output values derived from the activation function are limited \$(0,1)\$. This means on the continuous one-dimensional space from 0 to 1, the output classes are represented as a discrete region, which is expressed as an equal length \$\delta=1/M\$, where \$M\$ means the number of the classes (Fig. 1).

The second encoding system called binary encoding is based on the second criterion, the maximum scatter degree evaluated with Euclidean distance between vectors. Form 2-dimensional demonstration (Fig. 2), one can get an impression that the vectors to express the maximum scatter locate at the four corners, which expressed by coordination as \$(0,0)\$, \$(0,1)\$, \$(1,0)\$, and \$(1,1)\$. One can easily recognize that the coordination is the binary forms of integer number 0, 1, 2, and 3. In the 3-dimensional situation (Fig. 3), however, the points with the maximum scatter degree each other are \$(0,0,0)\$, \$(0,0,1)\$, \$(0,1,0)\$, \$(0,1,1)\$, \$(1,0,0)\$, \$(1,0,1)\$, \$(1,1,0)\$, and \$(1,1,1)\$, which can be composed in decimal form as 0,1,2,3,4,5,6, and 7, respectively. According to this principle, only \$\log\_2 M\$ output nodes are required to represent \$M\$ classes.

Taking into account the third and fourth criteria, we can design another encoding system, ortho-encoding method. In this system, the number of outputs is equal to the number of classes, but the code for class \$n\$ consists of a 1 value for the first \$n\$ outputs and a 0 value for the remaining outputs. The examples of the code vector in 2-dimensional and 3-dimensional situations (Fig 2 & Fig. 3) can be found from figure labeled with bold lines. This method results in a bigger codeword length than the above two ways, and also in a larger Hamming distance for the output representations of the classes than in the previous two cases.

The most important character existing in the third method is its capability to detect the mixture features. According to this method, each output can be thought of as a membership value to a particular class. When a sigmoid activation function is used, these membership values are linearly proportional. Thus, a pixel can have a very high value in one node and low values in the others denoting a strong likelihood of belonging to that one class. If it has two high outputs then a mix of two classes is detected, and so on.

**Table 1:** The feature description about the data composting the test images (Adopted from ENVI)

|   |  |
|---|--|
| A | aspenlf2.spc Aspen_Leaf-B DW92-3,        |
| B | blackbru.spc Blackbrush ANP92-9A leaves, |
| C | bluespru.spc Blue_Spruce DW92-5 needle,  |
| D | cheatgra.spc Cheatgrass ANP92-11A mix,   |
| E | drygrass.spc Dry_Long_Grass AV87-2,      |
| F | firtree.spc Fir_Tree IH91-2 Complete,    |
| G | Grass.spc Lawn_Grass GDS91 (Green),      |
| H | juniper.spc Juniper_Bush IH91-4B whol,   |
| I | maplelea.spc Maple_Leaves DW92-1,        |
| J | pinonpin.spc Pinon_Pine ANP92-14A ndl,   |
| K | rabbitbr.spc Rabbitbrush ANP92-27 whol,  |
| L | russiano.spc Russian_Olive DW92-4}       |

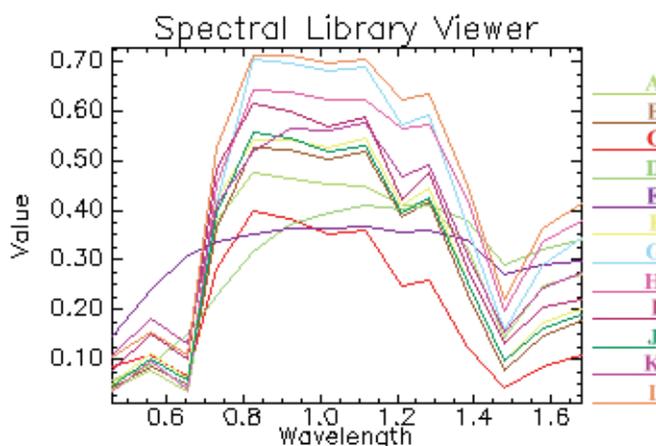


Fig 4. Spectral curves associated with selected features

### 3 Experiments and results

#### 3.1 Design of experiments

Network testing is conducted using a program called NN4RS, which is developed in Visual C++ (6.0) based on Windows systems. To test the program, a series of test images have been used. They are generated as multi-spectral images on the base of real spectral library (Fig. 4), collected from ENVI library, with a mixture of various levels of noise to simulate the situations in the real-world. These images can be used to test the effectiveness of NN4RS, with respect to the different output layer encoding methods: single

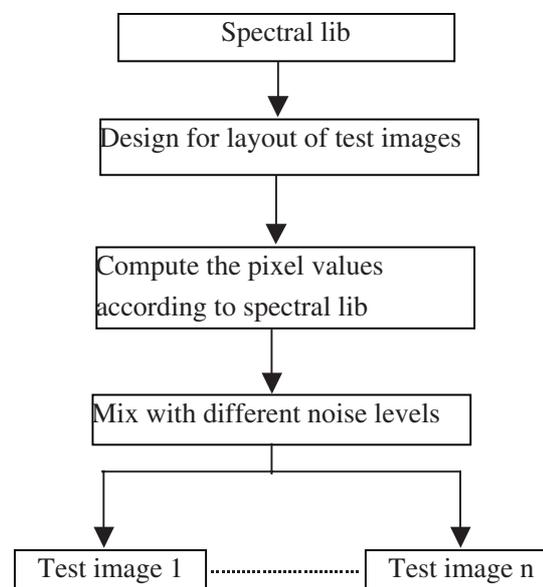


Fig 5. The general procedure for design of the test images

output encoding, binary encoding, and ortho-encoding method. The general procedure is demonstrated in Fig 5.

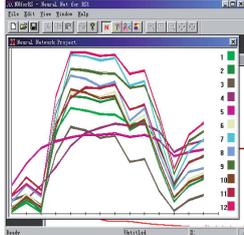
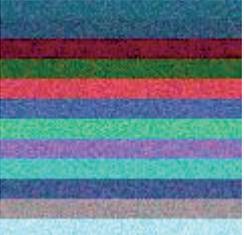
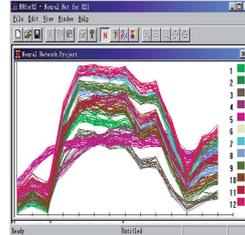
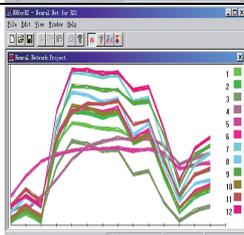
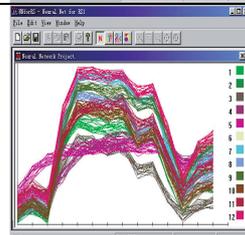
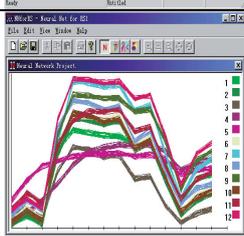
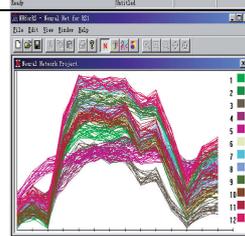
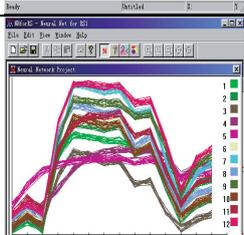
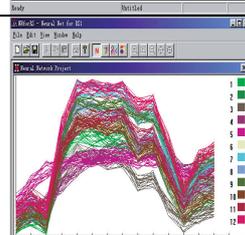
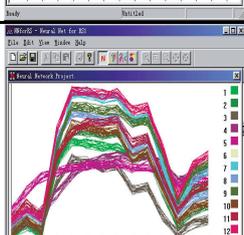
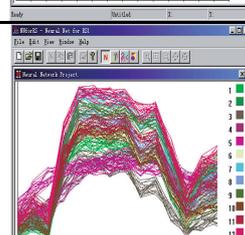
#### 3.2 Test images and training data sets

The test images include different levels of noise. Therefore, they can be used to test the relevant abilities of recognition according to the different output encoding methods. To simulate the situations in the real world, even more complicated than the real ones, a special spectral library, which includes 12 ground features extracted from a USGS Vegetation Spectral Library, is chose due to its similarity between the features. This similarity usually makes the traditional classification, such as Maximum Likelihood, SAM, difficult to distinguish features one from another. Table 3 shows the test images (120x120 pixels) with different noise levels, and their associated spectral curves adopted as training data.

**Table 2:** The wavelength ( $\mu m$ ) details about bands in the test images

|               |               |                |                |                |                |                |
|---------------|---------------|----------------|----------------|----------------|----------------|----------------|
| <b>Band 1</b> | <b>Band 2</b> | <b>Band 3</b>  | <b>Band 4</b>  | <b>Band 5</b>  | <b>Band 6</b>  | <b>Band 7</b>  |
| 0.4603        | 0.5587        | 0.6579         | 0.7326         | 0.8283         | 0.9242         | 1.0203         |
| <b>Band 8</b> | <b>Band 9</b> | <b>Band 10</b> | <b>Band 11</b> | <b>Band 12</b> | <b>Band 13</b> | <b>Band 14</b> |
| 1.1163        | 1.2124        | 1.2812         | 1.3804         | 1.4798         | 1.5792         | 1.6786         |

**Table 3:** The test images with noise and training data sets

| Noise level: 1%~5%  |   | Noise level: 6%~10%  |   |
|---|---|--|---|
| Test images   | Training data sets  | Test images  | Training data sets  |
|    |    |    |    |
|   |   |   |   |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 3.3 Experimental results

Table 4 & 5 show the confusion matrix expression from the classification results. From these results, the final conclusions can be conducted from four perspectives to assess effectiveness of the classification system, NN4RS: accuracy perspective, convergence perspective, mixture detection perspective, and confidence perspective.

**Table 4:** The confusion matrix of classification. Parameters: source image with 1% noise; 3-layer network; 21 nodes in hidden layer; binary encoding for output layer; learning rate: 0.05 & momentum: 0.003; global confidence: 5.2%; convergence: 1.7%

|                                | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | Row Total                  | Commission Error [%] |
|--------------------------------|------|------|------|------|------|------|------|------|------|------|------|------|----------------------------|----------------------|
| 1                              | 1200 |      |      |      |      |      |      |      |      |      |      |      | 1200                       | 0                    |
| 2                              |      | 1200 |      |      |      |      |      |      |      |      |      |      | 1200                       | 0                    |
| 3                              |      |      | 1200 |      |      |      |      |      |      |      |      |      | 1200                       | 0                    |
| 4                              |      |      |      | 1200 |      |      |      |      |      |      |      |      | 1200                       | 0                    |
| 5                              |      |      |      |      | 1200 |      |      |      |      |      |      |      | 1200                       | 0                    |
| 6                              |      |      |      |      |      | 1186 |      |      |      | 14   |      |      | 1200                       | 1.17                 |
| 7                              |      |      |      |      |      |      | 1200 |      |      |      |      |      | 1200                       | 0                    |
| 8                              |      |      |      |      |      |      |      | 1200 |      |      |      |      | 1200                       | 0                    |
| 9                              |      |      |      |      |      |      |      |      | 1200 |      |      |      | 1200                       | 0                    |
| 10                             |      |      |      |      |      |      |      |      |      | 1200 |      |      | 1200                       | 0                    |
| 11                             |      |      |      |      |      |      |      |      |      |      | 1200 |      | 1200                       | 0                    |
| 12                             |      |      |      |      |      |      |      |      |      |      |      | 1200 | 1200                       | 0                    |
| Column Total                   | 1200 | 1200 | 1200 | 1200 | 1200 | 1186 | 1200 | 1200 | 1200 | 1214 | 1200 | 1200 | <i>Total pixels: 14400</i> |                      |
| Omission Error[%]              | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 1.15 | 0    | 0    |                            |                      |
| <b>Final Accuracy: 99.90 %</b> |      |      |      |      |      |      |      |      |      |      |      |      |                            |                      |

**Table 5:** The confusion matrix of classification. Parameters: source image with 10% noise; 3-layer network; 28 nodes in hidden layer; binary encoding for output layer; learning rate: 0.05 & momentum: 0.003; global confidence: 5.8%; convergence: 2.0%

|                                | 1    | 2     | 3     | 4    | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | Row Total                  | Commission Error [%] |
|--------------------------------|------|-------|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|----------------------------|----------------------|
| 1                              | 917  |       | 20    |      | 178   | 1     | 17    | 2     | 59    | 1     |       | 5     | 1200                       | 23.58                |
| 2                              |      | 421   | 8     | 1    |       | 115   | 4     | 64    |       | 587   |       |       | 1200                       | 64.92                |
| 3                              | 24   | 134   | 921   |      |       | 20    | 50    | 11    | 1     | 17    | 22    |       | 1200                       | 23.25                |
| 4                              |      |       |       | 1187 | 6     |       | 1     | 2     |       |       |       | 4     | 1200                       | 1.08                 |
| 5                              | 40   |       |       | 1    | 1150  |       | 9     |       |       |       |       |       | 1200                       | 4.17                 |
| 6                              | 1    | 163   | 1     | 21   |       | 152   | 1     | 291   |       | 520   | 1     | 49    | 1200                       | 87.33                |
| 7                              | 1    |       | 11    | 1    | 24    | 28    | 765   | 11    | 110   | 27    | 166   | 56    | 1200                       | 36.25                |
| 8                              | 1    | 3     | 1     | 5    |       | 11    |       | 1013  | 4     | 103   | 8     | 51    | 1200                       | 15.58                |
| 9                              | 23   |       |       | 1    | 4     |       | 1     | 49    | 1036  |       | 1     | 85    | 1200                       | 13.67                |
| 10                             |      | 119   |       | 8    |       | 91    |       | 255   |       | 717   | 3     | 7     | 1200                       | 40.25                |
| 11                             |      | 2     | 103   |      | 2     | 6     | 110   | 11    | 12    | 112   | 820   | 22    | 1200                       | 31.67                |
| 12                             |      |       |       | 2    | 1     |       |       | 153   | 89    |       |       | 955   | 1200                       | 20.42                |
| Column Total                   | 1007 | 842   | 1065  | 1227 | 1365  | 424   | 958   | 1862  | 1311  | 2084  | 1021  | 1234  | <i>Total pixels: 14400</i> |                      |
| Omission Error[%]              | 8.94 | 50.00 | 13.52 | 3.26 | 15.75 | 64.15 | 20.15 | 45.60 | 20.98 | 65.60 | 19.69 | 22.61 |                            |                      |
| <b>Final Accuracy: 69.82 %</b> |      |       |       |      |       |       |       |       |       |       |       |       |                            |                      |

1. **Accuracy perspective.** The classification results from the networks armed with a single output encoding system are the worst, compared to the other two encoding systems. With respect to the increases of noise within test images, from 1% to 10%, the recognition accuracy gets down, from 99.9% to 69.8%, in the networks armed with binary encoding output layer.
2. **Convergence perspective.** The experiment reveals that the training procedures can be stopped at good convergence, from 1.7% to 5.6%, when binary encoding method and ortho-encoding method are used. However, when single output encoding method is used, the training procedure shows unstable results, sometimes with good convergence, while sometimes even divergent.
3. **Mixture detection perspective.** As we mentioned earlier, the ability of detecting feature mixture only exists in neural classifiers with ortho-encoding systems. This character can be used to show the local confidence in the future work.
4. **Confidence perspective.** The global confidence can only be measured in binary encoding method and ortho-encoding method. The experimental results show that ortho-encoding method can derive better convergence than binary encoding method, which is 95.3% in the former one in average, while 97.4% in the latter one.

## References

- [1] Atkinson, P.M. and Tatnall, A.R.L., 1997. Neural networks in Remote Sensing. *International Journal of Remote Sensing*, 18(4), pp. 699-709.
- [2] Benediktsson, J.A., Swain, P.H. and Ersoy, O.K., 1990. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *I.E.E.E. Transactions on Geoscience and Remote Sensing*, 28, pp. 540-551.
- [3] Chen, K.S., Yen, S.K. and Tsay, D.W., 1997. Neural classification of SPOT imagery through integration of intensity and fractal information. *International Journal of Remote Sensing*, 18(4), pp. 763-783.
- [4] Civco, D.L., 1993. Artificial neural networks for land cover classification and mapping. *International Journal of Geographic Information Systems*, 7, pp. 173-186.
- [5] Fischer, M.M. and Stauffer, P., 1999. Optimization in an error backpropagation neural network environment with a performance test on a spectral pattern classification problem. *Geographical Analysis*, 31(3), pp.
- [6] Heermann, P.D. and Khazenie, N., 1992. Classification of multispectral remote sensing data using a back-propagation neural network. *I.E.E.E. Transactions on Geoscience and Remote Sensing*, 30, pp. 81-88.
- [7] Kaminsky, E.J., Barad, H. and Brown, W., 1997. Textural neural network and version space classifiers for remote sensing. *International Journal of Remote Sensing*, 18(4), pp. 741-762.
- [8] Kanellopoulos, I. and Wilkinson, G.G., 1997. Strategies and best practice for neural network image classification. *International Journal of Remote Sensing*, 18(4), pp. 711-725.
- [9] Kavzoglu, T. and Mather, P.M., 1999. Pruning artificial neural networks: An example using land cover classification of multi-sensor images. *International Journal of Remote Sensing*, 20(14), pp. 2787-2803.
- [10] Mather, P.M., Tso, B. and Koch, M., 1998. An evaluation of Landsat TM spectral data and SAR-derived textural information for lithological discrimination in the Red Sea Hills, Sudan. *International Journal of Remote Sensing*, 19(4), pp. 587-604.
- [11] Miller, D.M., Kaminsky, E.J. and Rana, S., 1995. Neural network classification of remote-sensing data. *Computers and Geosciences*, 21(3), pp. 377-386.
- [12] Paila, J.D. and Schowengerdt, R.A., 1995. A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *International Journal of Remote Sensing*, 16(16), pp. 3033-3058.
- [13] Pao, Y.-H., 1989. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley.
- [14] Yang, H., Meer, F.V.D. and Bakker, W., 1999. A back-propagation neural network for mineralogical mapping from AVIRIS data. *International Journal of Remote Sensing*, 20(1), pp. 97-110.