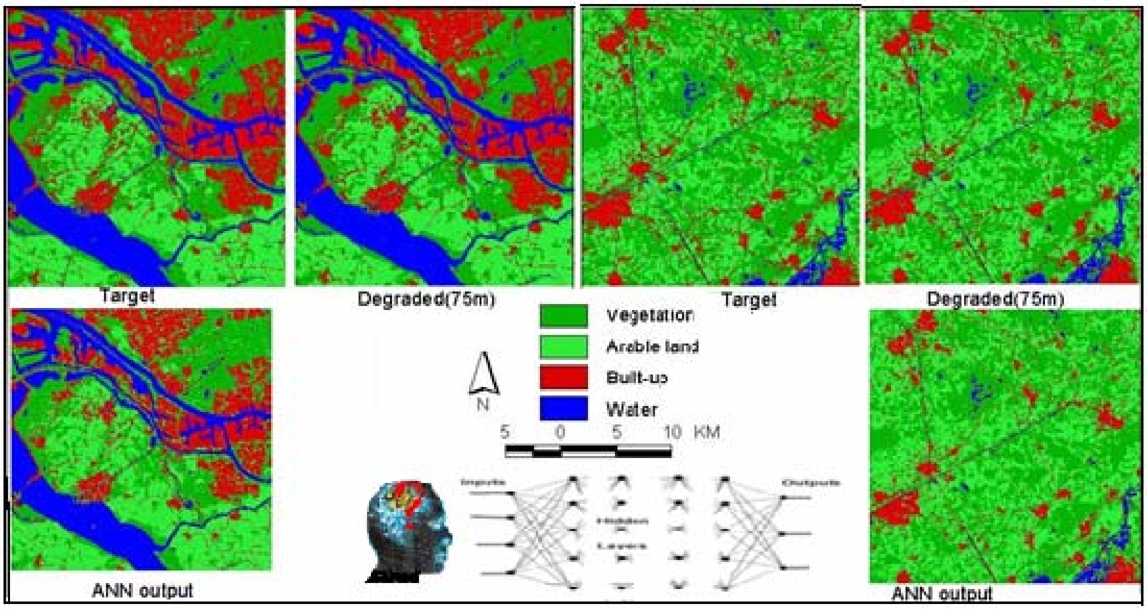
Centre for Geo-Information

Thesis Report GIRS-2006-08

---

# POSSIBILITIES AND LIMITATIONS OF ARTIFICIAL NEURAL NETWORKS FOR SUB-PIXEL MAPPING OF LAND COVER

Demeke Nigussie Alemu



March 2006

WAGENINGEN UNIVERSITY
WAGENINGEN UR

**……………………to my family**

# Possibilities and Limitations of Artificial Neural Networks for Sub- Pixel Mapping of Land Cover

Demeke Nigussie Alemu

Registration number 68 05 24 010 010

Supervisors:

Raul Zurita-Milla
J.G.P.W.Clevers

A thesis submitted in partial fulfillment of the degree of Master of Science at Wageningen University and Research Centre, The Netherlands.

March 2006
Wageningen, The Netherlands

Thesis code number: GRS-80436
Wageningen University and Research Centre
Laboratory of Geo-Information Science and Remote Sensing
Thesis Report: GIRS-2006-08

# Acknowledgements

This thesis work would not have been executed without the technical and moral support of many of the GRS Department staffs, friends, and families.

First, I would like to express my sincere gratitude and appreciation to my supervisors Mr. Raul Zurita-Milla and Dr. Jan Clevers who gave me a continuous support, encouragement and constructive criticisms to finalize this work. I am therefore highly indebted to them.

I would like to thank Willy Ten Haaf who gave me continuous advices whenever I encountered problems related to academic and financial cases.

I am very grateful to my employer, the Ethiopian Agricultural Research Organization, for giving me the opportunity to pursue my M.Sc. degree study at Wageningen University.

I am very grateful to my father Nigussie Alemu, my mother Yeshimebet Negalign, my wife Sosina Reda, and all my brothers and sisters who encouraged me throughout my study in The Netherlands.

Finally yet importantly, I would like to thank my friends particularly Abebe Mohammed and Alemu Gonsamo who encouraged me and gave me friendly advices.

# Abstract

Up-to-date and reliable land cover information is vital for many decision-making processes. Though the developments of satellite remote sensing greatly improved land cover information acquisition, there are still unaddressed problems to achieve the intended accuracy level. One of such problems is the mixed pixel. Soft-classification techniques were introduced to address the problem; but they do not show the spatial location of the classes' proportions in a pixel. Sub-pixel mapping techniques that transform the soft-classification results into hard-class maps at sub-pixel scale were introduced to address the drawbacks of soft-classifications. In this thesis, an artificial neural network (ANN), specifically the feedforward backpropagating neural network (FFBPNN) was used for sub-pixel mapping.

To prepare the input fraction images, which are to be treated as soft-classification results, the LGN5 database was thematically aggregated into 2, 4 and 8 thematic classes. Then, these thematically aggregated data were spatially aggregated into three spatial resolution sizes, namely, 75m, 150m, and 300m. The three chosen thematic classes and three spatial resolutions end up in 9 different combinations that are considered as study cases in this thesis work. The fraction images were used to train several FFBPNN. After training and selecting the best network, each case was simulated using the fraction image of two small sites to reconstruct the 25mx25m sub-pixel hard-class map. These sites were selected from the Southwest and Southeast of the Netherlands to examine the effect of the land cover heterogeneity. The overall accuracies obtained revealed that the response of the network was highly influenced by the spatial frequency, shape and size of the different land covers. Moreover, it revealed that most of the errors are on class boundaries where highly mixed pixels are expected. The accuracies achieved had a wide range depending on the complexity of the cases. In general, the overall accuracies ranged from 38.05 % (complex cases) to 98.97 % while the Kappa coefficients ranged from 0.14 to 0.97. Although it was not possible to exhaustively explore all network architectures for the various case studies, the results achieved demonstrated the potential of the FFBPNN for sub-pixel mapping.

**Keywords:** Sub-pixel, land cover, neural networks, class fraction, target

# Table of Contents

# List of Figures

.

# List of Tables

.

# Chapter 1
# INTRODUCTION

## 1.1 Background

Remotely sensed images are an important source of information for land cover mapping at local, regional and global scales. The advent of satellite remote sensing, coupled with other parallel developments in computer technology and processing techniques, has enhanced and diversified the potential application of satellite remote sensing to extract land cover information. However, due to hardware and cost limitations, imaging systems often provide us with multiple low resolution images (Molina *et al.* 2003).

Remote sensing images usually exhibit either high spectral resolution and low spatial resolution, or low spectral resolution (broadband) and high spatial resolution, but not both (Ranchin *et al.* 2003). There is a growing effort to combine such different qualities of sensors. This requires appropriate and advanced processing techniques to match the spectral and spatial tradeoff (Tatem *et al.* 2003) and extract maximum information from it. Amongst these processing techniques, sub-pixel mapping using class proportion fractions obtained from a sub-pixel classification, is becoming one of the subjects of research. Without solving the problem of mixed pixels, it is likely that the full potential of remote sensing as a source of land cover information will remain unrealized (Foody 2004).

Sometimes, conversion of data from coarse to fine resolution or vise-versa is a necessary step in order to match and integrate different data with different resolution and application scale requirements. But coarse spatial resolution data may be dominated by mixed pixels (e.g. Figure 1.1); and these mixed pixels are often a source of error as pure pixels are often assumed (Atkinson *et al.* 1997; Foody 1997; 2004). Though the mixed pixel problem is relatively small in images with high spatial resolution, acquiring such data has limitations such as cost, temporal resolution and high volume of data. Sub-pixel mapping from classes proportion fractions (typically from soft classification) is one of the newly introduced techniques to overcome mixed pixel problems in low spatial resolution images.

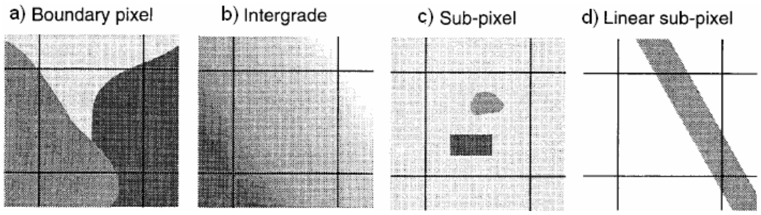


**Figure 1.1 Four causes of mixed pixels (Fisher 1997)**

Sub-pixel mapping enables to locate the class proportions at sub-pixel level and minimizes such information loss. Some researches on this technique proved the possibility of creating a super resolution map from a coarse resolution map. Tatem *et al.* (2001a; 2001b; 2001c; 2002; 2003), for instance, introduced a Hopfield Neural Network (HNN) for sub-pixel mapping and demonstrated its potential for various land cover applications. Determining sub-pixel information is considered as down scaling (Liang 2004).

The focus of this thesis is to examine the potentials and limitations of artificial neural networks, specifically feedforward backpropagating neural networks (FFBPNN), for sub-pixel mapping. The backpropagating method, presented in 1986 by Rumelhart et al. is one of the most commonly used neural networks (Paola and Schowengerdt 1995). Though the fraction images used in this case are from existing land cover maps, the demonstration is supposed to be applicable also to sub-pixel mapping from soft-classification results of remotely sensed images.

## 1.2 Problem definition

### 1.2.1 General Problem – mixed pixels

Spatial scaling is an important aspect in remote sensing and GIS analysis. For integrating and matching various data obtained at local, regional or global scale, re-scaling is a fundamental step. Remotely sensed imagery with moderate and coarse spatial resolutions is ideal for regional and global scales since large contiguous regions are completely covered and data are provided synoptically; however, such a coarse resolution images have mixed pixels (Atkinson et al, 1997). Soft classification algorithms may be more appropriate than the per-pixel classifiers for classification of images dominated by mixed pixels (Xu *et al.* 2005).

Moreover, aggregating of images to coarser spatial resolutions often causes a mixed pixel problem. Most images acquired from remote sensing satellites also have a mixed pixel problem. Occurrence of mixed pixels in remote sensing images is a major problem particularly at coarse spatial resolutions (Foody 1997; Kasetkasem *et al.* 2005).

### 1.2.2 Specific problem: sub-pixel mapping

The solution to the mixed pixel problem centers on sub-pixel classification, which may be used to predict the proportion of each class within each pixel (Tatem *et al.* 2001d; 2003). Sub-pixel classification techniques can only provide the estimates of class compositions of a pixel. Locating class proportions within each pixel is among the limitations of sub-pixel classification. It does not show the exact distribution of these class components within each pixel. Therefore, the resultant prediction still contains uncertainty (Tatem *et al.* 2003). Moreover, Tatem *et al.* (2003) and Foody (2004) stated that the complexity of the datasets produced from soft classification may be large and more difficult to interpret and this puts many GIS users off using soft classification techniques since users typically want easy to use map products. To overcome such drawbacks, sub-pixel mapping techniques are suggested by many authors. Atkinson (1997) introduced this concept of sub-pixel mapping (Mertens et al. 2004). The key problem of sub-pixel mapping is determining the most likely locations of the fractions of each land cover class within the pixel (Verhoeye and De Wulf 2002). Sub-pixel mapping, which is the topic of this thesis, is clearly distinct from sub-pixel classification: sub-pixel mapping uses fractions (e.g. resulting from sub-pixel classification) in order to retrieve an appropriate spatial location and transform into a finer scaled hard classification (Mertens et al. 2004).

To overcome the limitation of soft-classification in a mixed pixel problem, researchers have developed different techniques. Sub-pixel mapping using a neural network and particularly the Hopfield Neural Network introduced by Tatem *et al.* (2001a) is one of those promising techniques. The solution for sub-pixel mapping problems is mainly based on an assumption of spatial dependency within and between pixels; i.e., objects close together should be more alike than those further apart (Atkinson *et al.* 1997; Tatem *et al.* 2001a; Verhoeye and De Wulf 2002; Mertens *et al.* 2003)( see Figure 1.2).

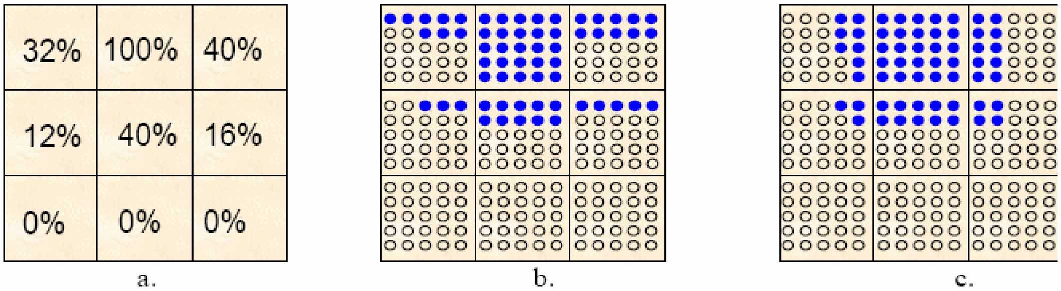


Figure 1.2.a) a 3x3 grid with class fractions is discretised into 5x5 grids, (b) and (c) without and with considering spatial dependency (Verhoeye and De Wulf, 2000 & 2002, After Atkinson, 1997).

Since sub-pixel mapping is a relatively recent technique, the possibilities and limitations of the various sub-pixel mapping techniques are not exhaustively studied yet. Studies on the application of ANN for sub-pixel mapping are very limited. Although the potential of sub-pixel mapping using ANN is acknowledged by some studies, researches using real data are not exhaustively explored. Existing studies are conducted mostly based on a very small area and/or very few classes. This does not exhaustively show possibilities and limitations of ANN for sub-pixel mapping. The performance also may depend on the study area (difference in spatial frequency). Therefore, this study will try to explore the performance of FFBPNN in a relatively large study area; and see the influence of changing the number of classes using a real land use database in the context of Netherlands' land cover.

## 1.3 Study area

For this study, training samples were extracted from the land cover data of the entire Netherlands and two subset locations were used for simulation (real implementation of sub-pixel mapping) purposes. The trained neural network was presented (simulated) with two new data sets that cover a small part of the country. The two locations that have relatively heterogeneous land cover were selected from the Southwest and Southeast part of the country (**Figure 1.3**).

The geographic location of site 1 is 51°46’3’’N-51°59’17’’ N and 4°1’58’’E-4°23’19’’E. Site 2 is located between 51°11’38’’N-51°24’55’’N and 5°40’29’’E-6°1’52’’E. The two sites have 993x993, 984x984, and 972x972 pixels for the study cases with 75mx75m, 150x150m and 300m low resolution images, respectively.

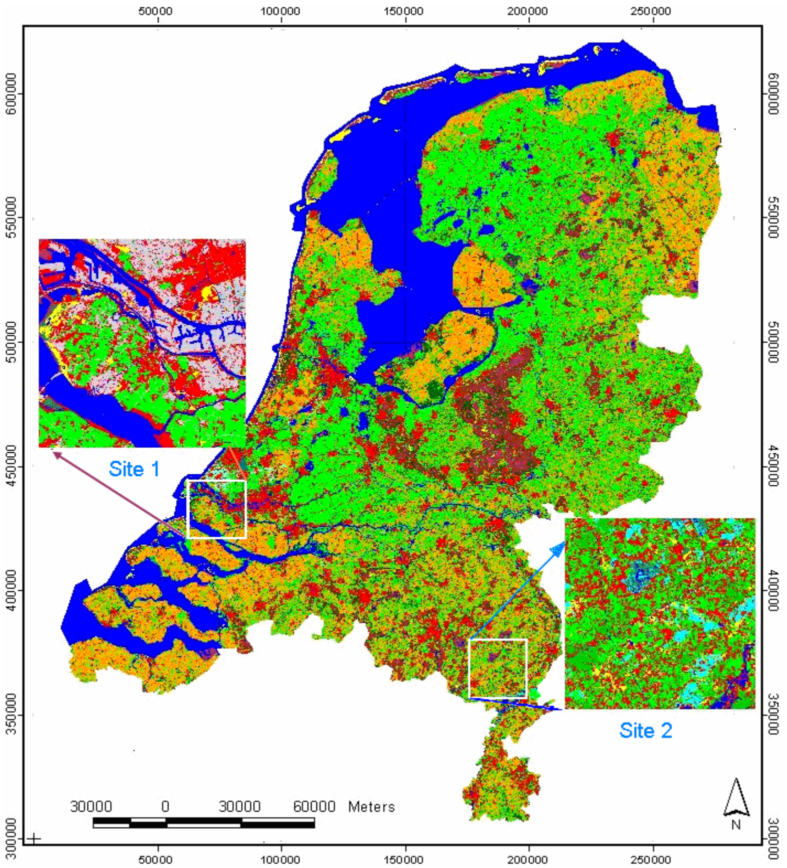


Figure 1.3 Location map of the study area

## 1.4 Objectives

The main objective of this thesis work is to explore the possibilities and limitations of artificial neural networks (ANN) for sub-pixel mapping with a specific application to the land use database of The Netherlands (LGN). The performance of feedforward backpropagating neural networks (FFBPNN) in predicting locations of class proportions within each pixel and the accuracy of the resulting super-resolution map will be assessed at various levels (the response with change of degradation cell size and number of classes). It will mainly focus on the interrelated techniques of sub-pixel mapping and deriving a super-resolution land cover map from up-scaled (degraded) images. Sub-objectives are:

- To assess the effect of change in spatial resolution and number of thematic classes on the performance of the FFBPNN for sub-pixel mapping;
- to demonstrate the approach on sub-pixel mapping using a neural network for multi-resolution and multi-class data;
- to see the effect of varying spatial and thematic aggregation levels for sub-pixel mapping using FFBPNN with respect to shape, size and spatial frequency.

## 1.5 Research questions

Aforementioned objectives will enable to answer questions like:

- What are the overall limitations and possibilities of the FFBPNN for sub-pixel mapping?
- How is the capability of the FFBPNN for pattern/location prediction of sub-pixels within a pixel at the different spatial and thematic aggregation levels?
- Is there a change in accuracy with change in spatial frequency, shape and size of the cover types (e.g. urban and agriculture)?

## 1.6 Set-up of the thesis report

This thesis report consists of five chapters including this one. The second chapter deals with the general concepts and review of related works focusing on sub-pixel mapping and artificial neural networks.

The third chapter covers the materials and methods employed. This includes data preparation (thematic and spatial aggregation, training, testing and validation sample selection), ANN implementation (neural network training and evaluation) and accuracy assessment.

The fourth chapter presents and discusses the results and findings of this thesis work. Finally, the fifth chapter presents the conclusions and recommendations of this thesis work.

# Chapter 2

# GENERAL CONCEPTS AND REVIEW OF LITERATURES

## 2.1 Neural networks

### 2.1.1 Main concepts

The history of artificial neural networks (ANN) is described by various authors. According to Martin et al (1997), the modern view of neural networks began in the 1940s with the work of Warren McCulloch and Walter Pitts who showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function and their work is often acknowledged as the origin of the neural network field. Researchers, inspired by the biological neuron, were putting efforts to mimic the complex human brains. Even though ANNs do not approach the complexity of the brain, there are, however, two key similarities between biological and ANNs; first, the building blocks of both networks are simple computational devices (although artificial neurons are much simpler) that are highly interconnected; second, the connections between neurons determine the function of the network (Martin et al 1997).

Haykin (1994) defined a neural network as a massively parallel distributed processor that has a natural tendency for storing empirical knowledge and making it available for use. The author further stated that it resembles the brain in the following two respects:

- The network acquires knowledge from its environment using a learning process (algorithm)
- Synaptic weights, which are interneuron connection strengths, are used to store the learned information.

Artificial neural network techniques are being used for wide varieties of disciplines including remote sensing and GIS. Despite its wide application and popularity, ANNs have still many limitations. Verbeke *et al.* (2004) enumerated some of the common limitations and the corresponding efforts to tackle these drawbacks. Some of the limitations are:

- ANNs are black-box models of poor semantic quality: information learned is coded into weight values that are hard to interpret;
- before training, the network architecture, input/output representation, learning algorithms and parameters have to be selected; literature provides only rules of thumb that are often contradictory;
- training an ANN is a time-consuming trial and error process;
- it is not possible to reuse trained neural networks when new (although analogous) problems are encountered;
- incorporating prior knowledge is difficult;
- the unpredictable network behavior after random weight initialization.

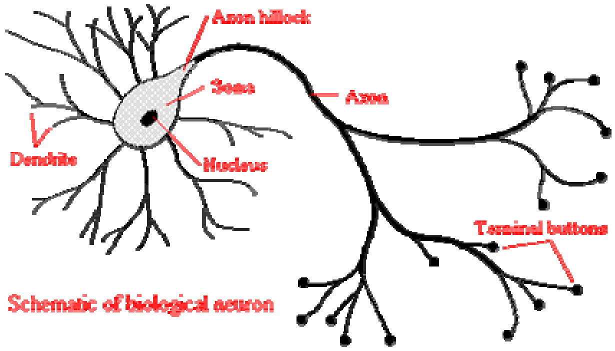


Figure 2.1 The basic features of biological neuron

Source: Fraser (1998)

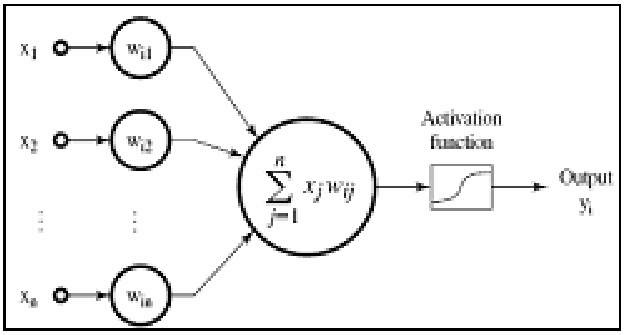


Figure 2.2 A simple structure of an ANN neuron

In addition, Paola and Schowengerdt (1995) have enumerated some of the advantages and disadvantages of neural network methods over traditional classifiers. These advantages are its non-parametric nature, arbitrary decision boundary capabilities, easy adaptation to different types of data and input structures, fuzzy output values that can enhance classification, and good generalization for use with multiple images. The disadvantages of the method are slow training time, inconsistent results due to random initial weights, and the requirement of obscure initialization values (e.g. learning rate and hidden layer size) (Paola and Schowengerdt 1995).

### 2.1.2 Feedforward backpropagating neural network

Among the many varieties of neural network architectures, the feedforward backpropagating neural networks (FFBPNN) are considered as the most popular algorithms (Kavzoglu and Mather 2003). As this particular algorithm is employed in this thesis work, it is briefly described. The FFBPNN is one of the widely used methods in remote sensing and GIS. A feedforward neural network belongs to the supervised training category in which inputs and desired outputs (target) are provided for the network. It consists of an input layer of non computational units, one or more hidden layers of computational units and an output layer of computational units (Kavzoglu and Mather 2003; Foody 2004)(see Figure 2.3). The size of the input layer is defined by the size of the input feature vectors and the output layer size is determined by the number of classes; however, no formula exists for determining the number of hidden nodes, although several heuristics have been proposed (Logar et al. 1994).

In FFBPNN, an input is presented to the network and passed forward. The output of the ANN is compared with the desired output/target and then the error is transferred backward through the network until it stops with an acceptable error level or other stopping criteria of network training. The neural network implementation has the following general steps:

- Initialize the weights
- Present input and the desired output vectors to the network
- Propagate these inputs forward through the network
- Calculate the error between the network output and the target
- Propagate the error back through the network

- Adjust the weights to minimize errors
- Repeat steps starting from 2 through 6 until the error level is within the acceptable range.

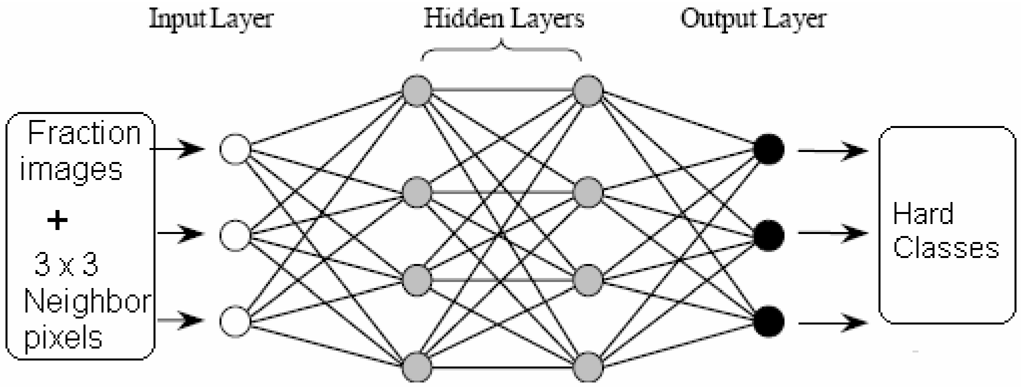


Figure 2.3 A typical four layer feedforward neural network

### 2.1.3 Selecting neural network architecture

The architecture of a neural network depends on a number of factors mainly depending on the nature of the problem and the available data. Most of the parameters to define neural network architecture are set based on a series of try and error experiments. Several researches have been done to minimize such time consuming trial runs and to improve the generalization performance of ANNs. Current research mostly concentrates on the optimal setting of initial weights, optimal learning rates and momentum, finding optimal ANN architectures using pruning techniques and construction techniques, sophisticated optimization techniques, and adaptive activation functions (Engelbrecht 2001).

A comprehensive study and review on selecting and designing a proper ANN architecture relevant to geo-information particularly to remote sensing is done by Paola and Schowengerdt (1995), Kavzoglu and Mather (2003) and Kanellopoulos and Wilkinson (1997). These authors reviewed various previous studies and experimented on various settings of important neural network definitions for remote sensing, and provided a kind of guideline for similar studies. Some of the issues discussed include number of input, hidden and output nodes, learning rate and momentum, initial weight range, number of training samples, stopping criterion, output encoding and validating.

#### 2.1.3.1 Input and output layer nodes

The first layer in a neural network is the input layer that is responsible for distributing the input data values to the first hidden layer. Unlike the hidden and output layer nodes, the input layer nodes are non-computing. The number of input layer nodes generally corresponds to the number of input data. This number also depends on the input encoding technique used (Kavzoglu and Mather 2003). The size of the input and output layer nodes is determined by the size of input and output vectors, respectively.

#### 2.1.3.2 Number of hidden layers and nodes

The number of hidden layers and nodes determine the generalization capacity of the network. The optimum structure depends on a number of factors such as the number of input nodes, output nodes and training sample size. A feedforward neural network often has one input layer, one or more hidden layers and one output layer. The structure of the first and the last layer of the neural network are controlled by external factors. The optimum number of hidden layers and their size must be determined experimentally depending on the problem at hand (Paola and Schowengerdt 1995).

.

The processing elements in neural networks in various literatures are called nodes, units or neurons. One of the challenging jobs in neural network designing is to determine the number of hidden layers and the number of nodes in each hidden layer. Although the generalization of a network depends on many factors, like training sample size, number of iterations, etc, determining the optimum size of the hidden layer is an important aspect that determines the learning and generalization capability of the network. The number of hidden nodes as stated in many literatures is dependent on the training sample size and on the number of input and output nodes.

Generally, one single hidden layer can solve most classification and information extraction problems (Paola and Schowengerdt 1995; Kanellopoulos and Wilkinson 1997; Han *et al.* 2003), but if samples are comparatively complex, a two-hidden-layer neural network can be adopted (Han *et al.* 2003). The number of nodes in the hidden layer(s) should be large enough for more flexibility in partitioning the decision space (Paola and Schowengerdt 1995) and for the correct representation of the problem, but at the same time, should be low enough to have adequate generalization capabilities (Kavzoglu and Mather 2003). However, networks that are too small or too large will likely encounter underfitting or overfitting problems, respectively (Paola and Schowengerdt 1995; Foody and Boyd 1999; Mathworks 2005; and Kanellopoulos and Wilkinson 1997). Kanellopoulos and Wilkinson (1997) suggested the number of hidden layer nodes should be at least equal to double of the number of inputs and perhaps four times as many to be safe (see Table 2-1). Atkinson *et al.* (1997), in their study on mapping sub-pixel proportional land cover, concluded that increasing the number of nodes in the hidden layer did not affect performance. Unfortunately, there are no specific theoretical guidelines for the size of the hidden layer (Paola and Schowengerdt 1995).

Concerning the number of hidden layer nodes, as can be seen in Table 2-1, a number of alternative formula suggested by various authors are reviewed in Kavzoglu and Mather (2003). For details on the different heuristics suggested by different authors, readers are referred to this source too. For this thesis work, formula 3.3 indicated in chapter 3 was used with some modification of Ripley's formula listed in Table 2-1.

Table 2-1 Heuristics proposed to compute the optimum number of hidden layer nodes. (Source: Kavzoglu and Mather, 2003)

| Heuristic | Source |
|---|---|
| $2N_i$ or $3N_i$ | Kanellopoulos and Wilkinson (1997) |
| $3N_i$ | Hush (1989) |
| 2Ni +1 | Hecht-Nielsen (1987) |
| $2N_i/3$ | Wang (1994b) |
| $(N_i + N_o)/2$ | Ripley (1993) |
| $N_{\boldsymbol{p}}/[r(Ni + N_o)]$ | Garson (1998) |
| $(2+N_o \text{ x } N_i + 1/2N_o (N_i^2 + N_i)-3)/ (N_i + N_o)$ | Paola (1994) |

*Where* $N_i$, $N_o$, $N_h$, $N_f$, *and* $N_p$ *are number input, outputs, and hidden layer nodes, node factor and the* number of training samples, *respectively*

### 2.1.3.3 Training algorithms, transfer and error functions

There are many varieties of ANN architectures, and many possible learning rules, or strategies (Atkinson *et al.* 1997). Each training algorithm has its own weakness and strength that makes it preferable for a particular application or not. The weakness and strength of the various training algorithms are provided in Mathworks (2005).

For this thesis, the Scaled Conjugate Gradient training algorithm (SCG), introduced by Moller (1993), was used. It is a fully automated, very good, general purpose training algorithm, including no user dependent parameters and avoids the time consuming line-search process (Mathworks 2005). SCG does not contain any user-dependent parameters whose values are crucial for the success of SCG (Moller 1993). Some sources also recommend using the SCG training algorithm for complex cases where memory and speed of convergence is a problem. For complex neural networks, the SCG training algorithm is suggested by Moller (1993), Mathworks (2005) and others. SCG is also one of the algorithms that work well for trainings that employ early stopping techniques (Mathworks 2005). Datta and Banerjee (2004) in their training exercise found out that SCG gives the best training result among resilient backpropagation (RP), Levenberg-Marquardt (LM), Fletcher-reeves Conjugate Gradient, and quasi-Newton training algorithms.

A detailed description of the algorithm can be found in Moller (1993). Taking into account these and other advantages of SCG over the rest of the algorithms available in the MATLAB software package, SCG (TRAINSCG) was employed in most of the training undertakings of this thesis work (see Table 4-1 and Appendix B).

*Transfer functions*

One of the major components of an artificial neuron is the transfer function (sometimes called activation function). It is a function that maps a neurons (or layer's) net output (n) to its actual output (Mathworks 2005). Among the various transfer functions, the most common is the sigmoid transfer function (Paola and Schowengerdt 1995). In this thesis, hyperbolic tangent sigmoid (TANSIG) and linear (PURELIN) transfer functions were used in the hidden and output layers, respectively. Sigmoid is a monotonic S-shaped function mapping numbers in the interval (-∞,∞) to a finite interval such as (-1 to 1) or (0 to 1) whereas a linear transfer function produces its input as its output (Mathworks 2005). TANSIG and LOGSIG transfer functions map the input to the interval (-1, 1) and (0, 1), respectively.

*Error functions*

In a neural network training process the difference between the ANN output and the desired output are computed using an error function (performance function) and the errors are backpropagated to the previous layer. There exist various types of error functions. One of these is the mean square error (MSE). It measures the network's performance according to the mean of squared errors (Mathworks 2005). This performance function was used for this thesis work.

### 2.1.4 Training Data

Neural network training requires a set of training data. This training dataset should consist of carefully selected input and the corresponding target output (if supervised training is used). As many authors agree, there is not a clear rule of thumb to use as a guide for an optimum network architecture and definition of many of the input parameters. Likewise, the nature of samples and the sample size is a subject that needs experimentation.

#### 2.1.4.1 Sampling and size

Although the size of the training dataset is very important, the characteristics and the distribution of the data as well as the sampling strategy used can affect the result (Kavzoglu and Mather 2003). The ANN training sample should be split into training, testing and validation sets. This sample partitioning into training, testing and validation set is also an important aspect that needs appropriate selection and partitioning. Some sources (e.g. Mathworks (2005)) suggest a partitioning of the total sample into ½ for training, ¼ for testing and the rest ¼ for validation. Kavzoglu and Mather (2003) reviewed several heuristics proposed by various authors for the computation of the optimum number of training samples.

Paola and Schowengerdt (1995) in their review stated that the samples must be representative of the classes. According to the authors, a study conducted using 22000 and 4200 samples did not show a difference in classification accuracy.

Kavzoglu and Mather (2003) recommend using a validation dataset around 50 to 100 samples per class to terminate the training process. In this thesis work, both the training, validation and testing data were employed partitioning them as suggested in (Mathworks 2005). Validation data are used to stop training early if the network performance on the validation data fails to improve, or remains the same for some iterations; whereas test data are used as a further check that the network is generalizing well, but do not have any effect on training (Mathworks 2005). Since use of the validation and testing sets is optional, some researchers use only training and testing datasets for the training process.

#### 2.1.4.2 Input and output encoding

Like several analysis techniques, neural network training needs data preprocessing. One of them is encoding of inputs and outputs. This preprocessing step depends on the nature of the problem, input data and the intended output. Moreover, this task also depends on the activation function to be employed. Therefore, selection of the appropriate encoding is crucial for both the network convergence speed and meeting the error goal. Paola and Schowengerdt (1995) described the various input and output encoding options being used in a neural network training process.

The encoding types commonly employed are binary, one-of-N, 0 to 1, and -1 to 1 range. It is advised, in many literatures (e.g. Paola and Schowengerdt 1995), to normalize the input data to the 0 and 1 range or -1 to 1 range to ease neural network training. For some neural networks, the -1 to 1 range is favored. For instance, Tatem *et al.* (2001a; 2001b; 2001c; 2001d; 2002; 2003) used this range. The other recommended encoding technique is commonly called one-of-N encoding. In such encoding the value of the output node corresponding to a specific class will be assigned 1 and others assigned 0. For example, in a three class case the first, second and the third classes will be represented in a 100, 010 and

001 scheme of output classes, respectively. This technique is one of the most popular methods of encoding. Kavzoglu and Mather (2003) also recommend this encoding scheme. But one of the limitations of this encoding technique is that the network size will increase with the number of classes making the training process more difficult since it becomes computationally demanding. This is also true for binary encoding. For instance, a binary encoding of four layer (band) data needs 32 nodes, which is an eight-fold increment.

### 2.1.5 Early Stopping

One of the challenging tasks in neural network training is to decide when to stop training to avoid the so called overfitting problem in neural networks. Overfitting will occur as training continues and  the network tends to fit to the noise rather than the data structure (Kavzoglu and Mather 2003). Overtraining is a common problem of backpropagating algorithms and setting up of an appropriate condition for the termination of the training session is very important (Bhattacharya and Parui 1997).

In some literatures, the terms testing and validation set are used interchangeable; i.e. there are confusions on the technique on dividing and using the datasets into three subsets; namely training, testing and validation data. Some authors use a three subset and others use a training and test set (validation set in the three subset definitions). More commonly, the whole set of available data for training is split into 1) the training-set that is used for computing the gradient and updating the network weights and biases; 2) the validation set used to stop training when the validation error increases for a specified number of iterations;  3) testing set used to verify the performance of the network.

The training process can stop using different mechanisms to avoid overtraining. ANN training is commonly done by splitting the input data into training, testing and validation sets. In a training process using validation datasets, if the neural network starts overfitting, even if the training error is decreasing, the error on the validation set will start to increase. When the validation error continues increasing up to a user specified number of iterations, the network training will stop and deliver the results, even if it may not reach the final error goal. The training set is used to update the weights and biases whereas the validation dataset is used to stop training to avoid overfitting. The testing set can also be included in the training process to evaluate the network performance.

ANN training stops when any of these conditions occur (Mathworks 2005):

- The maximum number of epochs (repetitions) is reached;
- The maximum amount of time has been exceeded;
- Performance has been minimized to the error goal;
- The performance gradient falls below mingrad.(minimum performance gradient);
- Validation performance has increased more than max_fail (maximum validation failures) times since the last time it decreased (when using validation).

## 2.2 Sub-pixel Mapping

### 2.2.1 Land cover mapping

Though providing a review on this topic is, somehow, beyond the scope of this thesis, it is worthwhile to point out some of the links between remote sensing and land cover mapping. Land cover is a fundamental variable that supports much scientific research. Developments in satellite remote sensing, GIS and advances in image analysis have extensively improved the availability and accessibility of land cover information (Comber *et al*. 2005). Despite their importance, much of the land cover data currently being used are of inadequate quality (incomplete spatially, out-of-date or inaccurate) and expensive (Atkinson 2005). In addition to the care on the data quality, Comber *et al*. (2005) urged the need for users to understand the wider meaning of the data concepts. The realization of the full potential of remote sensing as a source of land cover data is, however, restricted by factors such as the problem of the presence of mixed pixels, which cannot be appropriately accommodated in conventional image classification techniques (Foody 2000).

Despite the high contribution of remote sensing as the most important source for land cover mapping, according to Townshed (1992), after Foody (2004) it is, in practice, a difficult task to accurately map land cover from remote sensing data. Foody (2004) argues that those studies that have succeeded in accurately mapping land cover have also been based on small test sites often with ideal conditions such as large and relatively homogenous parcels, insignificant topographic variability, etc and using carefully selected and processed imagery.

To address the various limitations of remote sensing for land cover mapping and other areas of applications, so far, researchers introduced various advanced image processing techniques to extract accurate and maximum information. The existence of the mixed pixel problem, for instance, led to the development of sub-pixel classification (Atkinson 2005); whereas the shortcomings of sub-pixel classification in addressing the spatial location of class fractions in a pixel led to the introduction of sub-pixel mapping.

### 2.2.2 The need for sub-pixel mapping

As pointed out in the previous sections, locating class proportions within each pixel is among the limitations of sub-pixel classification. The solution to the mixed pixel problem as suggested by many authors is the soft-classification, which used to predict the proportion of each class within each pixel. However, a soft-classification technique has still limitations since it does not show the exact distribution of these class components within each pixel. Therefore, the resultant prediction still contains uncertainty (Tatem *et al.* 2003). Moreover, Tatem *et al.* (2003) stated that the complexity of the datasets produced from soft classification puts many GIS users off using soft-classification techniques. Foody (2004) stated that map users typically want easy to use map products, like standard hard-classification, but the result of soft-classification may be large and more difficult to interpret and integrate with other datasets. Various sub-pixel mapping techniques are suggested by various authors to overcome such drawbacks. The subject of transforming multi-spectral (multivariate) data into spatial (univariate) data is referred to as sub-pixel mapping (Atkinson 2005). The key problem of sub-pixel mapping is determining the most likely locations of the fractions of each land cover class within the pixel (Verhoeye and De Wulf 2002).

According to Mertens *et al.* (2004), the concept of sub-pixel mapping was first introduced by Atkinson (1997). Although studies on sub-pixel mapping are still very limited, various researchers attempted to introduce different sub-pixel mapping techniques. The following section highlights some of the related works on sub-pixel mapping.

### 2.2.3 Overview of previous studies on sub-pixel mapping

Efforts to develop better techniques to improve spatial representation are one of the research focuses. Sub-pixel classification and sub-pixel mapping are among those research interest areas that are getting greater attention. Sub-pixel mapping has great potential as a step beyond sub-pixel classification (Verhoeye and De Wulf 2002). Possible applications of sub-pixel mapping are expanding and already range from use in raster to vector conversion to refinement of estimates of ground control point location (Mertens *et al.* 2003). According to Tatem *et al.* (2001a), sub-pixel scale studies are based on either direct processing of a raw image or the pre-processed (e.g. soft-classified) images. This review will mainly focus on studies that demonstrated sub-pixel or super resolution mapping based on a soft-classified image since it is the center of the research problem in this thesis.

A limited amount of work has been reported in the literature that considers the spatial distribution of class proportions within and between pixels in order to produce land cover maps at sub-pixel scale (Kasetkasem *et al.* 2005). Various names are being used to this task by various authors that are being used interchangeably in various literatures. Some of these alternative terms being used are sub-pixel mapping (Verhoeye and De Wulf 2002; Tatem *et al.* 2004), super-resolution mapping (Tatem *et al.* 2001a; 2001c; 2002; Kasetkasem *et al.* 2005; Nguyen *et al.* 2005), and sub-pixel sharpening (Mertens *et al.* 2004).

According to Nguyen *et al.* (2005), super-resolution mapping is a set of techniques to obtain a sub-pixel map from land cover proportion images produced by soft-classification. A super-resolution map is a map that is derived at a spatial resolution finer than the size of the pixel of the coarse spatial resolution image being classified (Kasetkasem *et al.* 2005) .

In the following section, a brief description on some of the related previous works will be provided. For the sake of presentation convenience, the reviews are categorized as neural network approaches and optimization approaches (other than neural networks).

#### 2.2.3.1 Neural networks approaches for sub-pixel mapping

Artificial neural networks (ANN) are being widely used in various remote sensing application areas such as hydrology (Islam and Kothari 2000), agriculture (Tatem *et al.* 2003), estimation of air temperatures (Jang *et al.* 2004) land cover mapping (Atkinson *et al.* 1997; Tatem *et al.* 2001a; 2002; Kavzoglu and Mather 2003; Verbeke *et al.* 2004). Some of the applications of ANNs in remote sensing are supervised and unsupervised classification, image segmentation, geometric correction, image compression, model inversion or variable estimation, and multi-source data analysis (Foody 2004).

Despite the potential application of ANNs for remote sensing and promising reports by some researchers on the potential application for sub-pixel mapping, researches in the area of sub-pixel mapping are still very few. As pointed out in the preceding section, a relatively comprehensive study on sub-pixel mapping was made by Tatem *et al.* (2001a; 2001b; 2001c; 2001d; 2002; 2003). Even though Hopfield Neural Network (HNN) application to remote sensing and other pattern recognition issues were demonstrated in several studies, Tatem et al. was the one who introduced HNN for sub-pixel mapping. The authors employed Hopfield neural networks for sub-pixel mapping and super-resolution target identification for different study cases. The second study that used ANN is made by Mertens *et al.* (2003) that used wavelets and artificial neural networks for sub-pixel mapping and sub-pixel sharpening.

Tatem *et al*. used a Hopfield neural network and tried to explore the different approaches and applications by demonstrating each study and extending to a further and advanced research. The Hopfield neural network technique has been used to estimate the location of the class proportions within pixels to produce a land cover target map of sub-pixel accuracy from Landsat-TM images. Tatem *et al.* (2001c), as a continuation of their previous works, used HNN for multiple class land cover mapping at the sub-pixel scale. In this study, the authors showed, from a simulated remote sensing image, an accurate prediction suggesting the potential of HNN to predict land cover at sub-pixel scale from satellite images.

Tatem *et al.* (2002) extended their earlier studies of super-resolution mapping of land cover features larger than a pixel, to sub-pixel land cover pattern prediction at sub-pixel scale features. In their study, a Hopfield neural network was used. Besides their effort to predict features at sub-pixel scale, they incorporated another approach different from their earlier (spatial clustering) approach described in Tatem *et al.* (2001c) that used spatial order and class area proportions as a goal and constraint, respectively. This additional approach, which is an extension to the spatial clustering method, used the semi-variance as an objective (goal) function in the energy minimization (the mathematical formulation is provided in Tatem *et al.* (2002)). The network converges to a minimum of an energy function defined as a goal and several constraints and prior information on the typical spatial arrangement of the particular land cover types is incorporated into the energy function as a semi-variance constraint (Tatem *et al.* 2002). The technique was applied to synthetic and simulated Landsat Thematic Mapper (TM) images and compared to the existing super-resolution land cover map. The applications and advantages of these techniques over those earlier ones is elaborated in Tatem *et al.(* 2002).

A more practical application of the Hopfield neural network was demonstrated in Tatem *et al.* (2003). The earlier studies of Tatem et al. were based on synthetic data and simulated satellite images. In this study, still an extension of the earlier studies stated above, they tried to increase the spatial resolution of agricultural land cover maps from a real Landsat TM image using a reasonably large number of classes.

The strong side of the techniques Tatem et al. employed is that there is no need for a reference image to train the network. In many classifications, the network is trained on a part of the image, but the one proposed by e.g. Tatem *et al.* (2001b) and Verhoeye and De Wulf (2002) did not require any additional information about the problem to be solved i.e. the networks were trained on other data than the source imagery, as the finer scaled version of the source image is not available in real situations (Mertens *et al.* 2004).

A recent study that used a HNN for sub-pixel mapping is described by Nguyen *et al.* (2005). They use the elevation data from a light detection and ranging (LIDAR) instrument as an additional source of information for super-resolution mapping using the HNN to generate a super-resolution map.

Another study that used an ANN for sub-pixel mapping other than the HNN is demonstrated by Mertens *et al.* (2004). This study makes use of wavelets and feedforward backpropagating artificial neural networks for sub-pixel mapping and sub-pixel sharpening through estimation of the detail wavelet coefficients with neural networks.

### 2.2.3.2 Optimization approaches for sub-pixel mapping

Besides ANN, various researchers introduced different techniques of sub-pixel mapping. One of such studies was by Verhoeye and De Wulf (2002). In this study the authors used linear optimization techniques for sub-pixel mapping of Sahelian wetlands using ERS2-ATSR images (Verhoeye and De Wulf 2000) and land cover mapping at sub-pixel level from SPOT-Vegetation images (Verhoeye and De Wulf 2002). The authors attempted to generate a sub-pixel map with various spatial resolutions at 500m, 200m and 100m and demonstrated a corresponding decline in accuracy in the attempt to generate a finer resolution sub-pixel scale map. One of the constraints, as the authors pointed out, that prevented further research into finer resolutions was the computational limitation due to the increased number of variables that the mathematical model contained (Verhoeye and De Wulf 2000). These studies were based on the assumption of spatial dependence and the application of linear optimization techniques using synthetic and real satellite images classified into three classes.

Mertens *et al.* (2003) tried to solve the sub-pixel mapping problem using Genetic algorithms together with the assumption of spatial dependency. The authors performed their study using both synthetic and degraded real images with 2 and 3 classes, respectively.

Aplin and Atkinson (2001) have attempted to transform a soft land cover classification result into sub-pixel scale hard classes based on field boundaries and per field hard classification. However, Tatem *et al.* (2001a) argue on this approach since accurate vector data in most cases will rarely be available.

Recent studies on sub-pixel mapping are the works of Kasetkasem *et al.* (2005) and Atkinson (2005). Kasetkasem *et al.* (2005) used Markov Random Field Models for sub-pixel and super resolution mapping of land cover. Using this technique the authors assert that their results show a significant increase in the accuracy of land cover maps at fine spatial resolution over that obtained from a linear optimization approach suggested by Verhoeye and De Wulf (2002). Atkinson (2005) implemented a simple pixel swapping algorithm as an alternative to the HNN for sub-pixel target mapping from soft classified satellite imagery. For this research, the author used a very small image, 256 by 256 pixels subset images, from IKONOS with only 2 classes (woodland and non woodlands). From this small size of data, it is hard to deduce the applicability of the technique for practical or extensive work (large image plus more classes). The author also states the limitations of this technique.

In most of the studies covered in this literature review, the studies are based on a small number of classes and a small size of synthetic and/or real images. Further studies with a large number of classes and size of image should be conducted to assess the feasibility of the introduced different techniques and move forward to a real application beyond experiments. The writer of this thesis did not find any work that used either of the aforementioned techniques for any practical and real problems solving cases. Despite the reported potential contribution of the technique in reducing land cover map information loss, very few people, mostly developers of the techniques, are being involved/appear in such works. This could be partly due to lack of software tools that implemented the technique.

### 2.2.4 Accuracy assessment of sub-pixel mapping

Accuracy assessment is a crucial task in geo-information analysis and extraction processes. Lillesand and Kiefer (2000) expressed the need for accuracy assessment as ‘’A classification is not complete until its accuracy is assessed’’. There are many alternative measures of accuracy in geo-information processing particularly for classification assessment. The most widely used measures of classification accuracy are derived from a classification confusion or error matrix which shows the predicted and actual class label for a set of pixels sampled from the classification; but they are only appropriate for ‘hard’ classifications and are inappropriate for fuzzy classification (Foody *et al.* 1996). For accuracy assessment of sub-pixel mapping, Tatem *et al.* (2001a; 2001b; 2001c; 2002; 2003) used the area error proportion (AEP), correlation coefficients and closeness and root mean square error (RMSE) techniques. Mertens *et al.* (2003) stated that there is no predefined standard yet for assessing the accuracy of sub-pixel mapping, but the author used the kappa coefficient, the overall accuracy and the root mean square error (RMSE) to evaluate the accuracy of the sub-pixel mapping algorithms. Moreover, Mertens *et al.* (2003) introduced an adjusted kappa coefficient to calculate the kappa coefficient only for the mixed pixels. Kappa coefficient is a measure of difference between the actual agreement between reference data and an automated classifier and the chance agreement between the reference data and random classifier (Lillesand and Kiefer 2000; Kavzoglu and Mather 2003).

# Chapter 3

# MATERIALS AND METHODS

## 3.1 Materials

### 3.1.1 Data

For this thesis work, the latest Dutch land use database (LGN5) is used. The LGN land use data is based on a multi-temporal classification of satellite imagery and integration with ancillary data (Clevers et al. 2004). This data, which is a grid structure with cell size of 25mx25m, was degraded to various pixel/grid sizes in a way that class mixtures of various proportions are created at a coarse resolution. Moreover, it was thematically aggregated into relatively few classes to reduce the number of classes to be used. Training, validation and testing sets for neural network training were extracted randomly from the aggregated LGN data of the entire country.

For the actual sub-pixel mapping implementation, small subset sites from Southeast and Southwest parts of the country were used (see Figure 1.3, Table 3-1). The selection of site 1 is with the intention of including significant proportion of water body that is particularly important to see the performance of the two-class cases. The second site is more diverse and heterogeneous than the first location that has more mixed pixels while aggregated and is helpful to test the performance of the neural network for mapping of these fraction images at sub-pixel level.

**Table 3-1 Area of the two sites used for simulation**

| | Classes | Site 1 | | Site 2 | |
|---|---|---|---|---|---|
| | | Area(pixel) | % | Area(pixel) | % |
| 2 classes | Land | 782,625 | 79.37 | 943,013 | 97.39 |
| | Water | 203,424 | 20.63 | 25,243 | 2.61 |
| 4 classes | Vegetation | 312,317 | 31.67 | 468,706 | 47.53 |
| | Arable land | 222,711 | 22.59 | 349,662 | 35.46 |
| | Built-up | 247,597 | 25.11 | 141,034 | 14.30 |
| | Water | 203,424 | 20.63 | 26,647 | 2.70 |
| 8 classes | Grassland | 267,514 | 27.13 | 317,007 | 32.15 |
| | Arable land | 222,711 | 22.59 | 349,662 | 35.46 |
| | Built-up | 238,578 | 24.20 | 139,905 | 14.19 |
| | Deciduous forest | 31,032 | 3.15 | 54,056 | 5.48 |
| | Coniferous forest | 5,618 | 0.57 | 76,624 | 7.77 |
| | Water | 203,424 | 20.63 | 26,647 | 2.70 |
| | Bare soil | 9,019 | 0.92 | 1,129 | 0.11 |
| | Natural vegetation | 8,153 | 0.83 | 21,019 | 2.13 |

### 3.1.2 Hardware and software

A Pentium IV processor, 1.7 GHz, 512Mb RAM computer was used. The MATLAB neural network Toolbox was used for the neural network implementation. Moreover, the data processing tasks were done mostly using MATLAB version 7.1 software. For the spatial aggregation, a program developed in the department for such purpose was used. The program enables to extract class fraction proportion images that will be used for sub-pixel mapping from the hard class land cover map. ENVI software package version 4.1/2, Arcview and ArcGIS were used for some data preprocessing and presentation activities like evaluating and subset the results of the aggregation, attribute updating and for accuracy assessments.

## 3.2 Methods

In this study, the possibilities and limitations of an artificial neural network (ANN) for sub-pixel mapping particularly for land cover application is assessed. The cases considered to assess capabilities of an ANN are aggregation cell size (75m x 75m, 150m x 150m and 300m x300m) and thematic classes (2, 4 and 8). Initially, the ANN sub-pixel mapping performance was assessed based on only two classes (water and land) having a 75m x 75m grid size. Following the same procedure, the activity was also extended to the rest of the parameter combinations stated above by varying the combination of the spatial aggregation level (degradation) and thematic aggregation (increasing the number of classes).

In this thesis work, there were three main categories of activities: 1) data preparation; 2) implementation that includes selection, designing, training & testing the neural network and running the trained ANN using data of the study areas; and 3) assessing the accuracy of the resulting images. The workflow diagram showing the overall procedure followed in this thesis work is found at the end of this chapter (Figure 3.9). The neural network to be used for the sub-pixel mapping is the feedforward backpropagating neural network (FFBPNN).

### 3.2.1 Preprocessing

The input data for this thesis work were derived by thematically and spatially aggregating the existing LGN data into 3 different thematic classes and three spatial resolution sizes. The input data required for the sub-pixel mapping are the class proportion fraction images derived through spatial aggregation using a special program made for this purpose. To prepare the input fraction images, which are to be treated as soft-classification results, the LGN5 database was thematically aggregated into 2, 4 and 8 thematic classes using the MATLAB software package. Then, these thematically aggregated data were spatially aggregated into three spatial resolution sizes, namely, 75m, 150m, and 300m. The three chosen thematic classes and three spatial resolutions end up in 9 different combinations (Figure 3.1) that are considered as study cases in this thesis work to assess the potentials of neural networks for sub-pixel mapping.

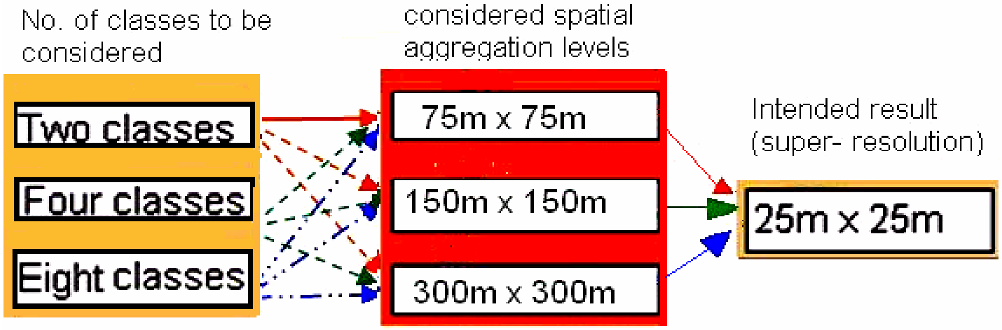

No. of classes to be considered
considered spatial aggregation levels
Two classes
Four classes
Eight classes
75m x 75m
150m x 150m
300m x 300m
Intended result (super- resolution)
25m x 25m

Figure 3.1 Schematic presentation of the 9 study cases

#### 3.2.1.1 Thematic Aggregation

Though the original LGN5 data has 39 classes, in this thesis 9 class aggregated data was used for further aggregation into the 3 levels of thematic aggregations (see Table 3-2). The 25m x 25m and 9-class dataset was thematically aggregated into the following three cases.

Table 3-2 Thematically aggregated classes

| old code | Input class name | 2 classes | | 4 classes | | 8 classes | |
|---|---|---|---|---|---|---|---|
| | | New code | New name | New code | New name | New code | New name |
| 1 | Grassland | 1 | Land | 1 | Vegetation | 1 | Grassland |
| 2 | Arable land | 1 | Land | 2 | Arable land | 2 | Arable land |
| 3 | Greenhouses | 1 | Land | 3 | Built-up | 3 | Built-up |
| 4 | Deciduous forest | 1 | Land | 1 | Vegetation | 4 | Deciduous forest |
| 5 | Coniferous forest | 1 | Land | 1 | Vegetation | 5 | Coniferous forest |
| 6 | Water | 2 | Water | 4 | Water | 6 | Water |
| 7 | Built-up | 1 | Land | 3 | Built-up | 3 | Built-up |
| 8 | Bare soil | 1 | Land | 3 | Built-up | 7 | Bare soil |
| 9 | Natural vegetation | 1 | Land | 1 | Vegetation | 8 | Natural Vegetation |

**Two-classes**: This class contains only land and water. All features other than water are reclassified as land whereas water was kept as it was (see Figure 3.2 and Table 3-3).

Table 3-3 Area of the 2 classes

| Classes | Area(sq.km) |
|---|---|
| Land | 33755.32 |
| Water | 7771.41 |
| Total | 41526.72 |

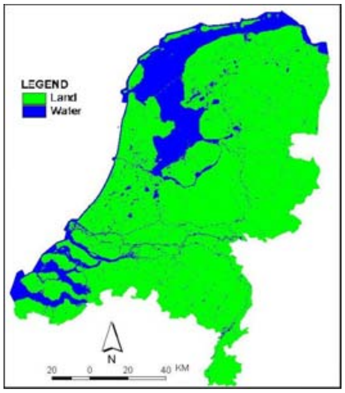


Figure 3.2 LGN data thematically aggregated to 2 classes

**Four classes:** For this case, grassland, deciduous forest, coniferous forest and natural vegetation classes were assigned to vegetation; greenhouses, bare soil and built-up were classified as built-up; water and arable land were unchanged (remain as water and arable land, respectively)(see Figure 3.3). The total area of each class is provided in Table 3-4.

Table 3-4 Area of the 4 classes

| Classes | Area(Sq.km) |
|---|---|
| Vegetation | 18767.43 |
| Arable land | 9594.61 |
| Built-up | 5393.27 |
| Water | 7771.41 |
| Total | 41526.72 |

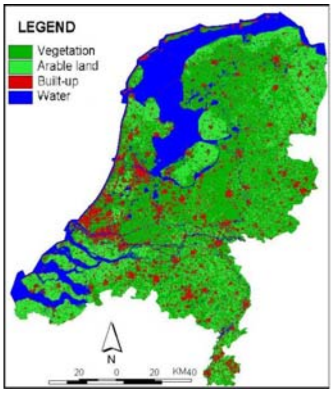


Figure 3.3 LGN data thematically aggregated to 4 classes

**Eight classes:** Only greenhouses and built-up classes are merged. The rest of the classes are kept as the original 9 class data. The new classes in this case are Grassland, Arable land, Built-up, Deciduous forest, Coniferous forest, Water, Bare soil, and Natural vegetation ( Table 3-5 and Figure 3.4).

Table 3-5 Area of the 8 classes

| Classes | Area(Sq.km) |
|---|---|
| Grassland: | 14406.77 |
| Arable land: | 9594.61 |
| Built-up: | 5079.27 |
| Deciduous forest: | 1653.60 |
| Coniferous forest: | 2020.50 |
| Water: | 7771.41 |
| Bare soil: | 314.00 |
| Natural vegetation | 686.56 |
| Total | 41526.72 |

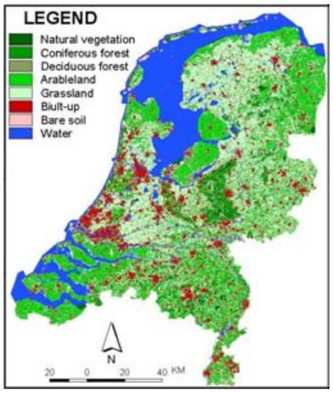


Figure 3.4 LGN data thematically aggregated to 8 classes

#### 3.2.1.2 Spatial aggregation

The spatial aggregation was undertaken on the three thematically aggregated datasets (2, 4 and 8 classes). The spatial aggregation levels chosen are 3 levels having aggregation scale factor ($S_f$) of 3, 6 and 12 with respect to the 25mx25m dataset. The aggregation scale factors 3, 6 and 12 are used to degrade the 25mx25m LGN5 dataset to 75mx75m, 150mx150m and 300mx300m, respectively.

The spatial aggregation was performed using a small program that aggregates based on the majority rule. This program aggregates the original image and generates various layers. The outputs of this program include class fraction layers and the main spatially aggregated image.

*The Class Fraction images*

In the spatial aggregation procedure, besides deriving the aggregated low resolution hard map, class fraction images to be used as soft classification result were derived. Like any soft classification procedure, it enables to derive class proportion images that show the level of class mixture. These class fraction images are produced by degrading the high resolution image (25m) to the low resolution images (75m, 150m, and 300m).

The outputs of the spatial aggregation procedure are fraction image layers, which are equal to the number of classes, and each fraction image layer contains the proportion of a particular land cover class within a pixel. The individual layers contain values ranging from 0 to 1 reflecting the membership to a class. The sum of all class fraction layers of every pixel is 1 except those pixels mixed with the background at the border of the image. The class area proportion fractions of each land cover class within each pixel of the coarse resolution image were computed. Hence, it is this product together with the corresponding thematically aggregated high resolution (25m) data that is used to train the neural network and consequently to reconstruct the 25m resolution image.

### 3.2.2 Neural network design and training

#### 3.2.2.1 Neural network input data preparation and sampling

Samples for the neural network training, testing and validation were taken through a stratified random sampling approach. For stratified sampling, the fraction images were clustered into limited number of classes using kmeans classification. Then, using this image as strata, random sampling within each stratum was applied. The sampling was done simultaneously from the input fraction images and the target image.

*N minus 1*

In all cases, since a pixel is a composition of all classes that sum 1, one layer of the fraction image in the training process was excluded; i.e. one minus the summation of all other classes will be the fraction value of the one excluded. For instance, for the eight-class cases, whichever class fraction comes last, it was excluded and 7 layers were used in the training process with the above assumption. This was done with the intent to minimize the training load that limits the computer time and memory. This procedure was particularly crucial for those cases with a large number of classes and low spatial resolution.

*3x3 window neighbor inputs*

To incorporate the aspects of spatial dependency/order; the neighboring pixels from each of the fraction (the coarse resolution) images was also taken into account in the training process. This has the advantage of introducing texture information into the training procedure, but results in greatly increased training time per-cycle because of the nine-fold increase in input pixels (Paola and Schowengerdt1995). Though not applied in this work, another approach as a compromise for this problem is to use a 3x3 window in one band and only the center pixel in the rest of the bands (Paola and Schowengerdt 1995). The window size considered in this thesis work too is a 3 by 3 pixels (8 neighbors for the central pixel) (see Figure 3.5). Samples mixed with background were excluded from the training process to avoid the possible influence on the network performance. The samples without full (3x3) neighbors were also excluded to maintain consistency with the proportion of class fraction in every sample and to avoid the possible influence on the training performance.

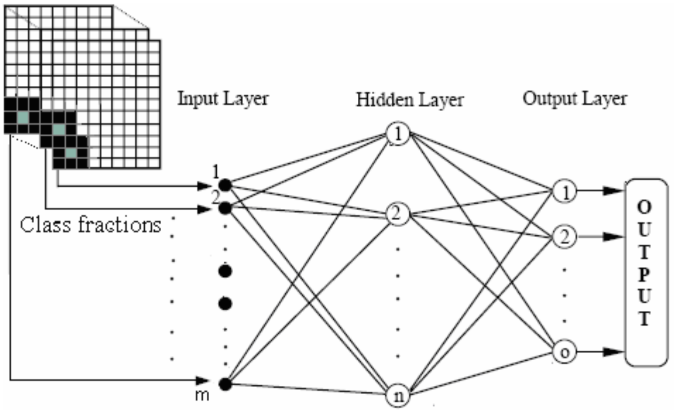


Figure 3.5 A simple three layer backpropagating neural network structure when presented with a 3x3 window (adapted from Paola and Schowengerdt 1995)

To facilitate the sampling process, various functions were used. Using the MATLAB functions, training samples from both the high-resolution (25m) and low resolution images (75m, 150m, or 300m) were extracted from the same location. In this study, a single pixel in the low resolution image is counted as one sample. To elaborate it more, a sample from 75m, 150 and 300m will have 3x3 (9 pixels), 6x6(36) and 12x12(144) pixels, respectively, in the high-resolution (25m) image. In short, for ***n*** number of samples in the low resolution image, we have ***n x scalefactor^2*** pixels in the high-resolution image that correspond to **n** samples (Figure 3.6).

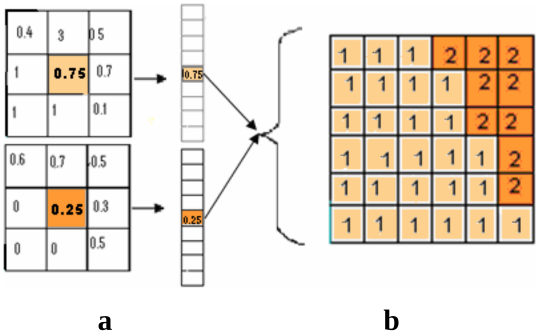


Figure 3.6 Schematic representation of a sample from the 2 class case (the central pixel with its 8 neighbors at 150m pixel size (a) and the corresponding sample of the high resolution 6x6 pixel)(b).

Since the network has to be trained and tested with selected datasets, for which the actual output is known (Arora *et al.* 2004), the network training, testing and validation datasets were extracted from the entire LGN data. The training, testing and validation datasets were extracted from both the fraction (low resolution) image and the target (high resolution) images. The selected samples were again split into 3 groups:

- ½ for training the neural network;
- ¼ for testing the performance of the trained network and
- ¼ for validation and for early stopping the neural network to avoided overfitting.

The sample size is also an important aspect that needs to be determined by the user. There is no clear guideline on the optimum size of samples. But for this thesis work, formula 3.2 was used as a modification of formula 3.1 of Hush (1989) cited in Kavzoglu and Mather(2003).

$$\text{Sample size range} = 30 \text{ to } 60\left(N_i(N_i+1)\right) \qquad \text{....................... 3-1}$$

$$\text{Sample size range} = S_r N_i(N_i+1) \qquad \text{............................................ 3-2}$$

Where: $\mathbf{N}_i$ is the number of input nodes and $S_r$ is a sample range factor.

$S_r$ is used to define a variable sample size by changing the value of $S_r$ for each test case. The recommended range is between 30 and 60, which are used to define the least, and the optimum sample size limits (formula 3.1). But in this work, a bit wider ranges (10 to 80) were employed depending on the complexity of the cases.

Thus, cases with a large number of classes together with a large scale factor will need a significantly large number of samples. But, in this thesis, it was not possible strictly to follow the formula with the recommended range (30 to 60). Since the spatial coverage of cases with large $S_f$ is large, there will be a risk/possibility of overlapping the training and simulation samples. For instance, if an $S_r$ of 60 is used to define the sample size in cases with $S_f$ of 12, about 24% of the simulation data will also be included in the training samples. So, this is also another restricting condition to limit the sample size. Moreover, it will be too much for the training process as such an amount of samples needs large computer memory. Since these cases also need a corresponding large amount of input, hidden and output nodes, limiting the sample size and the number of hidden nodes was indispensable for this work.

No input and target normalization was implemented in most of the tests undertaken in this work, i.e., the fraction inputs and the target class codes were used without adjustment. The reason for this was due to concern that it might disrupt the relationship between the input fractions and the target classes. But, for few test cases, normalization of the input and target to (-1, 1) range was employed (e.g. case 9).

#### 3.2.2.2 Selection and designing of neural network architecture

*Neural network architecture*

In this thesis, a feedforward backpropagating neural network, also called a multi-layer perceptron (MLP), is used. In most cases a three or four layers backpropagating neural network is used. These layers are 1 input layer, 1 or 2 hidden layer(s), and 1 output layer. In some literatures, the input layer is disregarded in network layer counting. The number of nodes in the input layer in this work is dependent on the study cases; i.e. their size depends on the number of classes being used.

The number of input nodes ($N_i$) in the input layer is equal to: 9*(Number of classes-1). Number of classes in this case refers to the number of fraction images. The numbers -1 and 9 in the formula refer to the fraction image to be excluded in the training process and the 3 by 3 neighbor inputs, respectively (see section 3.2.2.1). Therefore, the input layer with 2, 4, and 8 class cases will have 9, 27 and 63 input nodes, respectively.

The number of nodes in the output layer ($N_i$) is equal to $S_f^2$. Where $S_f$ is a scale factor that corresponds to 3, 6 or 12 for the 75m, 150m or 300m resolutions, respectively. Hence, based on the above expression, the number of output layer nodes of the 75m, 150m and 300m resolution input data would have 9, 36 and 144 nodes, respectively.

*Number of hidden layers and nodes*

One of the big challenges in neural network use is determining the optimum number of hidden layers and hidden layer nodes. In most cases, this is defined through trial and error.

But in this case, since such approach is time consuming, formula 3-3 (modified from the formula of Ripley indicated in Table 2-1) was used as a general guide to start with defining the size of the hidden nodes.

$$N_h = N_f\,(N_i + N_o) \qquad \textbf{......................3-3}$$

Where $\mathbf{N}_h$ is the number of hidden nodes, $\mathbf{N}_\mathbf{i}$ is the number of input nodes (the $\mathbf{N}_\mathbf{i}$ is equivalent to the number of elements in the input data vector), $\mathbf{N}_\mathbf{o}$ number of output nodes (corresponds to the number of target pixels which is equal to the $S_f$^2), and $\mathbf{N}_\mathbf{f}$ is a node factor (a fraction used to vary the number of nodes for different cases). The $\mathbf{N}_\mathbf{f}$ is a constant and the recommended $\mathbf{N}_\mathbf{f}$ ranges from ½ to ⅔. But for this case, a range from 0.30 to 0.95 was used based on the complexity of the case. Whenever 2 hidden layer test cases were made, each layer was assigned with a number of nodes half of the $\mathbf{N}_h$. The number of hidden layer nodes is highly varying depending on the complexity of the study cases.

In general, the numbers of hidden layer nodes need to be large as a large number of classes and a large $S_f$ is used. Though this was the intent, it was not possible to use a corresponding large number of hidden nodes due to computational complexity. The large number of inputs and output nodes plus the required large number of hidden nodes and the corresponding connection weights makes the training process unmanageable. Moreover, the cases with large $S_f$ will have large spatial coverage per-sample (more heterogeneous spatial coverage) whereas those with small $S_f$ will have small spatial coverage per sample (less heterogeneous). The more heterogeneous cases, generally, need more hidden nodes to discriminate the different patterns. But, besides the computational limitation, there is still another limiting condition. In most literatures, it is stated that the number of hidden nodes is also determined by the training sample size.

### *Training algorithm and transfer functions*

The MATLAB neural networks toolbox supports several alternative training and learning algorithms. Among these training algorithms, the performance of some algorithms was assessed (though not exhaustive) using the less complex cases. These less complex cases in this context are those with a small number of classes and scale factor that need a corresponding small number of samples and nodes. These training algorithms were Levenberg-Marquardt (TRAINLM), BFGS Quasi-Newton (TRAINBFG), Resilient Backpropagation (TRAINRP), Scaled Conjugate Gradient (TRAINSCG) and One-step Secant (OSS) algorithms. Some of the training algorithms were found not suitable for large and complex cases. Due to the complexity of the different cases, most of the popular algorithms were not convenient to use for this case mainly due to memory requirement and speed of convergence. Due to such computational limitations further investigation on the effect of the different alternative parameters (number of nodes, number of layers, sample size, weight and bias, training algorithms, transfer functions, output formats, etc) was difficult. Considering all these constraints, the Scaled Conjugate Gradient (SCG) backpropagation algorithm was identified to be suitable to use for this particular case study (see section 2.1.3.3).

The performance function used for this thesis work was the mean squared error (MSE) (see section 2.1.3.3). For neurons in the hidden layer(s), a non-linear hyperbolic tangent sigmoid (TANSIG) transfer functions was used. A linear transfer function (PURELIN) was used in the output layer (see also section 2.1.3.3).

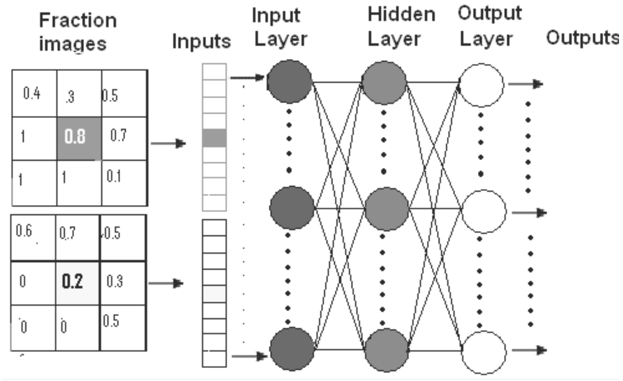


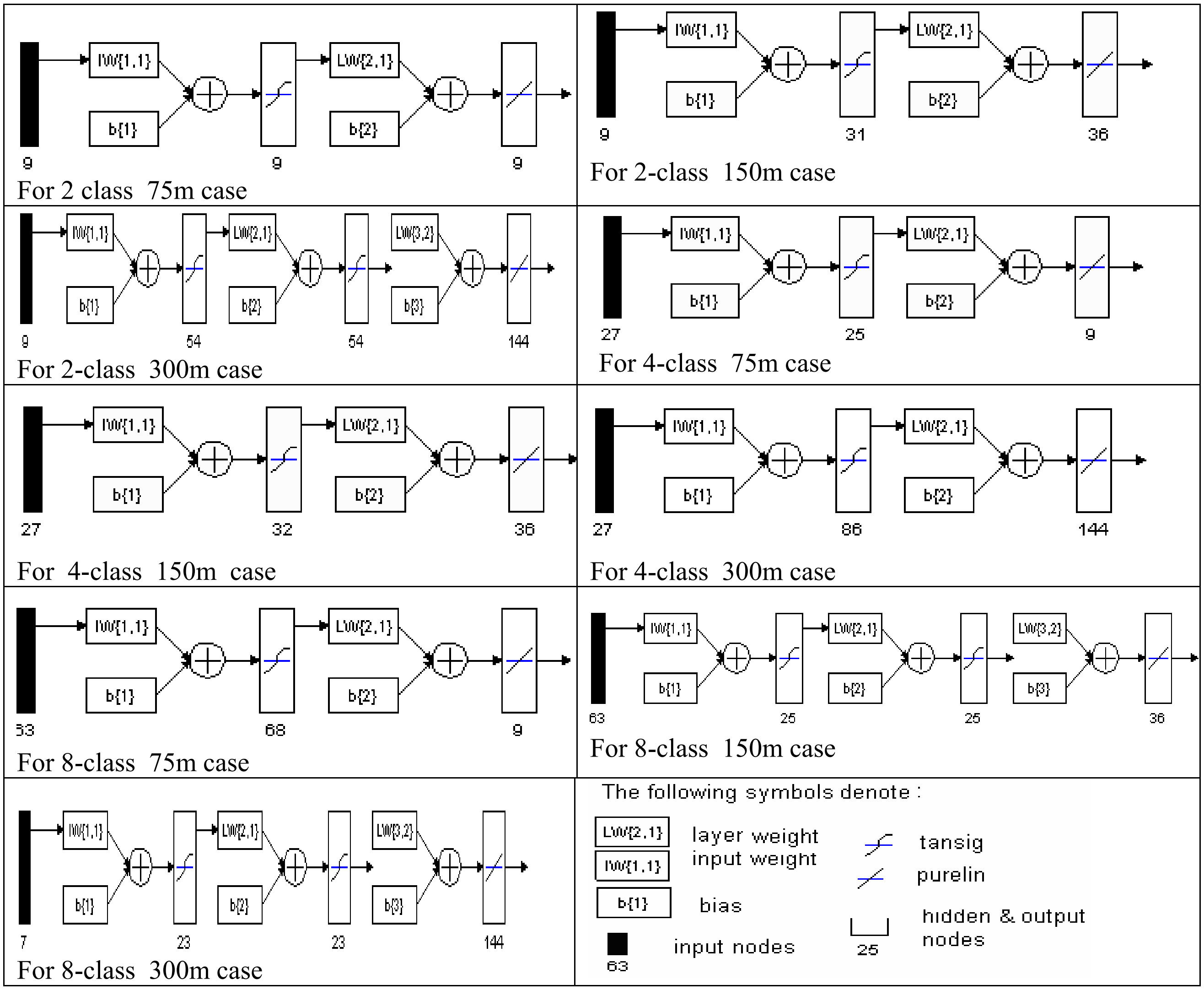
Figure 3.7 Schematic representation of the inputs and the neural network implementation

### 3.2.3 Simulating network response with the new input data

Among the several trial runs, the network architectures indicated in Table 3-6 and Figure 3.8 are those networks performing relatively better. These architectures were selected by assessing the error using the training data and the simulation data. The details will be provided in chapter 4. In all the networks listed in Table 3-6, sigmoid (TANSIG) and linear (PURELIN) transfer functions were used in the hidden and output layers respectively (see section 2.1.3.3). Moreover, the training function used in all chosen networks is the Scaled conjugate gradient decent function (TRAINSCG). Though other training functions also gave comparable results, emphasis was given to SCG and more test cases were run. The reasons why emphasis was on this algorithm have been stated in section 2.1.3.3.

**Table 3-6 List of final selected neural network architecture**

| Case no | Cases | Structure | hidden layers | Sample factor | Node factor |
|---|---|---|---|---|---|
| 1 | 2 classes & 75m resolution | 9-9-9 | 1 | 60 | 0.50 |
| 2 | 2 classes & 150m resolution | 9-31-36 | 1 | 60 | 0.70 |
| 3 | 2 classes & 300m resolution | 9-54-54-144 | 2 | 60 | 0.70 |
| 4 | 4 classes & 75m resolution | 27-25-9 | 1 | 60 | 0.70 |
| 5 | 4 classes & 150m resolution | 27-32-36 | 1 | 30 | 0.50 |
| 6 | 4 classes & 300m resolution | 27-86-144 | 1 | 30 | 0.50 |
| 7 | 8 classes & 75m resolution | 63-68-9 | 1 | 30 | 0.95 |
| 8 | 8 classes & 150m resolution | 63-25-25-36 | 2 | 30 | 0.50 |
| 9 | 8 classes & 300m resolution | 7-23-23-144 | 2 | 10 | 0.30 |



For 2 class 75m case

For 2-class 150m case

For 2-class 300m case

For 4-class 75m case

For 4-class 150m case

For 4-class 300m case

For 8-class 75m case

For 8-class 150m case

For 8-class 300m case

Figure 3.8 The selected neural network architectures

### 3.2.4 Accuracy assessment

For the accuracy assessment, a matching reference dataset was used from LGN5 data with thematic aggregation and compared to the corresponding predicted result. Kappa coefficient, per-class and overall accuracy and area error proportion (AEP) were calculated to assess the difference between the network result and the reference data. Moreover, the performance of the network result was assessed by visually inspecting and comparing the sub-pixel map resulting from the ANN with the original high resolution image. The AEP was calculated using formula 3-4 below (Tatem et al. 2001c).

:

$$\mathbf{AEP_j} = \frac{\mathbf{t_j} - \mathbf{y_j}}{\mathbf{t_j}} \quad \text{..........3-4}$$

Where **j** is the class, $\mathbf{t_j}$ and $\mathbf{y_j}$ are total areas(in pixel) of class **j** in the target and ANN output images, respectively.

Since the mean square error (MSE) shows the training performance on the training dataset but does not show the performance on new data, it was appropriate to assess the performance of the various trained networks on unseen data prior to the selection of the network for the final sub-pixel mapping. To choose the trained neural network with the highest accuracy, an overall accuracy and Kappa coefficient was evaluated for the various trained networks.
To assess if there is effect of some extraneous features, a majority filtering using 3x3 windows was applied to the ANN output. Then overall accuracy and Kappa coefficients were calculated for the majority analysis results.

Since a large proportion of the image is covered with pure pixels, the overall accuracy and kappa coefficient do not exactly show to what extent the mixed pixel problem is being addressed. To evaluate the accuracy of only the mixed pixels, an alternative accuracy assessment approach was adopted. This is done on a per-pixel basis by increasing the number of classes per pixel of the respective spatially aggregated low resolution image. In every successive computation of the mixed pixel accuracies, the less heterogeneous pixels are disregarded. For instance, if the aim is to calculate the accuracy of the neural network output for pixels having more than 3 types of classes, all pixels composed of 1 and 2 classes are excluded from the computation and only pixels composed of 3 and 4 classes (indicated as >=3) are considered for the specific case. Then overall accuracy assessment results were obtained through averaging the per-pixel overall accuracies.

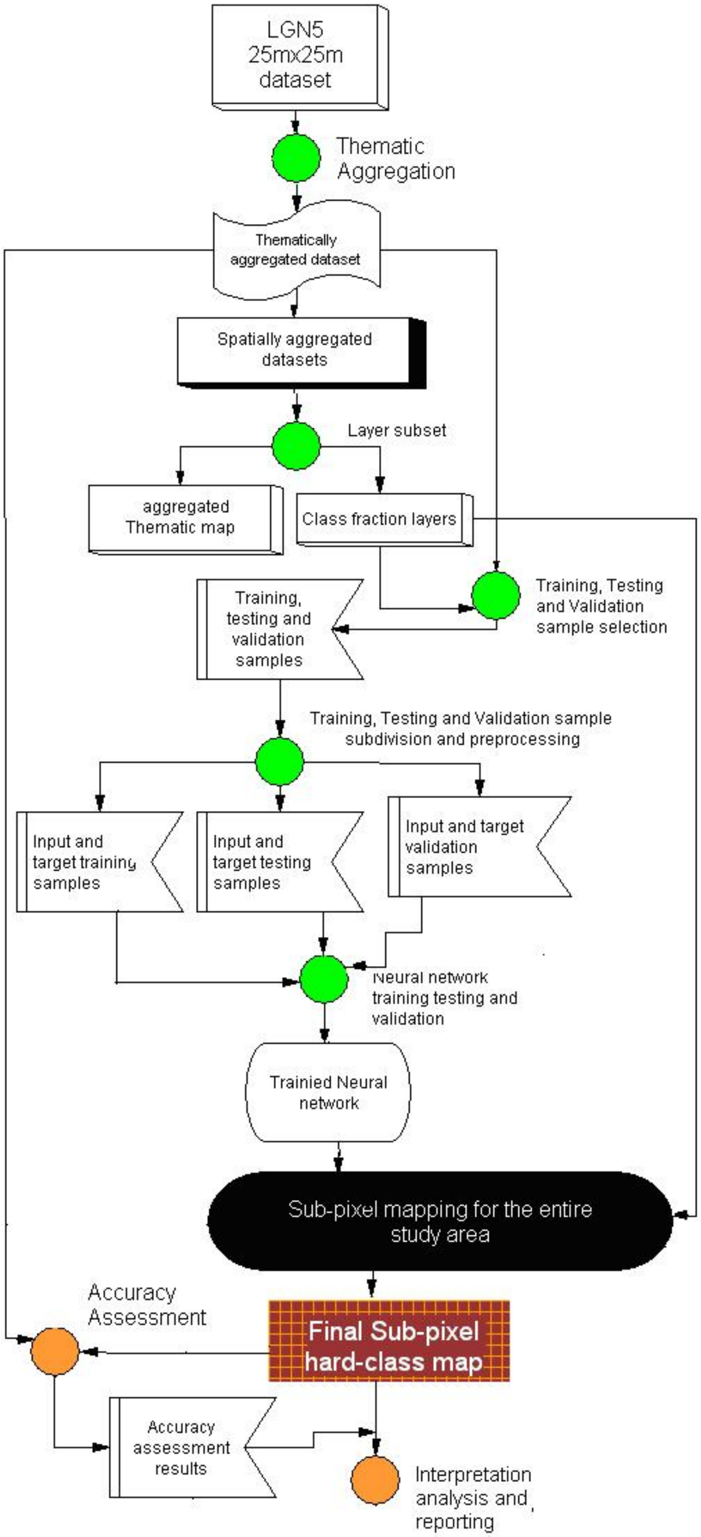


Figure 3.9 Work-flow diagram of the steps followed

# Chapter 4

# RESULTS AND DISCUSSION

## 4.1 Neural network training results

In this thesis, a number of neural network training tests were done by alternating several possible parameter combinations. Among several alternative network parameters, sample size, number of hidden layers and number of hidden layer nodes, training algorithms, weight initialization, input and output encoding/normalization and early stopping strategy were emphasized. Only the best trained networks (listed in Table 4-1) were selected for the final sub-pixel mapping exercise (see Appendix A-2 for more details on the different NN that were tested). In the subsequent discussion of this chapter, the different case studies will be referred using the case numbers indicated in Table 4-1.

Though an early stopping with the validation dataset was employed in most of the training exercises, some tests were executed without validation and testing data. In some test cases, some networks did well by stopping with user-defined epochs (iteration) and/or time (e.g. study case 6 in Table 4-1). In some cases particularly small networks, defining a small goal (the desired MSE error set by the user) resulted in an improved performance. In most test cases, the goal was set to 0.01, but for case 1, an error goal of less than 0.01 was used to see the effect on the performance of the network. For case 1, setting a low error goal (0.005) and letting to stop with the validation data showed a better performance than the 0.01 error goal which was stopping with the maximum error goal. However, larger networks stopped with the validation sets before reaching the goal.

In Table 4-1, the networks that have a small MSE on the training, validation and testing set are those with a small number of classes and a small scale factor ($S_f$) which generally needs a relatively small network. However, the errors increase steadily as the network gets larger. As indicated in the simulation results for each network, this property (error) is also reflected in the accuracy assessment results (kappa coefficient and overall accuracy).

**Table 4-1 Training results of the selected neural networks of the different study cases**

| Case N° | N° classes | $S_f$ | Structure | Epochs | Performance(MSE) Training set | Validation set | Testing set |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 9-9-9 | 216 | 0.007 | 0.007 | 0.007 |
| 2 | 2 | 6 | 9-31-36 | 566 | 0.012 | 0.001 | 0.010 |
| 3 | 2 | 12 | 9-54-54-144 | 1213 | 0.014 | 0.015 | 0.015 |
| 4 | 4 | 3 | 27-25-9 | 772 | 0.114 | 0.115 | 0.116 |
| 5 | 4 | 6 | 27-32-36 | 928 | 0.199 | 0.202 | 0.203 |
| 6* | 4 | 12 | 27-86-144 | 2000 | 0.292 | n/a | n/a |
| 7 | 8 | 3 | 63-68-9 | 610 | 0.473 | 0.472 | 0.475 |
| 8 | 8 | 6 | 63-25-25-36 | 1297 | 0.458 | 0.458 | 0.461 |
| 9** | 8 | 12 | 7-23-23-144 | 933 | 1.334 | n/a | n/a |

*trained with no testing and validation dataset(stopped with epochs)
** trained with no neighbor, no testing and validation dataset(stopped with time)
$S_f$ is scale factor, structure: indicates the N° of input, hidden and output nodes and layers

Except case number 6 and 9, all the training cases were stopped by the validation set.

In general, the two-class cases were the easiest and fastest to train. The overall accuracy levels achieved for almost all tests were above 90%. However, the training and simulation performance of case 3 was relatively poor. For case number 1, the 9-9-9 network structure provided better training and simulation performance than the rest of the network structures. Despite the general rule that the number of hidden nodes should be significantly larger than the number of inputs, the 9-9-9 structure gave a better result. The network was stopped by the validation after running for 216 iterations.

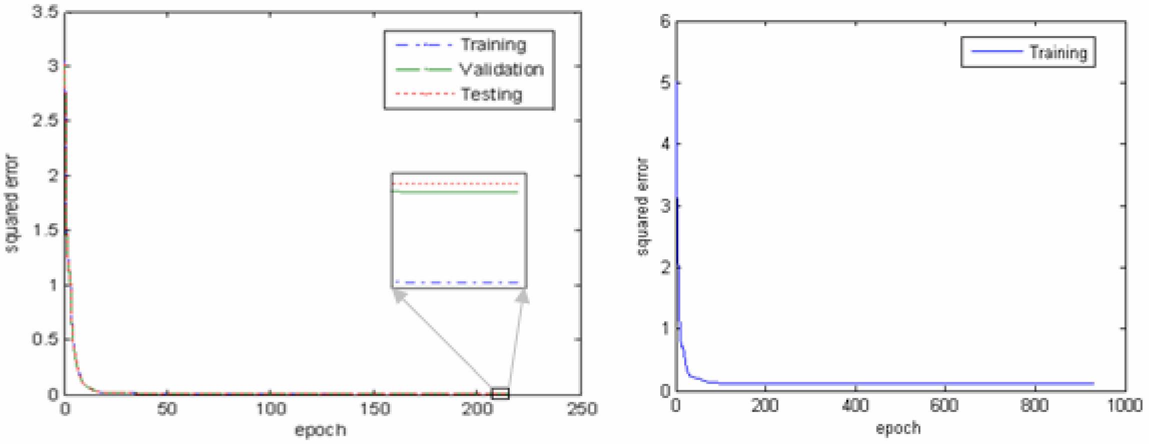
The 4-class cases are intermediate between the 2 and 8-class cases in complexity for training. The number of test cases carried out for the 4-class cases are fewer than for the 2-class cases.

The test cases were few due to the long time it takes to train a single network. Though the goal was set to a MSE of 0.01, none of the 4-class cases reached the error goal. The training process was stopped by the validation set with the exception of case number 6 which was stopped with a user defined maximum epochs (iterations) of 2000.

The 8-class cases were the most difficult cases to train due to the large volume of inputs required. Hence, only two tests per case were executed. Particularly case number 9 was difficult to execute due to computational constraints. It was nearly impossible to train this case using the machine mentioned in section 3.1.2. After a number of trials, it was possible to carry out by minimizing the different inputs (excluding validation and testing dataset, using very small sample range factor ($S_r$) and small node factor ($N_f$)). The following were the values used for network training of case 9:

- Minimum sample size (Sr = 10) and relatively small number of hidden nodes ($N_f$=0.3).
- no validation and testing data were used
- A 3x3 window neighboring input was not applied. Instead, only the central pixel was input to the network.
- initial weight range was set to -0.25- 0.25 ranges
- Normalization of the input and target to -1,1 ranges
- The training process was stopped by setting length of time to train (3 days).

The error graphs of two extreme cases, case number 1 and 9 are shown below to demonstrate the general trend (Figure 4.1). In all cases, the 3 error graphs are very close to each other making it difficult to see the performance of each error graph per iteration. The graphs of those cases not included here also followed the same trend.



a b

**Figure 4.1 Training, validation and testing error graph for case 1 (a) and training error for case 9 (b)**

## 4.2 Simulation results

### 4.2.1 General

The selected networks were applied to the fraction images of the two sites (see section 1.3) which are supposed to have a large proportion of mixed pixels. The simulation was run for all the trained networks for sites 1 and 2. In this section, the potentials and limitation of the feedforward backpropagating neural network for sub-pixel mapping will be demonstrated based on:

- Overall accuracy and kappa coefficient statistics;
- looking into the performance of the network with respect to shape and size and orientation of objects/features;
- Pixel heterogeneity;
- Area heterogeneity (comparing the two sites);
- The effect of a change in scale factor and number of classes;
- The effect of a majority filtering.

From Table 4-2 to Table 4-13 and Appendix A-1 to Appendix A-9**,** various accuracy assessment results of all the cases and sites are provided. The kappa coefficient and overall accuracies in bracket are values obtained after applying majority filtering using 3x3 windows.

To help the comparison of the various results in the consecutive discussions, the area (number of pixels) of each class used in the simulation for the three thematic classes is provided in Table 3-1. Note also that the term target in this thesis refers to the 25mx25m thematically aggregated data whereas ANN output refers to the ANN simulation results having 25mx25m resolution.

### 4.2.2 Results of two-class cases

#### 4.2.2.1 Case 1: Two-classes with 75m spatial resolution

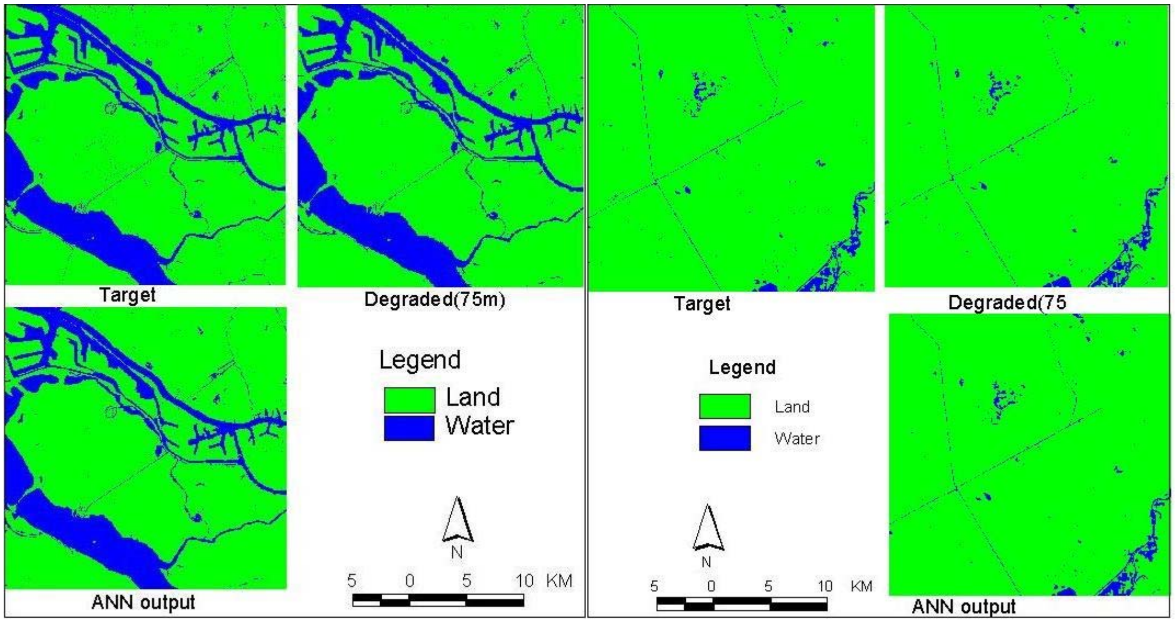


Figure 4.2 Comparison of the two-class target, ANN output and the degraded 75m resolution images of sites 1 (left) and 2 (right)

**Table 4-2 Accuracy assessment of two classes with 75m spatial resolution**

| | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| Class | PROD. ACC.(%) | USER ACC.(%) | AEP | PROD. ACC.% | USER ACC.(%) | AEP |
| Land | 99.12 | 98.56 | -0.0032 | 99.45 | 99.30 | -0.0015 |
| Water | 94.44 | 96.58 | 0.0122 | 74.75 | 79.12 | 0.0553 |
| Kappa coefficient | 0.969(0.971) | | | 0.7625(0.7646 ) | | |
| Overall Accuracy | 98.97%(99.04% ) | | | 98.78%(98.82% ) | | |

The results shown in Figure 4.2 and Table 4-2 are obtained by simulation of the 9-9-9 network structure as indicated in Table 4-1. This network was simulated using two classes with a fraction image at 75m to reconstruct the 25m hard sub-pixel map. In this case, the overall accuracy is comparable for both sites. However, the kappa coefficient of site 2 is considerably lower than that of site 1. The producers and users accuracies of land are high in both cases (over 99%). The Area Error Proportion (AEP) also highlights that a large proportion of water is omitted. The user's accuracy of water in site 1 is also higher than in site 2. The low accuracy level of water contributed to the low kappa coefficient of the site 2. Water is the least accurate in site 2 (74.75%) when compared with site 1 (94.44%). The reason for this difference could be that the proportion of water at site 2 is very small (2.61%) when compared with land (97.39%).

In addition, the shape and size of the water body in site 2 are dominated by narrow and linear/elongated shapes. In most cases, it is observed that the network failed to learn accurately classes with such linear and elongated shapes, and such classes are susceptible to be removed. However, in site 1, the water body is relatively large (see Table 3-1) and dominated by elongated, wide and contiguous areas. Moreover, the orientation of the feature has also influence on the reconstructing of the feature. Though it is difficult to compare every detail from Figure 4.2, it is possible to see and compare the overall patterns generated by the neural network in order to simulate the target. Both images do not have an easily noticeable major difference.

#### 4.2.2.2 Case 2: Two classes with 150m spatial resolution

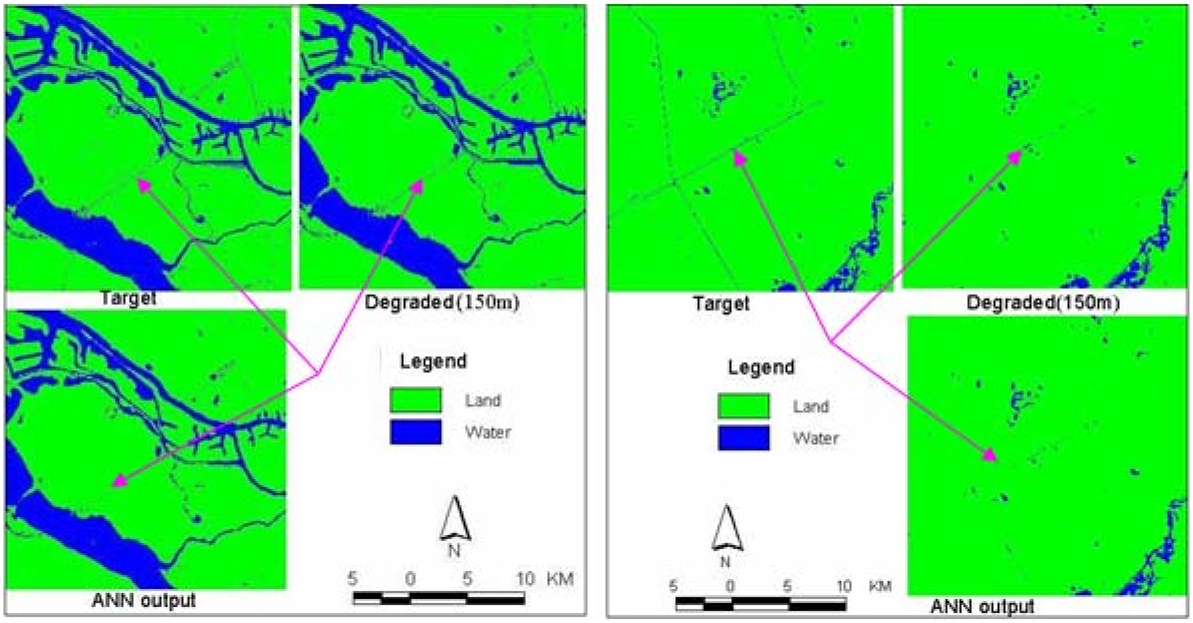


Figure 4.3 Comparison of the two classes target, ANN output and the degraded 150m resolution images of sites 1 (left) and 2 (right)

**Table 4-3 Accuracy assessment of two classes with 150m spatial resolution**

| | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| Class | PROD. ACC. (%) | USER ACC. (%) | AEP | PROD. ACC.(%) | USER ACC. (%) | AEP |
| Land | 99.12 | 98.39 | -0.0074 | 99.56 | 98.97 | -0.0059 |
| Water | 93.74 | 96.51 | 0.0288 | 61.28 | 78.74 | 0.2217 |
| Kappa coefficient | 0.939 (0.940) | | | 0.682 (0.683 ) | | |
| Overall Accuracy | 98.02% (98.05% ) | | | 98.56 %( 98.57%) | | |

Table 4-3 and Figure 4.3 highlight the simulation results of the 9-31-36 network structure indicated in Table 4-1. This network was simulated using two classes with a fraction image at 150m to create the 25m hard sub-pixel map. In this case, too, the overall accuracy gave a comparable result for both sites. Like case number 1, the kappa coefficient of site 2 is significantly lower than that of site 1. The accuracy of water is still the lowest. The accuracy of land is significantly high in both cases which are over 99%. The Area Error Proportion (AEP) also shows that a large proportion of water is omitted (converted to land). It is possible to see some easily noticeable features and compare the overall patterns generated by the neural network to simulate the target. As can be seen in Figure 4.3 for both sites, narrow linear features (for instance indicated with arrows) are missing in the neural network output. Despite the high accuracy obtained, it was unable to learn very tiny and narrow linear features. The overall accuracy and kappa coefficient are still high because of the high proportion of pure pixels. Since the number of classes is small, the images are relatively homogenous with a small proportion of mixed pixels. One important difference that can be observed between the degraded map and the ANN output is that the boundaries in the degraded images are blocky and rough whereas the ANN output boundaries are finer and more or less the same as the target images. Though it is not possible to identify features from the figure, there are situations where the neural network was able to reconstruct features that disappeared in the aggregated map (see also Figure 4.12).

#### 4.2.2.3 Case 3: Two classes with 300m spatial resolution

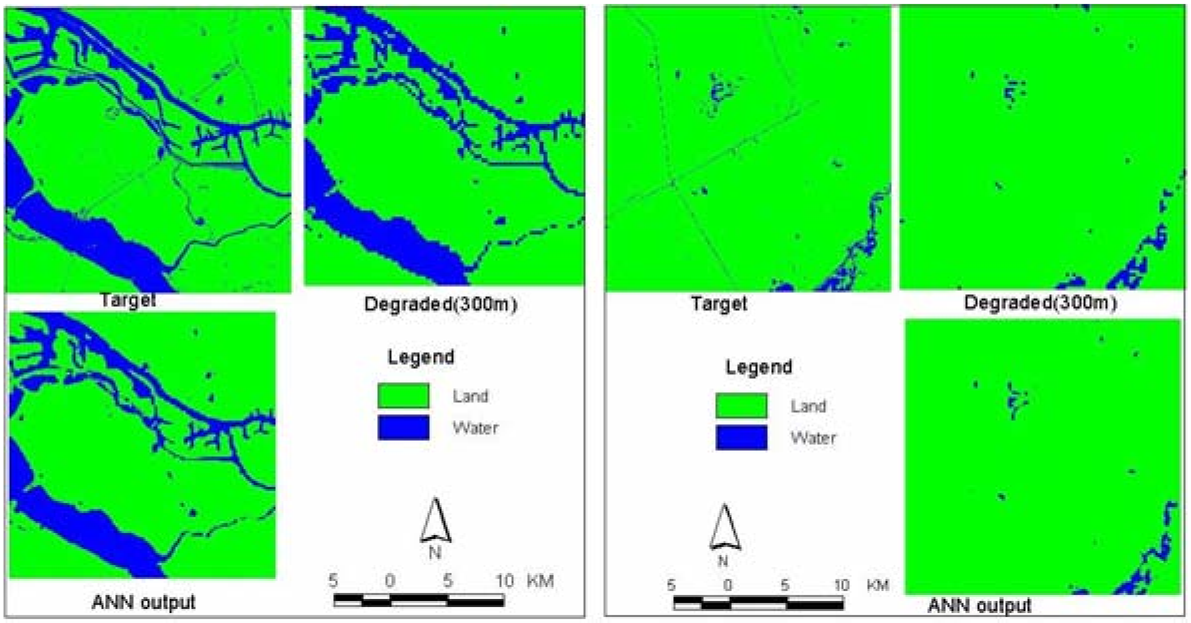


Figure 4.4 Comparison of the two classes target, ANN output and the degraded 300m resolution image of sites 1 (left) and 2 (right)

Table 4-4 Accuracy assessment of two classes with 300m spatial resolution

| | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| Classes | PROD. ACC.(%) | USER ACC.(%) | AEP | PROD. ACC.(%) | USER ACC.(%) | AEP |
| **Land** | 97.87 | 97.05 | -0.0084 | 99.35 | 98.52 | -0.0084 |
| **Water** | 88.61 | 91.64 | 0.0343 | 42.73 | 63.20 | 0.3248 |
| Kappa coefficient | 0.875 (0.882 ) | | | 0.500 (0.502) | | |
| Overall Accuracy | 95.96% (96.18% ) | | | 97.92% (97.93%) | | |

Case number 3 is the least accurate one from the two-class cases. The training performance was also relatively poor (see Table 4-1). This case was a simulation result of a network trained with the 9-54-54-144 network structure. From case number 1 to 3, the effect of the scale factor ($S_f$) can be noticed. The accuracy is decreasing steadily as the $S_f$ is increasing. This case has an $S_f$ of 12 which is very large when compared with case number 1 which has an $S_f$ of 3. In the input-output structure adopted in this study, this large $S_f$ needs 144 output nodes to represent each sub-pixel in the output. This factor aggravated the complexity of the problem when compared with the earlier cases. As can be seen in Table 4-4, the overall accuracy follows a similar trend as the earlier cases, i.e. a relatively low accuracy in water class and an overall low accuracy in site 2 in comparison to site 1. In site 2, a large proportion of water is confused with land (Appendix A-3). Particularly the class accuracy of water in site 2 is low. Moreover, the AEP is large which shows that a large proportion of the water is classified as land. In Figure 4.4, sites are visible that enable to pinpoint differences from the three images and somehow evaluate the potential and limitations of the ANN. In general, site 2 is not suitable for evaluating the two-class cases since the proportion of water is small. Such a disproportional class may lead to a wrong conclusion. Site 2 was meant for representing more diversified and heterogeneous conditions. The following sections (4 class and 8 class cases) will demonstrate how the performance of the network really varies with change in site heterogeneity.

### 4.2.3 Results of four-class cases

#### 4.2.3.1 Case 4: Four classes with 75m spatial resolution

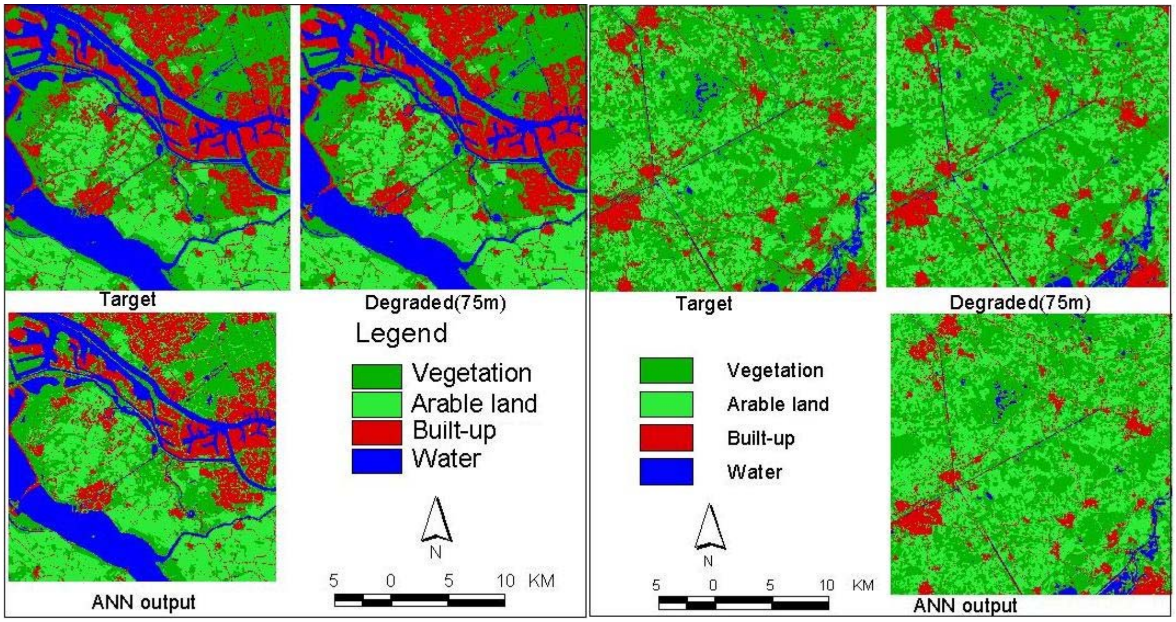


**Figure 4.5 Comparison of the four classes target, ANN output and the degraded 75m resolution images of sites 1 (left) and 2 (right)**

**Table 4-5 Accuracy assessment of four classes with 75m spatial resolution**

| | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| Class | PROD. ACC.(%) | USER ACC.(%) | AEP | PROD. ACC.(%) | USER ACC.(%) | AEP |
| Vegetation | 85.22 | 97.44 | 0.125 | 83.78 | 90.08 | 0.070 |
| Arable land | 97.32 | 74.15 | -0.312 | 89.21 | 74.65 | -0.1950 |
| Built-up | 83.36 | 90.52 | 0.079 | 60.80 | 76.43 | 0.205 |
| Water | 93.64 | 98.95 | 0.054 | 64.89 | 86.50 | 0.251 |
| Kappa coefficient | 0.856 (0.870) | | | 0.710 (0.720) | | |
| Overall Accuracy | 89.22% (90.26% ) | | | 81.91% ( 82.61% ) | | |

Table 4-5 and Figure 4.5 show the ANN output and the accuracy assessment results of case 4 after simulation with a network trained in a 27-25-9 network structure (see Table 4-1). From case number 1 to 3, the effect of the scale factor ($S_f$) is noticed. In this and the following sections, in addition to the $S_f$, it is possible to notice the effect of the number of classes. The overall accuracy dropped below 90%. But the kappa coefficient is still good, particularly the kappa coefficient of site 2 in case 4 is better than that of case 2 and 3. In this case, in addition to water, built-up was the least accurate. In both sites, the class arable land is the most accurate in terms of producer's accuracy. However, in site 2, still the AEP for water is large which indicates that a large proportion of it is converted into other classes (built-up) (see Appendix A-4.1 and A-4.2). Despite the high class accuracy of arable land in both sites, the negative AEP evidences that more classes are transformed to this class. This class is largely confused with vegetation and built-up classes.

#### 4.2.3.2 Case 5: Four classes with 150m spatial resolution

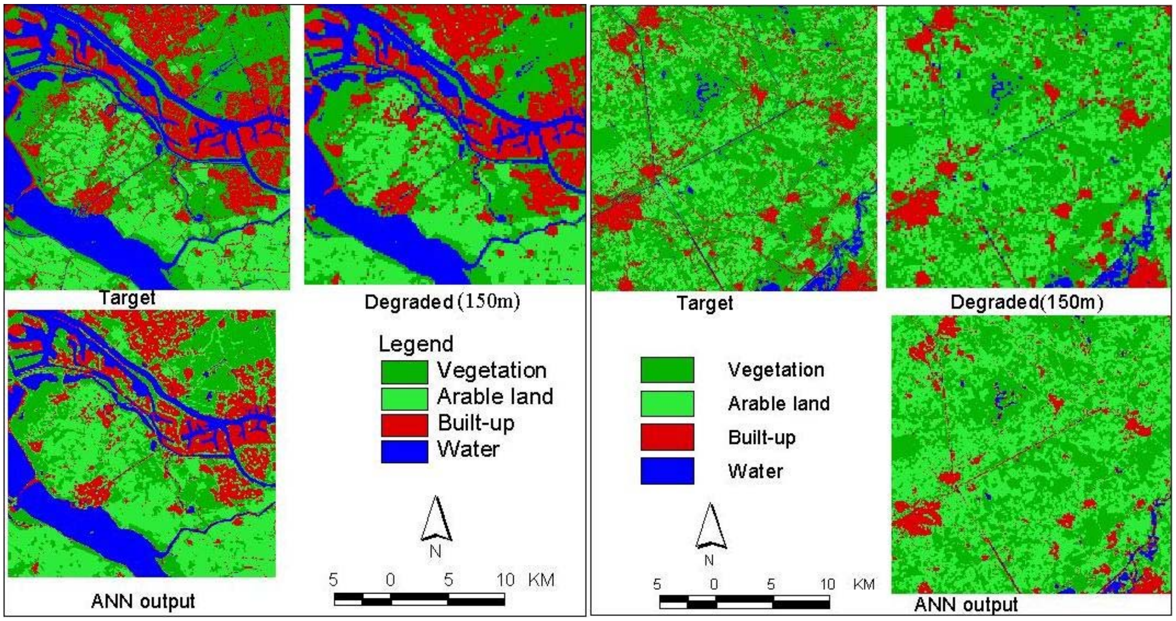


**Figure 4.6 Comparison of the four classes target, ANN output and the degraded 150m resolution images of sites 1(left) and 2(right)**

**Table 4-6 Accuracy assessment of four classes with 150m spatial resolution**

| | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| **Class** | PROD. ACC.(%) | USER ACC.(%) | AEP | PROD. ACC.(%) | USER ACC.(%) | AEP |
| Vegetation | 74.07 | 94.39 | 0.2153 | 75.64 | 86.54 | 0.1259 |
| Arable land | 96.48 | 63.07 | -0.5297 | 87.16 | 64.76 | -0.3458 |
| Built-up | 73.45 | 84.29 | 0.1286 | 47.53 | 73.43 | 0.3528 |
| Water | 88.81 | 98.31 | 0.0990 | 44.54 | 85.98 | 0.4833 |
| Kappa coefficient | 0.762 (0.763) | | | 0.594 (0.596) | | |
| Overall Accuracy | 82.03% (82.23% ) | | | 74.92% (75.04% ) | | |

Case number 5 is a case trained with the network structure of 27-32-36 (see Table 4-1). The decrease in accuracy is getting pronounced in this case. Both the statistical and visual comparison from Table 4-6 and Figure 4.6 showed the progressive decline in the accuracy. The general trend is the same as for case 4 above, except a difference in magnitude (i.e. low overall accuracy and kappa coefficient). The class confusion, for instance, doubled in comparison to case 4 in all classes (See also the confusion matrix in Appendix A-5.1 and A-5.2). The exaggerated class confusion with the nearest class code is also apparent in this case.

#### 4.2.3.3 Case 6: Four classes with 300m spatial resolution

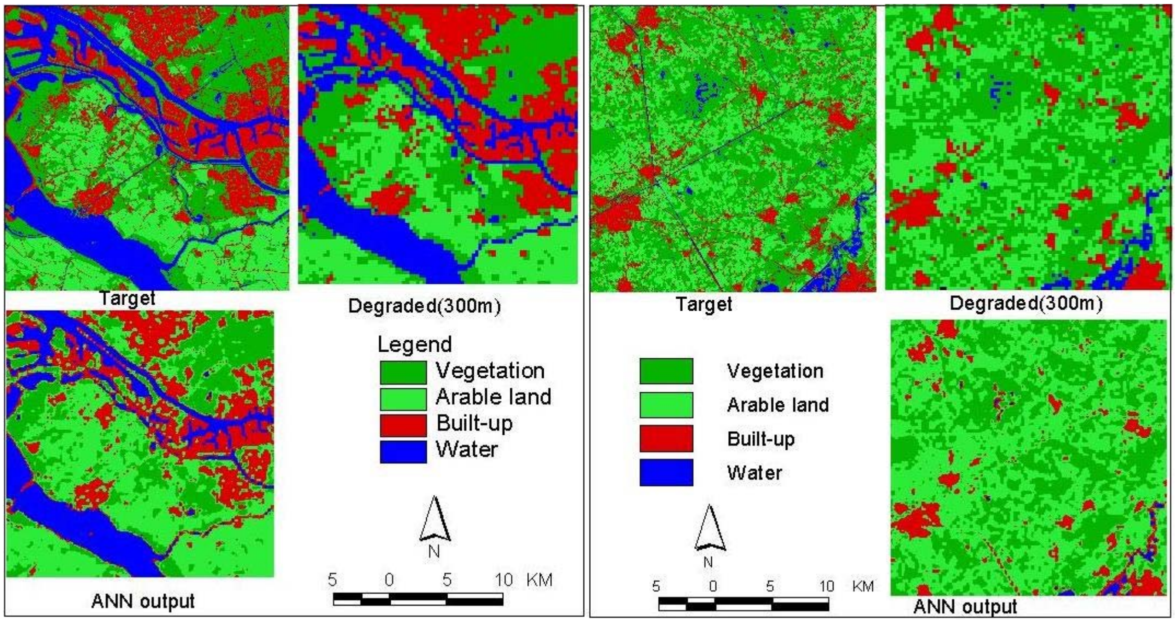


**Figure 4.7 Comparison of the four classes target, ANN output and the degraded 300m resolution images of sites 1(left) and 2(right)**

**Table 4-7 Accuracy assessment of four classes with 300m spatial resolution**

| | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| Class | PROD. ACC. (%) | USER ACC. (%) | AEP | PROD. ACC. (%) | USER ACC. (%) | AEP |
| Vegetation | 60.07 | 90.24 | 0.3343 | 59.34 | 76.18 | 0.2211 |
| Arable land | 94.17 | 53.88 | -0.7478 | 79.10 | 51.98 | -0.5218 |
| Built-up | 66.09 | 72.85 | 0.0929 | 35.86 | 63.09 | 0.4316 |
| Water | 78.72 | 96.57 | 0.1884 | 20.67 | 77.36 | 0.7334 |
| Kappa coefficient | 0.643 (0.644) | | | 0.3866 (0.3876) | | |
| Overall Accuracy | 73.08% (73.16% ) | | | 62.04%(62.11% ) | | |

The network architecture selected for case number 6 is the 27-86-144 network structure. The number of hidden nodes in this case is larger when compared to the earlier cases. Since this case has a large scale factor like that of case number 3, it was more complex to reconstruct the fraction images of the 4 classes to the 144 sub-pixels per pixel of the low resolution image. The accuracy considerably dropped when compared with case 4. Table 4-7, Appendix A-6.1 and A-6.2 show the accuracy analysis results of case 6. This case also shows that the scale factor is decisive as factor in the obtained performance. In this case, larger and intact features are easily recognizable; mostly linear and small size features are not reconstructed.

## 4.2.4 Results of eight-class cases

### 4.2.4.1 Case 7: Eight classes with 75m spatial resolution

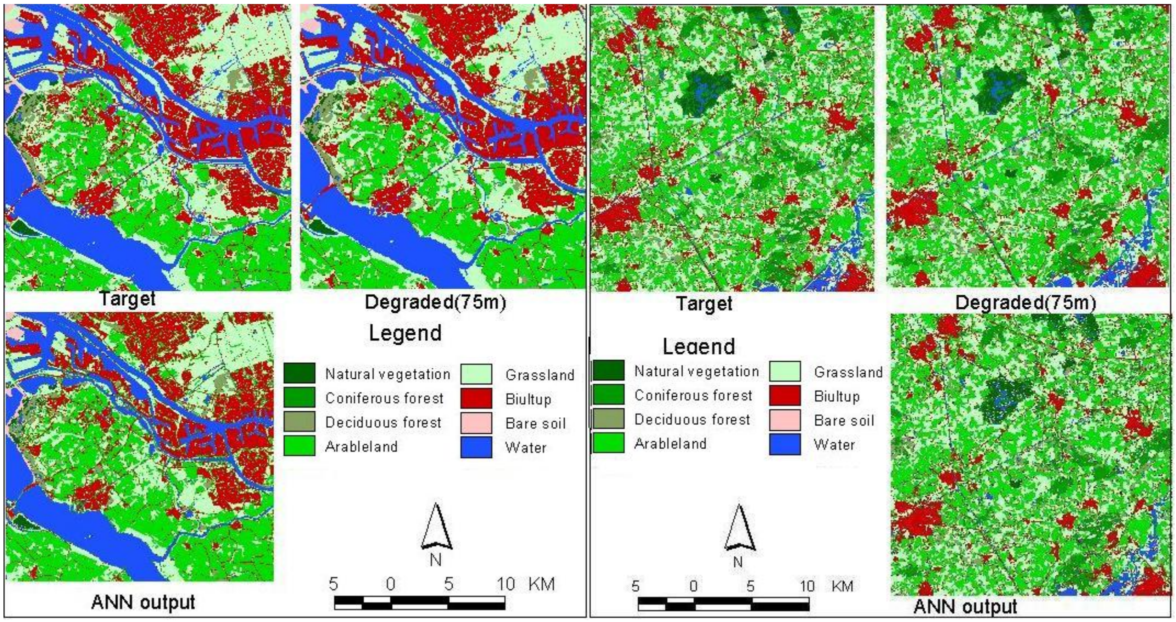


**Figure 4.8 Comparison of the eight classes target, ANN output and the degraded 75m resolution images of sites 1(left) and 2(right)**

**Table 4-8 Accuracy assessment of eight classes with 75m spatial resolution**

| Class | Site 1 | | | Site 2 | | |
|---|---|---|---|---|---|---|
| | PROD. ACC. (%) | USER ACC. (%) | AEP | PROD. ACC. (%) | USER ACC. (%) | AEP |
| Grassland | 82.02 | 97.23 | 0.2853 | 76.42 | 80.77 | 0.0588 |
| Arable land | 97.01 | 76.68 | -0.4467 | 83.54 | 82.91 | -0.0030 |
| Built-up | 83.40 | 90.58 | 0.0987 | 62.91 | 70.65 | 0.1177 |
| Deciduous forest | 77.27 | 63.40 | -0.4194 | 60.64 | 60.36 | 0.0072 |
| Coniferous forest | 27.13 | 10.93 | -4.0976 | 76.11 | 78.12 | 0.0318 |
| Water | 91.44 | 99.08 | 0.1800 | 60.19 | 65.18 | 0.0886 |
| Bare soil | 86.82 | 66.76 | -0.3888 | 23.24 | 1.58 | -13.034 |
| Natural vegetation | 68.70 | 91.91 | 0.4758 | 79.27 | 52.17 | -0.4540 |
| Kappa coefficient | 0.835 (0.862) | | | 0.675( 0.723) | | |
| Overall Accuracy | 87.16% (89.3% ) | | | 75.72% (79.50% ) | | |

The results in Table 4-8 and Figure 4.8 are obtained by simulation of the 63-68-9 network structure indicated in Table 4-1. In this case, despite the large MSE of 0.473, when compared with case 6 with a MSE of 0.292 and the large number of classes, it performed better than that of case 6. This also signifies the contribution of the scale factor in limiting the performance. From this case, one more thing is noticed, i.e. MSE may not necessarily guarantee the network performance for new data. In fact, as $S_f$ gets larger and larger, it is obvious that the possibility of creating a more mixed pixel will increase. Since the major source of error is from those mixed pixels, it is not a surprise to get a lower accuracy for cases with a larger $S_f$. Although the original area of bare soil is insignificant (0.11%) (Table 3-1), a large proportion of deciduous forest and water was transformed to bare soil (see Appendix A-7.1 and A-7.2) and that is why the AEP is so large. This occurred in all 8-class cases.

#### 4.2.4.2 Case 8: Eight classes with 150m spatial resolution

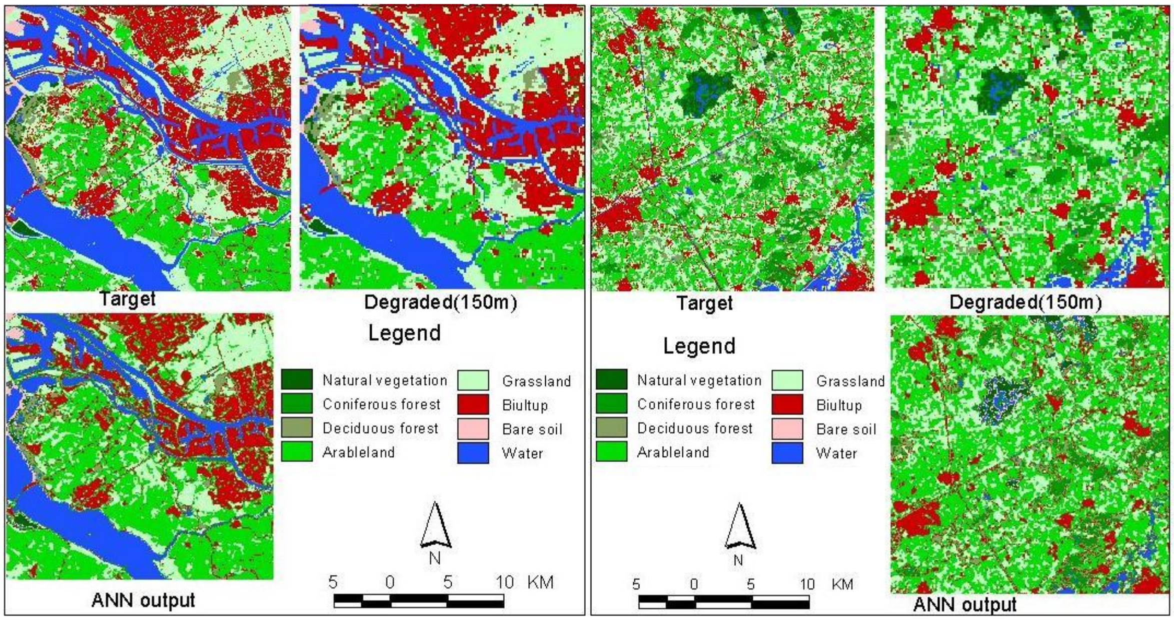


**Figure 4.9 Comparison of the eight classes target, ANN output and the degraded 150m resolution images of sites 1(left) and 2(right)**

**Table 4-9 Accuracy assessment of eight classes with 150m spatial resolution**

| | Site 1 | | | Site 2 | | |
|---|---|---|---|---|---|---|
| Class | PROD. ACC. (%) | USER ACC. (%) | AEP | PROD. ACC. (%) | USER ACC. (%) | AEP |
| Grassland | 68.34 | 93.54 | 0.2694 | 61.10 | 81.28 | 0.2483 |
| Arable land | 95.26 | 65.62 | -0.4518 | 85.05 | 64.60 | -0.3167 |
| Built-up | 72.85 | 82.50 | 0.1169 | 49.04 | 52.99 | 0.0745 |
| Deciduous forest | 57.05 | 42.92 | -0.3292 | 40.90 | 38.42 | -0.0644 |
| Coniferous forest | 8.30 | 1.82 | -3.5515 | 57.84 | 73.43 | 0.2123 |
| Water | 82.67 | 97.93 | 0.1557 | 40.29 | 60.36 | 0.3324 |
| Bare soil | 73.45 | 54.77 | -0.3404 | 5.66 | 0.78 | -6.2255 |
| Natural vegetation | 49.37 | 82.64 | 0.4072 | 60.74 | 92.21 | 0.3447 |
| Kappa coefficient | 0.715 (0.724 ) | | | 0.532(0.537 ) | | |
| Overall Accuracy | 77.68% (78.40%) | | | 65.92% (66.29% ) | | |

In this case, large proportions of the classes are misclassified. As it can be observed from Table 4-9, there is a large area error proportion in both sites. From the table, it is possible to observe also the nature of the AEP. The signs (+/-) of the AEP per class are the same in both sites. In both site, grassland, built-up, water and natural vegetation show a decrease in area, while arable land, deciduous forest and bare soil show an increase in area (see Appendix A-8.1 and A-8.2). This network showed similar characteristics per class in both sites even though there is a variation in magnitude of the errors. This shows how the network behaves with respect to the land cover pattern. Due to the large proportion of area error, the overall accuracy and kappa coefficient are very low. In the 4-class and 8-class cases, the area error in water and built-up is high (high omission), whereas arable land has the highest negative values evidencing the high commission. The land cover pattern of the arable land is relatively intact; this might be the one that contributed the highest class accuracy.

### 4.2.4.3 Case 9: Eight classes with 300m spatial resolution

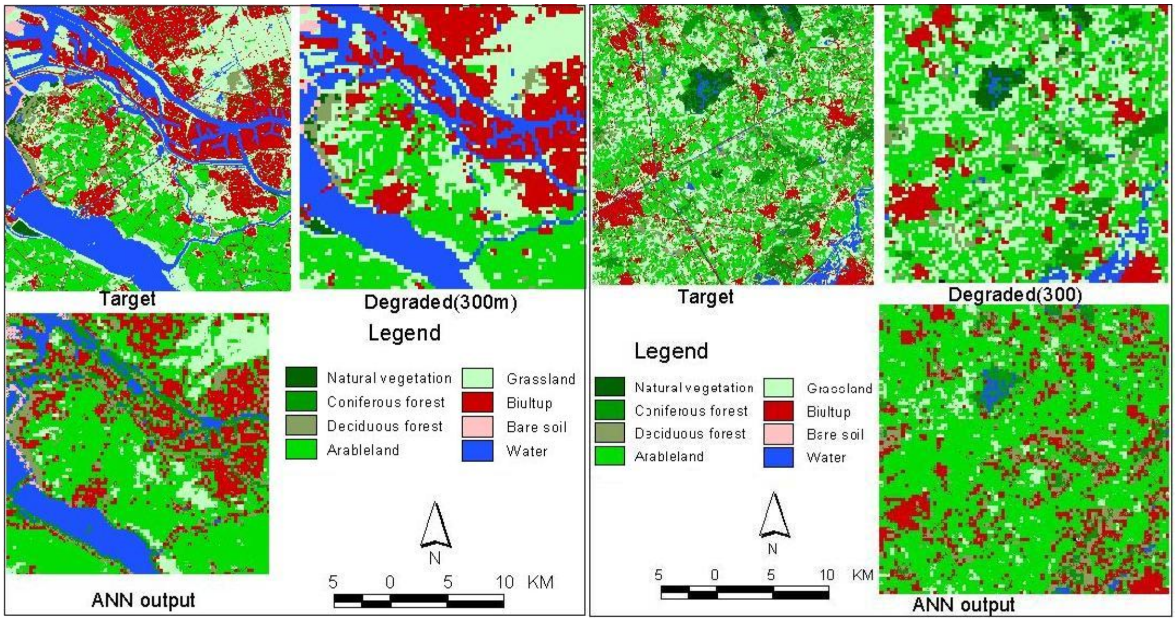


**Figure 4.10 Comparison of the eight classes target, ANN output and the degraded 300m resolution images of sites 1(left) and 2(right)**

**Table 4-10 Accuracy assessment of eight classes with 300m spatial resolution**

| **Class** | **Site 1** | | | **Site 2** | | |
|---|---|---|---|---|---|---|
| | PROD. ACC. (%) | USER ACC. (%) | AEP | PROD ACC. (%) | USER ACC. (%) | AEP |
| Grassland | 28.14 | 79.00 | 0.6438 | 9.87 | 50.37 | 0.804 |
| Arable land | 89.53 | 48.17 | -0.8588 | 78.32 | 43.30 | -0.809 |
| Built-up | 46.97 | 55.97 | 0.1608 | 28.14 | 22.03 | -0.278 |
| Deciduous forest | 54.05 | 19.48 | -1.7750 | 22.25 | 18.06 | -0.232 |
| Coniferous forest | 2.43 | 0.19 | -11.8568 | 19.33 | 40.95 | 0.528 |
| Water | 49.94 | 91.58 | 0.4547 | 12.95 | 36.70 | 0.648 |
| Bare soil | 47.43 | 42.99 | 0.1033 | 0.00 | 0.00 | 0.890 |
| Natural vegetation | 0.54 | 10.83 | 0.9506 | 0.00 | 0.00 | 0.997 |
| Kappa coefficient | 0.398 (0.399) | | | 0.138 (0.139 ) | | |
| Overall Accuracy | 51.56%(51.69% ) | | | 38.05%(38.18% ) | | |

This case was simulated using the 7-23-23-144 network structure indicated in Table 4-1. Though it was trained with the parameters specified in section 4.1, it was not possible to make an extra test experiment by altering the different factors since it is computationally demanding. So, the results shown in Table 4-10, Figure 4.10, Appendix A-9.1 and A-9.2 are presented here only to demonstrate the performance of the network using the settings described earlier. The results also demonstrated the effect of this limitation (the minimum network parameters). But, it should not be necessarily considered that training large networks is impossible. This could be resolved using a high capacity computer and/or another data input structure.

### 4.2.5 Accuracy assessment only for mixed pixels

The results of the mixed pixel accuracy are presented in Table 4-11 to Table 4-13 and Figure 4.11. The overall accuracy assessment results were obtained through averaging the per-pixel overall accuracies having above the specified number of classes per pixel. In all cases, the overall accuracy declined as the number of classes per pixel increased. All followed nearly a similar pattern with the exception of case 9. Case number 9 showed an odd performance for pixels with >=7 classes. Moreover, the network for case number 7 in site 2 was not able to learn the corresponding targets for pixels having >=7 classes. The reason for this may be due to the area of class 7 and 8 which are only 0.11% and 2.13% in site 2 (Table 3-1). Since the pixel size is relatively small (3x3), the probability of getting a pixel with 7 and/or 8 classes in the 9 sub-pixels is very low. This means the probability of getting >=7 classes in a pixel will be very low. Another reason that the networks for case 8 and 9, particularly case 8 detected more (>=7) classes, could be attributed to the number of hidden layers. The network for case 7 has 1 hidden layer whereas cases 8 and 9 are 2 hidden layer networks (see Figure 3.8 and Table 3-6).

**Table 4-11 Change in overall accuracy with change in pixel heterogeneity for the two sites and 2-class cases**

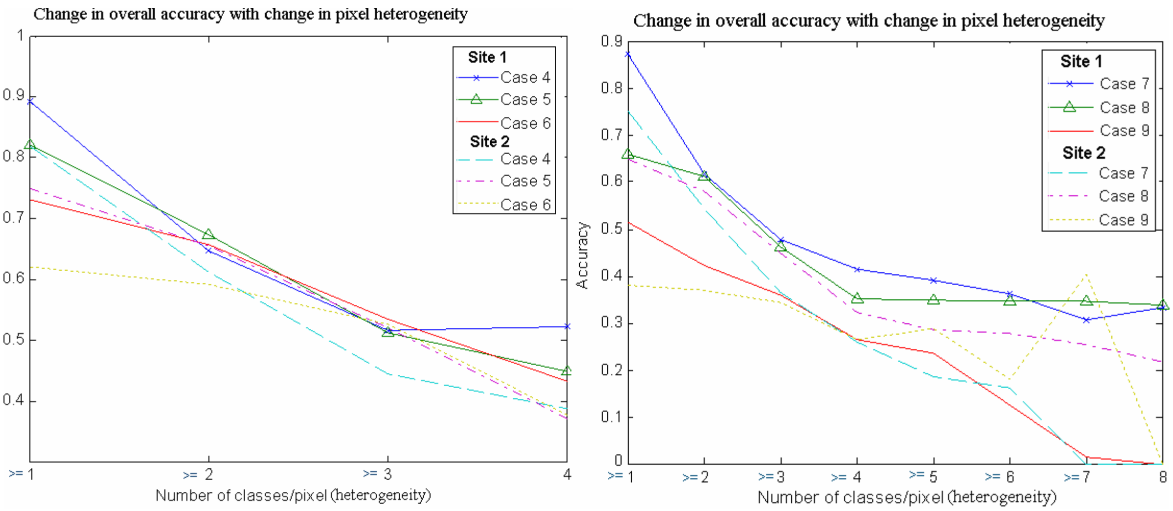
| | Number of classes per low resolution pixel(site 1) | | Number of classes per low resolution pixel(site 2) | |
|---|---|---|---|---|
| | >=1 | 2 | >=1 | 2 |
| Case 1 | 0.990 | 0.845 | 0.988 | 0.661 |
| Case 2 | 0.980 | 0.864 | 0.986 | 0.786 |
| Case 3 | 0.959 | 0.856 | 0.873 | 0.832 |

**Table 4-12 Change in overall accuracy with change in pixel heterogeneity for the two sites and 4-class cases**

| 4 class cases | Number of classes per low resolution pixel(site 1) | | | | Number of classes per low resolution pixel(site 2) | | | |
|---|---|---|---|---|---|---|---|---|
| | >=1 | >=2 | >=3 | 4 | >=1 | >=2 | >=3 | 4 |
| case 4 | 0.8921 | 0.6463 | 0.5154 | 0.5225 | 0.8180 | 0.6121 | 0.4440 | 0.3872 |
| Case 5 | 0.8199 | 0.6732 | 0.5128 | 0.4487 | 0.7491 | 0.6546 | 0.5197 | 0.3708 |
| Case 6 | 0.7301 | 0.6574 | 0.5344 | 0.4327 | 0.6204 | 0.5909 | 0.5253 | 0.3759 |

**Table 4-13 Change in overall accuracy of the ANN output with change in number of classes per pixel (pixel heterogeneity) in the two sites for 8-class cases**

| Sites | 8 class cases | Number of classes per low resolution pixel | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | >=1 | >=2 | >=3 | >=4 | >=5 | >=6 | >=7 | 8 |
| 1 | case 7 | 0.8714 | 0.6173 | 0.4773 | 0.4149 | 0.3895 | 0.3620 | 0.3056 | 0.3333 |
| | Case 8 | 0.6591 | 0.6129 | 0.4608 | 0.3516 | 0.3489 | 0.3448 | 0.3454 | 0.3377 |
| | Case 9 | 0.5151 | 0.4212 | 0.3579 | 0.2652 | 0.2353 | 0.1241 | 0.0139 | 0.000 |
| 2 | case 7 | 0.7520 | 0.5428 | 0.3633 | 0.2605 | 0.1856 | 0.1628 | 0.000 | 0.000 |
| | Case 8 | 0.6491 | 0.5804 | 0.4494 | 0.3231 | 0.2868 | 0.2787 | 0.2536 | 0.2163 |
| | Case 9 | 0.3805 | 0.3705 | 0.3442 | 0.2640 | 0.2885 | 0.1808 | 0.4039 | 0.0000 |



**a** **b**

Figure 4.11 Change in overall accuracy of the ANN output with change in number of classes per pixel (pixel heterogeneity) in the two sites for 4-class cases (a) and for 8-class cases (b)

### 4.2.6 Shape and size effect

This section is intended to show the performance of the network with respect to the nature of various shapes and sizes. The images in Figure 4.12, Table 4-14 and Table 4-15 are a magnified view of a certain location from study site 1. Figure 4.12 illustrates the performance of the neural network with regard to various shapes and sizes. Visual comparison of the 75m aggregated image, the neural network output and the target image revealed that the neural network created an intermediate result between the aggregated and the target image with good improvement of the blocky structure that is created during the aggregation. The following results also show the capability and limitation of the network in predicting such features. In this demonstration, we can see situations where features disappeared in the aggregated image and in some cases are recovered in the neural network output; but with some distortion of shape. It also demonstrates situations where features vanished in both the ANN output and the aggregated map (indicated in circle mark). Below are some shape comparisons with the three maps (Aggregated, ANN output and the target images).

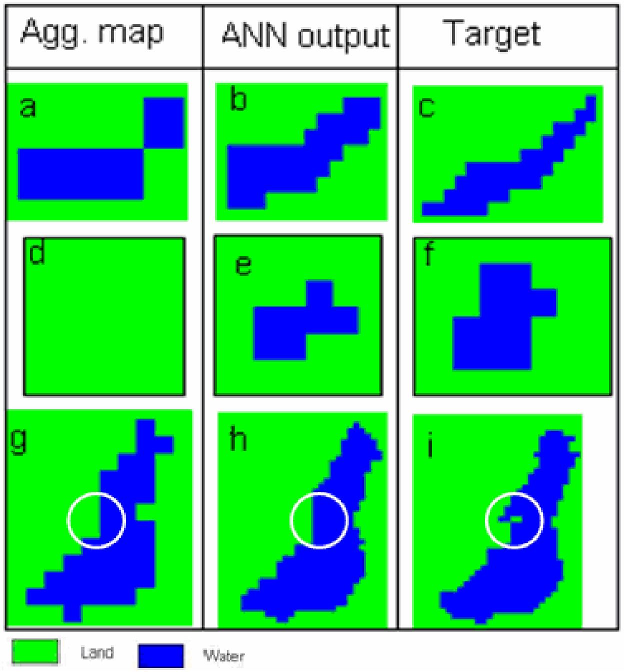


Figure 4.12 Demonstration for two-class case with 75m spatial resolution

In Figure 4.12, on the aggregated map (a), ANN output (b) and target (c) images, a progressive change in shape, boundary roughness and location distortion can be observed. In this figure, the area of the water body on the target image (f) is 12 pixels; but on the aggregated image (d) and ANN output (e), the area diminished to 0 and 7 pixels, respectively. This shows that the neural network somewhat reconstructed the water body that disappeared in the aggregated image based on the fractional proportion of the different classes of the mixed pixels. Even though it is still far to reach the target area, it is a promising indication on the potential of the FFBPNN in predicting locations of class fractions for sub-pixel mapping to minimize information losses. There are also situations where the information (object) has disappeared in both the ANN output and the aggregated map. The small sized feature on the target (i) indicated with circle, is lost in both the aggregated (g) and the ANN output (h). From this image, it can be observed that the ANN output boundaries are much finer than that of the aggregated.

**Table 4-14 Demonstrations on the effect of size and shape of objects for 4 class cases**

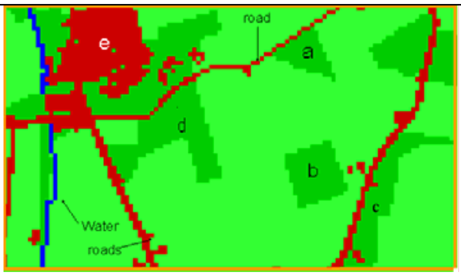
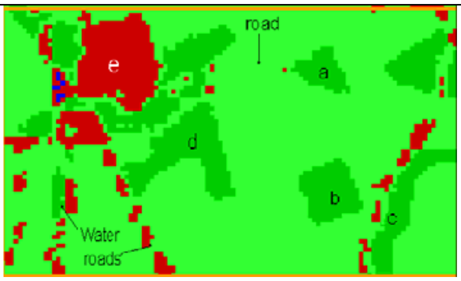
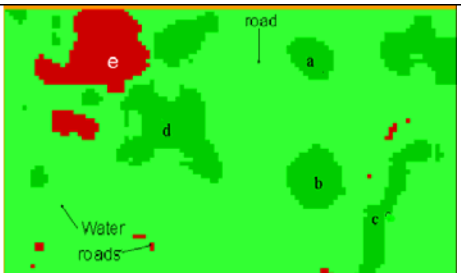
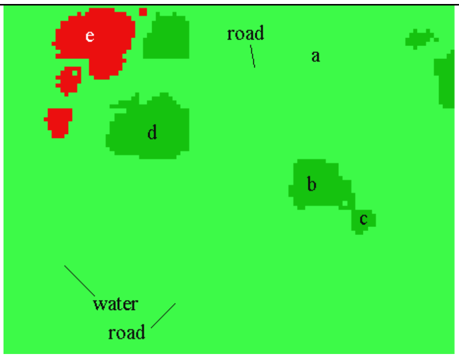
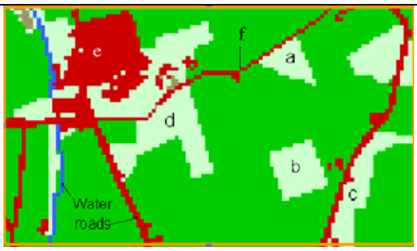
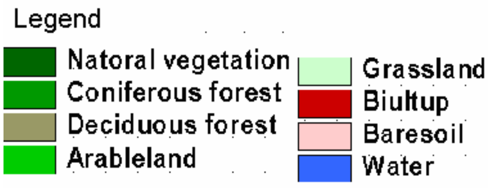
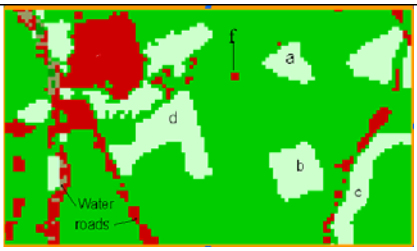
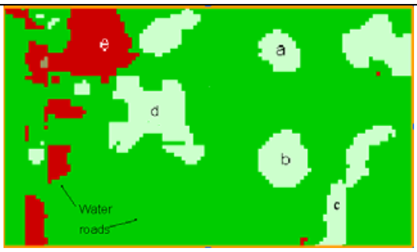
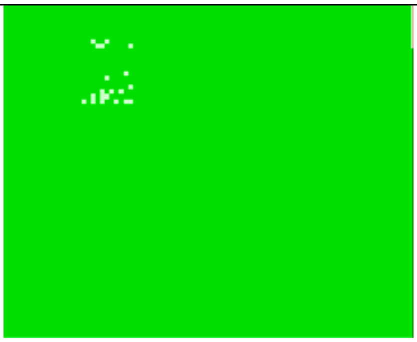
| cases | **4-class cases** | |
|---|---|---|
| Target(25m) |  | This image consists of various shapes (a triangle, rectangle, nearly circular, irregular and linear shapes). The progressive change can be observed in the following consecutive images of the ANN output.<br>**Legend**<br>🟩 Vegetation 🟥 Built-up<br>🟩 Arable land 🟦 Water |
| ANN output(25m) from 75m fraction images |  | From this image:<br>• water body is changed to various classes (mainly to arable land and built-up)<br>• only a small area of water remained (but in 8 class 75m case, it is totally lost)<br>• triangular, square and nearly circular shapes are maintained<br>• linear features dissected (changed to patches) and sometimes changed to other adjacent class type. |
| ANN output(25m) from 150m fraction images |  | From this image it can be observed<br>• all the linear features and small features vanished and<br>• highly generalized result<br>• rounding of shapes (e.g. feature labeled a and b)<br>• change in size (built-up feature e.g. at the bottom left decreased**)**<br>• 3 classes remained: water bodies are totally lost |
| ANN output(25m) from 300m fraction images |  | In this image:<br>• we can see how it is generalized.<br>• all linear, elongated and small sized features are eliminated.<br>• The remaining vegetation classes labeled b & c are connected<br>• Vegetation class (labeled as d) is changed to round shape |

**Table 4-15 Demonstrations on the effect of size and shape of objects for 8 class cases**

| cases | **8-class cases** | |
|---|---|---|
| Target(25m) |  |  Legend<br>Natoral vegetation<br>Coniferous forest<br>Deciduous forest<br>Arableland<br>Grassland<br>Biultup<br>Baresoil<br>Water |
| ANN output(25m) from 75m fraction images |  | In this image the following can be observed:<br>• water is totally lost;<br>• only a small patch remained from the road labeled with f and other roads are fragmented;<br>• The compact shapes (e, b and a) and elongated shapes(c and d) are almost maintained;<br>• a shift of classes (water changed to road). |
| ANN output(25m) from 150m fraction images |  | From this image it can be observed that<br>• all the linear features are vanished;<br>• generalization;<br>• rounding of shape (e.g. feature labeled a and b);<br>• change in size (built-up feature e.g. at the bottom left increased).<br>Shape change (rounding effect) is also reported by Tatem *et al.*( 2001a). This effect according to Tatem *et al.* (2001a) is due to the use of spatial order as the basis that means the network converges to curved rather than sharp corners. |
| ANN output(25m) from 300m fraction images |  | This network generalized all classes at the location to arable land except a few pixels representing grassland are remaining around the upper left side of the image. |

The following issues are general observations that are more or less common problems in all the cases considered in this thesis work.

**Effect of rounding off the values of ANN output to the nearest integer**: this resulted in some shift of classes to the nearest class code despite the spatial adjacency. To elaborate on this: if we have a 4 class case, a neural network output of e.g., 2.4 will be rounded to the nearest integer (to a class code of 2); but it might be class 3. Therefore, this shift to the adjacent class code is encountered in many of the cases and this is the main factor that contributed a lot to degrading the output of the ANN. This effect can also be observed in the confusion matrix. In most cases, classes are largely confused with their adjacent class code(s) (i.e. to the immediate upper or lower class code(s)). For instance, the code used for arable land was 2. In the confusion matrix, it showed that the largest confusions occurred with vegetation (code 1) and/or built-up (code 3), but relatively few or none with water (code 4). However, water (4) is confused with built-up (3), which has the nearest class code to water (Figure 4.13). On the target image, arable land was not present, but appeared on the ANN output image in the surroundings of the water due to the above-mentioned problem. In addition, unclassified pixels were found in many of the ANN output. The unclassified areas vary with the complexity of the cases (see Appendix A). However, this problem is minimum in case number 9. This could be the effect of the normalization that was employed. Such confusion problem may be resolved by using another output encoding technique, such as binary or one-of-N output if computational capacity is not limiting.

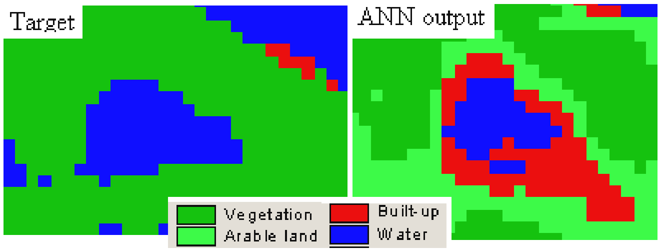


**Figure 4.13 Effect of adjacent class codes on the neighboring class assignment (case 5)**

**Size and contiguity:** From a visual assessment and comparison of both images, it is revealed that isolated objects of small size often disappeared even in the ANN output. Narrow linear features are also changed to a discontinuous and patchy structure. In most cases, these small sized features are changing to the surrounding class type. In most cases, classes with a relatively fragmented land cover pattern are more affected than intact land cover patterns.

### 4.2.7 Effect of applying majority filtering

Some isolated pixels were observed within large classes in the ANN output. To assess the effect of these isolated irrelevant classes, a majority filtering using 3x3 windows was applied to the ANN output. Applying majority analysis on the ANN output slightly improved the accuracy. The results are included in Table 4-2 to Table 4-10 (indicated in brackets). The statistical (overall accuracy and kappa coefficient) and visual comparisons indicated the improvement in all the 9 cases. This shows, somehow, that isolated features were created on the ANN output image. Kasetkasem et al. (2005) also reported that applying a filter improved the accuracy of the sub-pixel mapping outputs.

### 4.2.8 Summary

The accuracy of the neural network output showed a clear decline with an increase in scale factor (or aggregation cell size) and number of classes. In addition, the results showed that the more heterogeneous a pixel is the less its accuracy will be. The result also showed a considerable change in accuracy of the ANN output with change in site heterogeneity. The ANN performance was very low for the relatively heterogeneous site. The result also demonstrated that the accuracy is affected by the shape, size and adjacency of the different features to be mapped. Compact and large size features are easily detectable whereas linear and narrow elongated features are difficult to detect (depending on the scale factor). The main problem that is affecting the ANN output in this work is the class confusion with the nearest (adjacent) class code even if there is no spatial adjacency. Using one-of-N or binary encoding may solve this problem since rounding off the ANN output is not necessary.

The following summary table (Table 4-16) is convenient to compare the overall accuracies and kappa coefficients obtained for all the study cases and sites. In addition, the graph in Figure 4.14 shows the kappa coefficients for all cases and sites. From the figure, the effect of the different factors can be easily seen. It shows the change in accuracy (Z axis) with change in number of classes (X axis), and change in site and spatial resolution (Y axis). In general, the networks performance was poorer in site 2 (since it is more heterogeneous) than site 1. The accuracy shows a clear decline as the number of classes and the scale factor increases.

Table 4-16 Summary of the accuracy assessment results

| classes | Spatial resolution classes | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 75m | | | | 150m | | | | 300m | | | |
| | K1 | K2 | OA1 | OA2 | K1 | K2 | OA1 | OA2 | K1 | K2 | OA1 | OA2 |
| 2 | 0.969 | 0.763 | 98.97 | 98.78 | 0.939 | 0.682 | 98.02 | 98.56 | 0.875 | 0.500 | 95.96 | 97.92 |
| 4 | 0.856 | 0.710 | 89.22 | 81.91 | 0.762 | 0.594 | 82.03 | 74.92 | 0.643 | 0.387 | 73.08 | 62.04 |
| 8 | 0.835 | 0.675 | 87.16 | 75.72 | 0.715 | 0.532 | 77.68 | 65.92 | 0.398 | 0.138 | 51.56 | 38.05 |

K1 & K2 are Kappa coefficients for site 1 &2; OA1 & OA2 are overall accuracies for sites 1 and 2

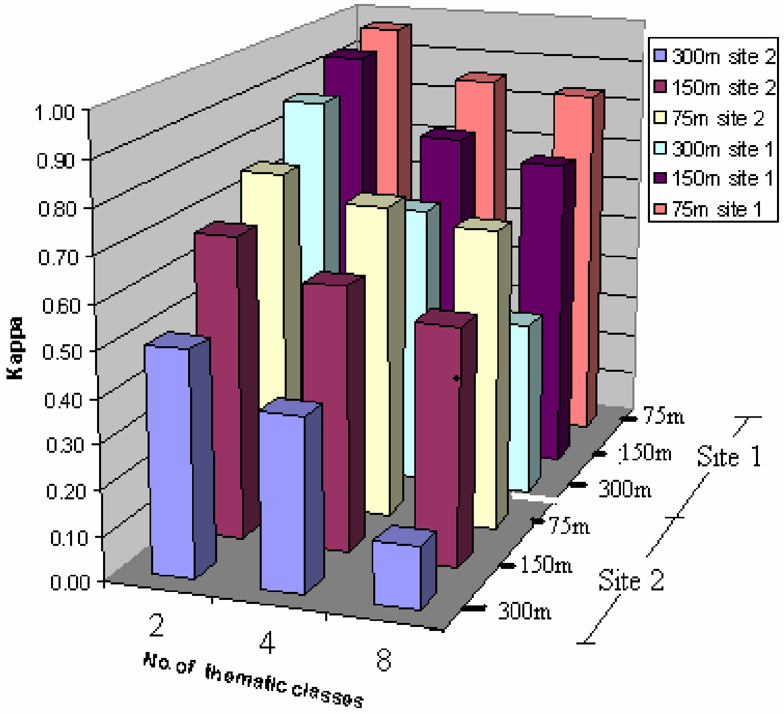


**Figure 4.14 Comparison of accuracy (kappa coefficients) of the study cases for the two sites**

# Chapter 5

# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 Conclusions

In this thesis, the application of a feedforward backpropagating neural network (FFBPNN) for sub-pixel mapping was assessed. In general, FFBPNN implementation is said to be relatively easy. Its successful application can be an important alternative for sub-pixel mapping techniques. This thesis work gave some insights to understand the behavior of the FFBPNN for sub-pixel mapping applications.

Although it was not possible to exhaustively explore all neural network architectures for the various case studies, the results achieved were in line with the 1st (main) research question: 'What are the overall possibilities and limitation of a FFBPNN for sub-pixel mapping?' The results suggest that, with further refinement work, the FFBPNN can have potential application for sub-pixel mapping of land cover as a real soft-classification result of satellite images. The potentials and limitations varied with the complexity of the cases as explained in the following paragraphs.

This work was also intended to answer the following two research questions.
The second research question was: How is the capability of the FFBPNN for pattern/location predictions of sub-pixels within a pixel at the different spatial and thematic aggregation levels? Relevant to this research question, it has been found that the accuracy of the neural network outputs showed a clear decline with an increase in scale factor (or aggregation cell size) and number of input classes. The effect of scale factor was very considerable since it has influences on the number of mixed pixels and output nodes.

The per-pixel and per-site accuracy assessment result confirmed that the network performance noticeably decreased with an increase in pixel heterogeneity and site (location) heterogeneity (spatial frequency). Moreover, the results obtained revealed that the response of the network in reconstructing the sub-pixel map is highly influenced by the shape and size distribution of the different land cover features. Compact and large-sized features are easily detectable whereas linear and narrow, elongated features were difficult for the network to learn accurately. This result was pertinent to the third research question: Is there a change in accuracy with change in spatial frequency, shape and size of the cover types?

In general, the accuracies achieved have a wide range depending on the complexity of the cases. The overall accuracies ranged from 38.05% (complex cases) to 98.97% (simple cases) and the corresponding kappa coefficients range from 0.14 to 0.97. Most of the errors occurred on class boundaries where highly mixed pixels were expected. The errors on the boundaries were those which did not reach to either of the target classes. This was the main problem that was introducing large class confusions with the adjacent class codes, even if there is no spatial adjacency. Using one-of-N or binary output encoding is expected to overcome this problem since rounding off the ANN output to the nearest integer is not necessary.

## 5.2 Recommendations

Though this study showed, to some extent, the possibilities and limitations of FFBPNN for sub-pixel mapping, a further study to determine the best network setting for better result is recommended. Due to computational constraints and as many things need to be tested, the effect of different parameters (including those mentioned below) were not exhaustively explored. The main bottlenecks in this thesis work that hindered a detailed work were training convergence speed and computer memory. The other main problem was the classes' confusion with the adjacent class code. A further detailed work by alternatively testing one or more of the following factors may provide a better result.

- Exploring if adding more hidden layers (if no computational limitation) improves class separability;

- Node and weight pruning algorithms may help to resolve the convergence, computational speed and memory problems, since the less important nodes and weight connections can be removed using this technique;
- Binary or one-of-N output encoding can be an alternative to overcome the adjacent class code confusion problem;
- investigating the effect of the neighboring pixels: using 3x3 window neighbors as input from all layers; only in one of the layers; or only using the central pixel for all layers (no neighbors);
- Exploring the effect of using the entire input fraction layers or disregarding the last input fraction layer (N-1);
- Exploring the initial weight and various early stopping strategies;

Since there are many factors that influence a neural network performance, the recommendations suggested in this section for further research should not be considered as exhaustive.

# REFERENCES

Aplin, P. and P. M. Atkinson (2001). "Sub-pixel land cover mapping for per-field classification." International Journal of Remote Sensing **22**(14): 2853.

Arora, M. K., A. S. Das Gupta and R. P. Gupta (2004). "An artificial neural network approach for landslide hazard zonation in the Bhagirathi (Ganga) Valley, Himalayas." International Journal of Remote Sensing **25**(3): 559.

Atkinson, P. M. (2005). "Sub-pixel Target Mapping from Soft-Classified Remotely Sensed Imagery." Photogrammetric Engineering & Remote Sensing. **71**(7): 839-846.

Atkinson, P. M., M. E. J. Cutler and H. Lewis (1997). "Mapping sub-pixel proportional land cover with AVHRR imagery." International Journal of Remote Sensing **18**(4): 917.

Bhattacharya, U. and S. K. Parui (1997). "An Improved backpropagation neural network for detection of road-like features in satellite imagery." International Journal of Remote Sensing **18**(16): 3379.

Clevers, J., H. Bartholomeus, S. Mücher and A. de Wit (2004). Land Cover Classification with the Medium Resolution Imaging Spectrometer (MERIS). EARSeL eProceedings.

Comber, A., Fisher, P., and Wadsworth, R. (2005). "What is land cover?" Environment and Planning B: Planning and Design **32**: 199-209.

Datta, S. and M. K. Banerjee (2004). "Optimizing parameters of supervised learning techniques (ANN) for precise mapping of the input-output relationship in TMCP steels." Scandinavian Journal of Metallurgy **33**(6): 310.

Engelbrecht, A. P. (2001). "Sensitivity analysis for selective learning by feedforward neural networks." Fundamenta Informaticae **46**(3): 219.

Fisher, P. (1997). "The pixel: A snare and a delusion." International Journal of Remote Sensing **18**(3): 679.

Foody, G. M. (1997). "Fully fuzzy supervised classification of land cover from remotely sensed imagery with an artificial neural network." Neural Computing and Applications **5**(4): 238.

Foody, G. M. (1998). "Sharpening fuzzy classification output to refine the representation of sub-pixel land cover distribution." International Journal of Remote Sensing **19**(13): 2593.

Foody, G. M. (2000). "Estimation of sub-pixel land cover composition in the presence of untrained classes." Computers and Geosciences **26**(4): 469.

Foody, G. M. (2002). "Hard and soft classifications by a neural network with a non-exhaustively defined set of classes." International Journal of Remote Sensing **23**(18): 3853.

Foody, G. M. (2004). Sub-pixel methods in remote sensing. Remote sensing image analysis: including the spatial domain. S. M. v. d. M. de Jong, F.D. Dordrecht, The Netherlands, Kluwer**:** 37-49.

Foody, G. M., Lucas, R. M., Curran, P. J. and Honzak, M. (1996). "Estimation of the areal extent of land cover classes that only occur at a sub-pixel level." Canadian Journal of Remote Sensing **22**(4): 428

Foody, G. M. and D. S. Boyd (1999). "Fuzzy mapping of tropical land cover along an environmental gradient from remotely sensed data with an artificial neural network." Journal of Geographical Systems **1**(1): 23.

Han, M., L. Cheng and H. Meng (2003). "Application of four-layer neural network on information extraction." Neural Networks **16**(5-6): 547.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. NY, Macmillan.

Islam, S. and R. Kothari (2000). "Artificial neural networks in remote sensing of hydrologic processes." Journal of Hydrologic Engineering **5**(2): 138.

Jang, J. D., A. A. Viau and F. Anctil (2004). "Neural network estimation of air temperatures from AVHRR data." International Journal of Remote Sensing **25**(21): 4541.

Kanellopoulos, I. and Wilkinson G. G. (1997). "Strategies and best practice for neural network image classification." International Journal of Remote Sensing **18**(4): 711-725.

Kasetkasem, T., M. K. Arora and P. K. Varshney (2005). "Super-resolution land cover mapping using a Markov random field based approach." Remote Sensing of Environment **96**(3-4): 302.

Kavzoglu, T. and P. M. Mather (2003). "The use of backpropagating artificial neural networks in land cover classification." International Journal of Remote Sensing **24**(23): 4907.

Liang, S. (2004). Quantitative remote sensing of land surfaces. John Wiley & Sons, Inc, USA., John Wiley & Sons, Inc, USA.

Lillesand, T. M. and R. W. Kiefer (2000). Remote Sensing and Image Interpretation. USA, John Wiley & Sons, Inc.

Logar, A., Corwin, E., Watters, S., Weger, R., and Welch, R. (1994). "A Don't Care Back Propagation Algorithm Applied To Satellite Image Recognition." ACM.

Martin, T. H., B. D. Howard and B. Mark (1997). *Neural Network Design*. Boston,PWS Publishing Co.

Mather, P. M. (1999). Land Cover Classification revised. Advances in Remote Sensing and GIS Analysis. P. M. a. T. Atkinson, N.J. England., John Wlly & Sons LTD.

Mathworks, T. (2005). Neural Network Toolbox: User's Guide For Use with MATLAB.

Mertens, K. C., L. P. C. Verbeke, E. I. Ducheyne and R. R. De Wulf (2003). "Using genetic algorithms in sub-pixel mapping." International Journal of Remote Sensing **24**(21): 4241.

Mertens, K. C., L. P. C. Verbeke, T. Westra and R. R. De Wulf (2004). "Sub-pixel mapping and sub-pixel sharpening using neural network predicted wavelet coefficients." Remote Sensing of Environment **91**(2): 225.

Mertens, K. C., L. P. C. Verbeke and R. R. Wulf (2005). Sub-Pixel Mapping: a comparison of Techniques. 25th EARSeL Symposium. Global Developments in Environmental Earth Observation from Space, Porto, Portugal.

Molina, R., J. Abad, M. Vega and A. K. Katsaggelos (2003). "Parameter estimation in bayesian high-resolution image reconstruction with multisensors." IEEE Transactions on Image Processing **12**(12): 1655.

Moller, M. F. (1993). "Scaled conjugate gradient algorithm for fast supervised learning." Neural Networks **6**(4): 525.

Nguyen, M. Q., P. M. Atkinson and H. G. Lewis (2005). "Superresolution mapping using a Hopfield neural network with LIDAR data." IEEE Geoscience and Remote Sensing Letters **2**(3): 366.

Paola, J. D. and R. A. Schowengerdt (1995). "A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery." International Journal of Remote Sensing **16**(16): 3033.

Ranchin, T., L. Wald, B. Aiazzi, S. Baronti and L. Alparone (2003). "Image fusion - The ARSIS concept and some successful implementation schemes." ISPRS Journal of Photogrammetry and Remote Sensing **58**(1-2): 4.

Tatem, A. J., H. G. Lewis, P. M. Atkinson and M. S. Nixon (2001a). "Super-resolution target identification from remotely sensed images using a Hopfield neural network." IEEE Transactions on Geoscience and Remote Sensing **39**(4): 781.

Tatem, A. J., H. G. Lewis, P. M. Atkinson and M. S. Nixon (2001b). Super-resolution mapping of multiple-scale land cover features using a Hopfield neural network. International Geoscience and Remote Sensing Symposium (IGARSS).

Tatem, A. J., H. G. Lewis, P. M. Atkinson and M. S. Nixon (2001c). "Multiple-class land-cover mapping at the sub-pixel scale using a Hopfield neural network." International Journal of Applied Earth Observation and Geoinformation **2001**(2): 184.

Tatem, A. J., H. G. Lewis, P. M. Atkinson and M. S. Nixon (2001d). Super-resolution mapping of urban scenes from IKONOS imagery using a Hopfield neural network. International Geoscience and Remote Sensing Symposium (IGARSS).

Tatem, A. J., H. G. Lewis, P. M. Atkinson and M. S. Nixon (2002). "Super-resolution land cover pattern prediction using a Hopfield neural network." Remote Sensing of Environment **79**(1): 1.

Tatem, A. J., H. G. Lewis, P. M. Atkinson and M. S. Nixon (2003). "Increasing the spatial resolution of agricultural land cover maps using a Hopfield neural network." International Journal of Geographical Information Science **17**(7): 647.

Tatem, A. J., A. M. Noor and S. I. Hay (2004). "Defining approaches to settlement mapping for public health management in Kenya using medium spatial resolution satellite imagery." Remote Sensing of Environment **93**(1-2): 42.

Verbeke, L. P. C., F. M. B. Vancoillie and R. R. De Wulf (2004). "Reusing back-propagation artificial neural networks for land cover classification in tropical savannahs." International Journal of Remote Sensing **25**(14): 2747.

Verhoeye, J. and R. De Wulf (2000). Sub-pixel mapping of sahelian wetlands using ERS2-ATSR images. European Space Agency, (Special Publication) ESA SP.

Verhoeye, J. and R. De Wulf (2002). "Land cover mapping at sub-pixel scales using linear optimization techniques." Remote Sensing of Environment **79**(1): 96.

Xu, M., P. Watanachaturaporn, P. K. Varshney and M. K. Arora (2005). "Decision tree regression for soft classification of remote sensing data." Remote Sensing of Environment **97**(3): 322.

# Appendix A

# Confusion Matrices

## Appendix A-1: Confusion matrix for case 1(2 classes & 75m)

| Ann output | *Reference (Pixels)* | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Site 2** | | | | ***Site 1*** | | | |
| | No data | Land | Water | Total | No data | Land | Water | *Total* |
| No data | 0 | 68 | 1 | 69 | 0 | 20 | 104 | *124* |
| Land | 0 | **775756** | 11300 | 787056 | 0 | **956136** | 10844 | *966980* |
| Water | 0 | 6800 | **192095** | 198895 | 0 | 3236 | **15699** | *18935* |
| *Total* | *0* | *782624* | *203396* | *986020* | *0* | *959392* | *26647* | *986039* |

## Appendix A-2: Confusion matrix for case 2(2 classes & 150m)

| ANN output | **Reference (Pixels)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Site 2** | | | | **Site** | | | |
| | No data | Land | Water | Total | No data | Land | Water | Total |
| No data | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Land | 0 | **771678** | 12621 | 784299 | 0 | **938836** | 9773 | 948609 |
| Water | 0 | 6825 | **188976** | 195801 | 0 | 4177 | **15470** | 19647 |
| Total | 0 | 778503 | 201597 | 980100 | 0 | 943013 | 25243 | 968256 |

## Appendix A-3: Confusion matrix for case 3(2 classes & 300m)

| **ANN output** | **Reference (Pixels)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Site 2 | | | | Site 1 | | | |
| | No data | Land | Water | **Total** | No data | Land | Water | Total |
| No data | 0 | 118 | 16 | **134** | 0 | 0 | 0 | **0** |
| Land | 0 | **733147** | 22254 | **755401** | 0 | **914842** | 13711 | **928553** |
| Water | 0 | 15806 | **173182** | **188988** | 0 | 5958 | **10232** | **16190** |
| **Total** | **0** | **749071** | **195452** | **944523** | **0** | **920800** | **23943** | **944743** |

## Appendix A-4.1: Confusion matrix for case 4(4 classes & 75m) site 1

| **ANN output** | **Reference (Pixels)** | | | | | |
|---|---|---|---|---|---|---|
| | No data | Vegetation | Arable land | Built-up | Water | Total |
| No data | 0 | 95 | 0 | 1 | 0 | 96 |
| Vegetation | 0 | **266158** | 2231 | 4591 | 173 | 273153 |
| Arable land | 0 | 38396 | **216738** | 34759 | 2395 | 292288 |
| Built-up | 0 | 7509 | 3739 | **206394** | 10362 | 228004 |
| Water | 0 | 159 | 3 | 1852 | **190371** | 192385 |
| Total | 0 | 312317 | 222711 | 247597 | 203301 | 985926 |

## Appendix A-4.2: Confusion matrix for case 4(4 classes & 75m) site 2

| **ANN output** | *Reference (Pixels)* | | | | | |
|---|---|---|---|---|---|---|
| | No data | Vegetation | Arable land | Built-up | Water | Total |
| No data | **0** | 87 | 4 | 1 | 1 | 93 |
| Vegetation | 0 | **392653** | 28196 | 13553 | 1499 | 435901 |
| Arable land | 0 | 62540 | **311935** | 40758 | 2615 | 417848 |
| Built-up | 0 | 11909 | 9308 | **85752** | 5228 | 112197 |
| Water | 0 | 1505 | 219 | 970 | **17267** | 19961 |
| **Total** | 0 | 468694 | 349662 | 141034 | 26610 | 986000 |

## Appendix A-5.1: Confusion matrix for case 5(4 classes & 150m) site 1

| **ANN output** | Reference (Pixels) | | | | | |
|---|---|---|---|---|---|---|
| | No Data | Vegetation | Arable land | Built-up | Water | Total |
| No data | **0** | 115 | 0 | 0 | 0 | 115 |
| Vegetation | 0 | **230216** | 5053 | 7859 | 758 | 243886 |
| Arable land | 0 | 64727 | **214530** | 54919 | 5957 | 340133 |
| Built-up | 0 | 15040 | 2758 | **180222** | 15781 | 213801 |
| Water | 0 | 690 | 11 | 2362 | **178573** | 181636 |
| Total | 0 | 310788 | 222352 | 245362 | 201069 | 979571 |

## Appendix A-5.2: Confusion matrix for case 5(4 classes & 150m) site 2

| **ANN output** | Reference (Pixels) | | | | | |
|---|---|---|---|---|---|---|
| | No data | Vegetation | Arable land | Built-up | Water | Total |
| No data | **0** | 17 | 0 | 0 | 1 | 18 |
| Vegetation | 0 | **349016** | 38109 | 14467 | 1725 | 403317 |
| Arable land | 0 | 100946 | **299564** | 57349 | 4685 | 462544 |
| Built-up | 0 | 10234 | 5932 | **65549** | 7555 | 89270 |
| Water | 0 | 1177 | 95 | 557 | **11214** | 13043 |
| Total | 0 | 461390 | 343700 | 137922 | 25180 | 968192 |

## Appendix A-6.1: Confusion matrix for case 6 (4 classes & 300m) site 1

| **ANN output** | Reference (Pixels) | | | | | |
|---|---|---|---|---|---|---|
| | No data | Vegetation | Arable land | Built-up | Water | Total |
| No data | **0** | 239 | 0 | 3 | 0 | 242 |
| Vegetation | 0 | **180550** | 9138 | 9388 | 1012 | 200088 |
| Arable land | 0 | 93388 | **199297** | 66567 | 10648 | 369900 |
| Built-up | 0 | 25319 | 3192 | **156546** | 29817 | 214874 |
| Water | 0 | 1055 | 16 | 4374 | **153398** | 158843 |
| Total | 0 | 300551 | 211643 | 236878 | 194875 | 943947 |

## Appendix A-6.2: Confusion matrix for Case 6 (4 classes & 300m) site 2

| ANN output | **Reference (Pixels)** | | | | | |
|---|---|---|---|---|---|---|
| | No data | Vegetation | Arable land | Built-up | Water | Total |
| No data | **0** | 41 | 10 | 8 | 0 | 59 |
| Vegetation | 0 | **267528** | 63126 | 17522 | 3011 | 351187 |
| Arable land | 0 | 168259 | **265464** | 68469 | 8548 | 510740 |
| Built-up | 0 | 13970 | 6781 | **48162** | 7421 | 76334 |
| Water | 0 | 1065 | 237 | 145 | **4945** | 6392 |
| Total | 0 | 450863 | 335618 | 134306 | 23925 | 944712 |

## Appendix A-7.1: Confusion matrix for case 7(8 classes & 75m) site 1

| **ANN output** | Reference (Pixels) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grass land | Arable land | Built-up | Deciduous | Coniferous | Water | Bare soil | Natural veget | Total |
| No data | 1588 | 5 | 8 | 3 | 2 | 5 | 0 | 0 | 1611 |
| Grassland | **219400** | 2030 | 3872 | 226 | 59 | 57 | 1 | 2 | 225647 |
| Arable land | 34458 | **216055** | 29248 | 1161 | 361 | 474 | 8 | 5 | 281770 |
| Built-up | 9684 | 4311 | **198967** | 3513 | 1394 | 1744 | 24 | 17 | 219654 |
| Deciduous | 1791 | 285 | 5420 | **23977** | 2204 | 3990 | 68 | 81 | 37816 |
| Coniferous | 489 | 24 | 892 | 1591 | **1524** | 8963 | 186 | 279 | 13948 |
| Water | 92 | 1 | 134 | 385 | 68 | **186004** | 480 | 562 | 187726 |
| Bare soil | 7 | 0 | 26 | 159 | 6 | 153 | **7825** | 1545 | 11721 |
| Natural veget | 0 | 0 | 11 | 16 | 0 | 33 | 421 | **5468** | 5949 |
| **Total** | 267509 | 222711 | 238578 | 31031 | 5618 | 203423 | 9013 | 7959 | 985842 |

## Appendix A-7.2:Confusion matrix for Case 7 (8 classes & 75m) site 2

| **ANN output** | *Reference (Pixels)* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grassland | Arable land | Built-up | Deciduous | Coniferous | water | Bare soil | Natural veget | Total |
| No data | 5926 | 444 | 697 | 284 | 253 | 363 | 25 | 12 | 8004 |
| Grassland | **240985** | 29355 | 18120 | 5038 | 3061 | 1516 | 138 | 146 | 298359 |
| Arable land | 33472 | **290769** | 18920 | 4239 | 2264 | 766 | 169 | 116 | 350715 |
| Built-up | 15538 | 14579 | **87212** | 2930 | 2007 | 1025 | 97 | 57 | 123445 |
| Deciduous | 6372 | 4247 | 5156 | **32393** | 4322 | 900 | 60 | 218 | 53668 |
| Coniferous | 3793 | 2020 | 1906 | 5076 | **57954** | 2127 | 78 | 1231 | 74185 |
| Water | 2385 | 1008 | 1038 | 692 | 2520 | **15831** | 107 | 706 | 24287 |
| Bare soil | 3517 | 2724 | 2588 | 1010 | 1482 | 2588 | **251** | 1684 | 15844 |
| Natural veget | 3337 | 2915 | 2990 | 1753 | 2280 | 1187 | 155 | **15944** | 30561 |
| **Total** | 315325 | 348061 | 138627 | 53415 | 76143 | 26303 | 1080 | 20114 | 979068 |

## Appendix A-8.1: Confusion matrix for case 8(8 classes & 150m) site 1

| **NN output** | Reference (Pixels) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grass land | Arable land | Built-up | Deciduous | Coniferous | Water | Bare soil | Natural veget | Total |
| No data | 578 | 4 | 2 | 1 | 0 | 4 | 0 | 0 | 589 |
| Grassland | **181906** | 5558 | 6119 | 437 | 123 | 310 | 2 | 2 | 194459 |
| Arable land | 58293 | **211821** | 47560 | 2594 | 690 | 1802 | 26 | 26 | 322804 |
| Built-up | 18869 | 4744 | **172540** | 5949 | 2587 | 4338 | 67 | 67 | 09135 |
| Deciduous | 4432 | 183 | 8403 | **17639** | 1685 | 8385 | 237 | 237 | 41099 |
| Coniferous | 1752 | 38 | 1929 | 3169 | **465** | 17140 | 694 | 694 | 25502 |
| Water | 300 | 4 | 237 | 852 | 48 | **166669** | 1031 | 1031 | 170200 |
| Bare soil | 28 | 0 | 32 | 260 | 4 | 2835 | **2015** | 2015 | 11440 |
| Natural veget | 1 | 0 | 6 | 19 | 1 | 114 | 3970 | **3970** | 4804 |
| **Total** | **266159** | **222352** | **236828** | **30920** | **5603** | **201597** | **8042** | **8042** | **980032** |

## Appendix A-8.2: Confusion matrix for case 8(8 classes & 150m) site 2

| *ANN output* | *Reference (Pixels)* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grass land | Arable land | Built-up | Deciduous | Coniferous | Water | Bare soil | Natural veget | Total |
| No data | 194 | 9 | 0 | 3 | 8 | 4 | 0 | 11 | 229 |
| Grassland | **190615** | 28912 | 11018 | 2490 | 889 | 482 | 29 | 70 | 234505 |
| Arable land | 92913 | **292326** | 51762 | 10111 | 3574 | 1518 | 168 | 167 | 452539 |
| Built-up | 17821 | 17431 | **67112** | 14075 | 7262 | 2516 | 253 | 190 | 126660 |
| Deciduous | 6877 | 4063 | 5254 | **21762** | 14342 | 3801 | 250 | 292 | 56641 |
| Coniferous | 2784 | 799 | 1584 | 4355 | **43737** | 5338 | 189 | 777 | 59563 |
| Water | 595 | 111 | 117 | 293 | 3373 | **10171** | 109 | 2082 | 16851 |
| Bare soil | 125 | 38 | 14 | 97 | 1707 | 1164 | **60** | 4454 | 7659 |
| Natural veget | 41 | 11 | 1 | 25 | 722 | 249 | 2 | **12442** | 13493 |
| **Total** | 311965 | 343700 | 136862 | 53211 | 75614 | 25243 | 1060 | 20485 | 968140 |

## Appendix A-9.1: Confusion matrix for Case 9 (8 classes & 300m) site 1

| **ANN output** | **Reference** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grass land | Arable land | Built-up | Deciduous | Coniferous | Water | Bare soil | Natural veget | Total |
| No data | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Grassland | **72470** | 8633 | 8734 | 40 | 61 | 1512 | 4 | 279 | 91733 |
| Arable land | 109853 | **189486** | 79813 | 2500 | 1077 | 9461 | 92 | 1118 | 393400 |
| Built-up | 43585 | 10292 | **107776** | 9373 | 3350 | 16763 | 267 | 1170 | 192576 |
| Deciduous | 17476 | 2749 | 19276 | **15958** | 779 | 22388 | 1014 | 2284 | 81924 |
| Coniferous | 10138 | 459 | 10860 | 1131 | **132** | 44585 | 873 | 1673 | 69851 |
| Water | 3700 | 24 | 2802 | 422 | 34 | **97740** | 1406 | 596 | 106724 |
| Bare soil | 318 | 0 | 204 | 97 | 0 | 3173 | **3516** | 871 | 8179 |
| Natural veget | 23 | 0 | 1 | 1 | 0 | 88 | 241 | **43** | 397 |
| Total | 257563 | 211643 | 229466 | 29522 | 5433 | 195710 | 7413 | 8034 | 944784 |

## Appendix A-9.2: Confusion matrix for Case 9 (4 classes & 300m) site 2

| **ANN output** | **Reference** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grass land | Arable land | Built-up | Deciduous | Coniferous | Water | Bare soil | Natural veget | Total |
| no data | 0 | 0 | 2 | 2 | 3 | 4 | 4 | 0 | 15 |
| Grassland | **30097** | 21758 | 5681 | 474 | 924 | 60 | 10 | 443 | 59747 |
| Arable land | 210675 | **262844** | 84149 | 18052 | 21250 | 5665 | 291 | 159 | 607085 |
| Built-up | 44710 | 38258 | **37529** | 20471 | 20201 | 6580 | 361 | 253 | 170363 |
| Deciduous | 14419 | 10773 | 4914 | **11526** | 15871 | 4536 | 182 | 1598 | 63819 |
| Coniferous | 4441 | 1879 | 913 | 1174 | **14356** | 3672 | 78 | 8541 | 35054 |
| Water | 694 | 107 | 169 | 72 | 1585 | **3099** | 4 | 713 | 8443 |
| Bare soil | 1 | 0 | 1 | 20 | 69 | 13 | **0** | 0 | 104 |
| Natural veget | 0 | 0 | 0 | 9 | 28 | 4 | 0 | **0** | 41 |
| Total | 305037 | 335619 | 133358 | 51800 | 74287 | 23933 | 930 | 19707 | 944671 |

# Appendix B

## Lists of neural network training test cases

Case 1: Two-class 75m

| | Test Code | Train algor | Structure | Nhl, | $S_r$ | $N_f$ | Training set | Validation set | Testing set | MSE | Epochs | Site 1 OAA | Site 2 OAA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1_13 | SCG | 9-13-9 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | <0.01 | 63 | 0.9815 | 0.9867 |
| 2 | 1_14 | BFG | 9-16-9 | 1 | 60 | 0.9 | 5400 | 2700 | 2700 | <0.01 | 46 | 0.9811 | 0.9878 |
| 3 | 1_15 | OSS | 9-16-9 | 1 | 60 | 0.9 | 5400 | 2700 | 2700 | <0.01 | 71 | 0.9805 | 0.9875 |
| 4 | 1_16 | SCG | 9-16-9 | 1 | 60 | 0.9 | 5400 | 2700 | 2700 | <0.01 | 46 | 0.9778 | 0.9862 |
| 5 | 1_17 | RP | 9-13-9 | 1 | 60 | 0.7 | 7200 | 3600 | 3600 | <0.01 | 130 | 0.9800 | 0.9855 |
| 6 | 1_18 | RP | 9-16-9 | 1 | 60 | 0.9 | 5400 | 2700 | 2700 | <0.01 | 181 | 0.9795 | 0.9847 |
| 7 | 1_18 | SCG | 9-16-9 | 1 | 60 | 0.9 | 5400 | 2700 | 2700 | <0.01 | 63 | 0.9815 | 0.9875 |
| 8 | 1_19 | SCG | 9-13-9 | 1 | 80 | 0.7 | 7200 | 3600 | 3600 | <0.01 | 51 | 0.9792 | |
| 9 | 1_20 | SCG | 9-13-9 | 1 | 60 | 0.7 | 2700 | 1350 | 1350 | <0.01 | 32 | 0.9808 | 0.98736 |
| 10 | 1_21 | OSS | 9-16-9 | 1 | 60 | 0.9 | 2700 | 1350 | 1350 | <0.01 | 71 | 0.9805 | 0.9875 |
| 11 | 1_25 | SCG | 9-13-9 | 1 | 80 | 0.7 | 2700 | 1350 | 1350 | <0.01 | 51 | 0.9792 | 0.9847 |
| 12 | 1_27 | SCG | 9-9-9 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.0069 | 198 | 0.9866 | |
| 13 | 1_29 | SCG | 9-9-9 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.00654 | 216 | 0.9897 | 0.98785 |
| 14 | 1_31 | SCG | 9-5-5-9 | 2 | 30 | 0.5 | 5400 | 2700 | 2700 | 0.00808 | 321 | 0.9821 | |
| 15 | 1_33 | SCG | 9-5-5-9 | 2 | 30 | 0.5 | 5400 | 2700 | 2700 | 0.00775 | 189 | 0.9829 | |
| 16 | 1_34 | SCG | 9-5-5-9 | 2 | 30 | 0.5 | 5400 | 2700 | 2700 | 0.00950 | 16 | 0.98376 | |
| 17 | 1_35 | SCG | 9-13-9 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.00950 | 33 | | |
| 18 | 1_36 | SCG | 9-3-3-9 | 2 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.00741 | 213 | | |

Nhl: number of hidden layers, $S_r$: sample range factor, $N_f$: node factor, MSE: Mean Squared Error, OAA: Overall Accuracy

SCG: Scaled conjugate gradient, RP: Resilience propagation, OSS: One step scant, Case 2: Two classes and 150m

| | Test code | Train. algor | Structure | Nhl, | $S_r$ | $N_f$ | Training set | Validation set | Testing set | MSE | epoch | OAA SITE 1 | OAA SITE 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2_1 | SCG | 9-23-36 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | <0.01 | 234 | 0.9770 | 0.9849 |
| 2 | 2_2 | SCG | 9-23-36 | 1 | 60 | 0.5 | 5400 | 2700 | 2700 | <0.01 | 529 | 0.9797 | 0.9852 |
| 3 | 2_5 | SCG | 9-23-36 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | <0.01 | 172 | 0.9742 | 0.9845 |
| 4 | 2-7 | SCG | 9-14-14-36 | 2 | 80 | 0.6 | 7200 | 3600 | 3600 | <0.01 | | 0.9754 | 0.9848 |
| 5 | 2_9 | SCG | 9-31-36 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | val | 567 | 0.9802 | 0.9856 |
| 6 | 2_10 | SCG | 9-12-12-36 | 2 | 30 | 0.5 | 2700 | 1350 | 1350 | val | 330 | 0.9734 | 0.9846 |
| 7 | 2_11 | BFG | 9-12-12-36 | 2 | 60 | 0.5 | 5400 | 2700 | 2700 | 0.010667 | 688 | 0.97968 | |
| 9 | 2_13 | BFG | 9-12-12-36 | 2 | 60 | 0.5 | 5400 | 2700 | 2700 | 0.010119 | 197 | 0.96905 | |
| 10 | 2_14 | BFG | 9-12-1236 | 2 | 60 | 0.5 | 5400 | 2700 | 2700 | 0.00962 | 448 | 0.97916 | |
| 11 | 2_15 | SCG | 9-23-36 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.009 | 335 | 0.9796 | 0.9853 |
| 12 | 2_17 | SCG | 9-23-36 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.01012 | 504 | 0.96944 | 0.9856 |
| 13 | 2_19 | SCG | 9-12-12-36 | 2 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.011854 | 251 | 0.97487 | |
| 14 | 2_20 | SCG | 9-12-12-36 | 2 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.010595 | 533 | 0.9802 | |
| 17 | 2_24 | SCG | 9-31-36 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.00865 | 2000 | 0.9801 | |
| 18 | 2_25 | SCG | 9-19-36 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.01061 | 2000 | 0.9642 | |

## Case 3: Two-class and 300m

| | | Structure | Nhl, | $S_r$ | $N_f$ | Training Set | Validation Set | Testing Set | MSE | Epoch | OAA Site 1 | OAA Site 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | SCG | 9-77-144 | 1 | 30 | 0.5 | 2700 | 1350 | 1350 | 0.012413 | 772 | 0.9580 | 0.9786 |
| 2 | SCG | 9-54-54-144 | 2 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.014396 | 1213 | 0.9593 | 0.9791 |
| 3 | SCG | 9-77-144 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.015822 | n/a | 0.9559 | 0.9787 |
| 4 | SCG | 9-77-144 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.015491 | 131 | 0.9574 | 0.9787 |
| 5 | SCG | 9-77-144 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.015293 | n/a | 0.9591 | 0.9788 |
| 6 | SCG | 9-31-144 | 1 | 20 | 0.2 | 1800 | 900 | 900 | 0.013939 | 207 | 0.9562 | 0.9791 |
| 7 | SCG | 9-77-144 | 1 | 030 | 0.5 | 2700 | 1350 | 1350 | 0.01395 | 539 | 0.95042 | 0.9787 |
| 8 | SCG | 9-77-144 | 1 | 030 | 0.5 | 2700 | 1350 | 1350 | 0.013152 | 804 | 0.94255 | |
| 9 | SCG | 9-77-144 | 1 | 030 | 0.5 | 2700 | 1350 | 1350 | 0.011089 | 2000 | 0.9584 | |
| 10 | SCG | 9-77-144 | 1 | 030 | 0.5 | 2700 | 1350 | 1350 | 0.015468 | 493 | 0.95324 | |
| 11 | SCG | 9-77-144 | 1 | 030 | 0.5 | 2700 | 1350 | 1350 | 0.011394 | 2500 | 0.9318 | |
| 12 | SCG | 9-77-144 | 1 | 030 | 0.5 | 2700 | 1350 | 1350 | 0.014192 | 4630 | 0.9420 | |
| 13 | SCG | 9-27-27-144 | 2 | 60 | N/A | 5400 | 2700 | 2700 | 0.014079 | 1475 | | |
| 14 | SCG | 9-107-144 | 1 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.014934 | 168 | | |
| 15 | SCG | 9-54-54-144 | 2 | 60 | 0.7 | 5400 | 2700 | 2700 | 0.016891 | 1197 | | |

## Case 4: 4 classes & 75

| | TEST | TRAIN ALDG. | STRUCTURE | Nhl, | $S_r$ | $N_f$ | Training Set | Validation Set | Testing Set | EPOCJH | OAA SITE 1 | OAA SITE 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4_1 | SCG | 27-25-9 | 1 | 60 | 0.7 | 45360 | 22680 | 22680 | 772 | 0.8921 | 0.8190 |
| 2 | 4_2 | SCG | 27-32-9 | 1 | 40 | 0.9 | 30240 | 15120 | 15120 | 1021 | 0.8899 | 0.8172 |
| 3 | 4_4 | SCG | 27-25-9 | 1 | 60 | 0.7 | 45360 | 22680 | 22680 | 559 | 0.8530 | |
| 4 | 4_5 | BFG | 27-25-9 | 1 | 60 | 0.7 | 45360 | 22680 | 22680 | 422 | 0.8937 | 0.8194 |
| 5 | 4_6 | BFG | 27-13-13-9 | 2 | 60 | 0.7 | 45360 | 22680 | 22680 | 574 | 0.8858 | |
| 6 | 4_7 | SCG | 27-13-13-9 | 2 | 60 | 0.7 | 45360 | 22680 | 22680 | 832 | 0.8529 | |

## Case 5: 4 Classes & 150

| | Test Code | Training Algor | Structure | Nhl, | $S_r$ | $N_f$ | Training Set | Validation Set | Testing Set | MSE | Epoch | OAA Site 1 | OAA Site 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5_1 | SCG | 27-32-36 | 1 | 30 | 0.5 | 22680 | 11340 | 11340 | 0.19837 | 928 | 0.8199 | 0.7491 |
| 2 | 5_2 | SCG | 27-32-36 | 1 | 30 | 0.5 | 22680 | 11340 | 11340 | 0.20226 | 1084 | 0.8112 | 0.7434 |
| 3 | 5_3 | SCG | 27-38-36 | 1 | 50 | 0.6 | 37800 | 18900 | 18900 | 0.20172 | 986 | 0.8148 | 0.7464 |
| 4 | 5_4 | SCG | 27-13-36 | 1 | 7 | n/a | 5292 | 2646 | 2646 | 0.21339 | 294 | 0.7899 | 0.7348 |
| 5 | 5_5 | SCG | 27-27-36 | 1 | 7 | 0.5 | 5292 | 2646 | 2646 | 0.20316 | 255 | 0.8102 | 0.7452 |
| 6 | 5_6 | SCG | 27-5--36 | 1 | 7 | 0.2 | 5292 | 2646 | 2646 | 0.21898 | 422 | 0.7862 | 0.7322 |
| 7 | 5_7 | SCG | 27-7-7-36 | 2 | 30 | 0.5 | 22680 | 11340 | 11340 | 0.21028 | 1670 | 0.7999 | 0.7399 |
| 8 | 5_8 | SCG | 27-32-36 | 1 | 50 | 0.5 | 37800 | 18900 | 18900 | 0.20436 | 1028 | 0.8171 | |
| 9 | 5_9 | SCG | 27-16-16-36 | 2 | 50 | 0.5 | 37800 | 18900 | 18900 | | | 0.8166 | |

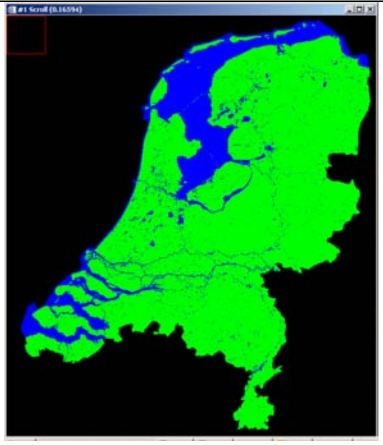
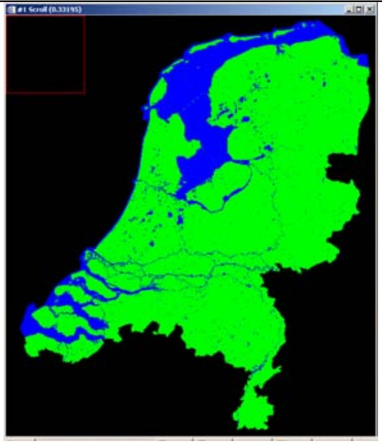
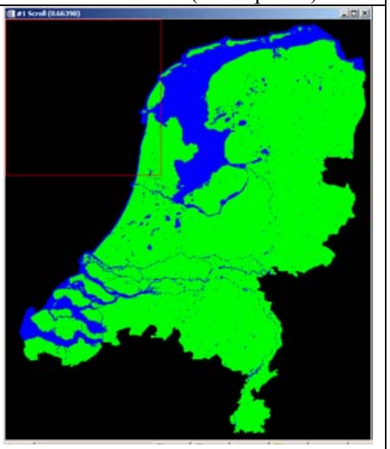
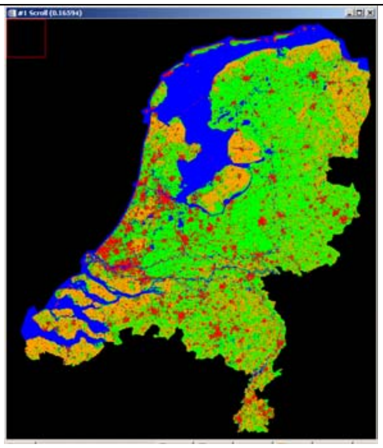
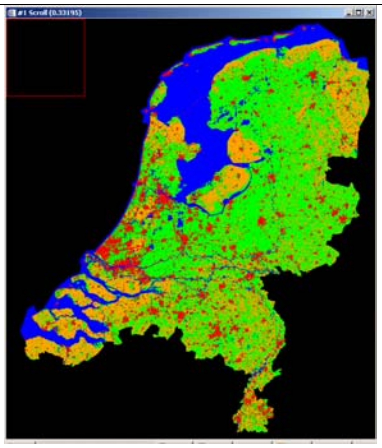
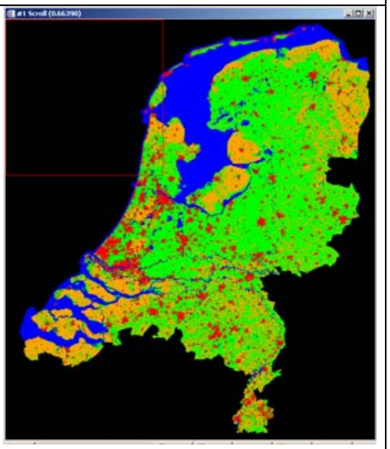
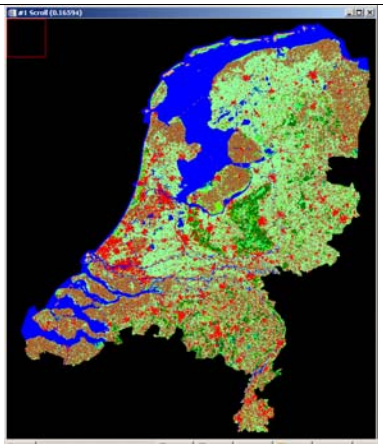
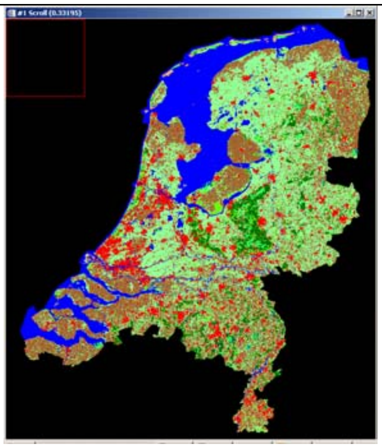
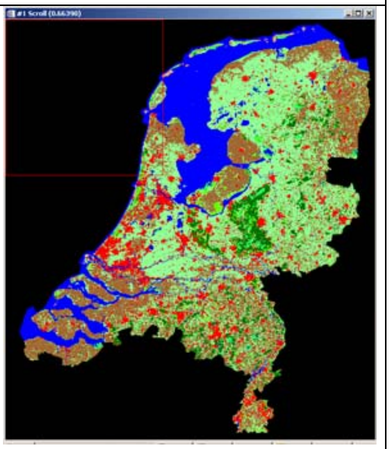
## Case 6: 4 classes & 300

| | Test code | Train Algor | Structure | Nhl, | $S_r$ | $N_f$ | Train | Vald | Test | MSE | Epochs | OAA Site 1 | OAA Site 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6_1 | SCG | 27-86-144 | 1 | 30 | 0.5 | 45360 | 22680 | 22680 | 0.299 | 383 | 0.7113 | 0.6155 |
| 2 | 6_2 | OSS | 27-86-144 | 1 | 30 | 0.5 | 30240 | 15120 | 15120 | 0.2955 | 672 | 0.7217 | 0.6210 |
| 3 | 6_3 | RP | 27-86-144 | 1 | 30 | 0.5 | 45360 | 22680 | 22680 | 0.29748 | 1001 | 0.7110 | 0.6161 |
| 4 | 6_4 | SCG | 27-86-144 | 1 | 30 | 0.5 | 45360 | 22680 | 22680 | 0.29664 | 1097 | 0.7136 | 0.6166 |
| 5 | 6_5 | SCG | 27-86-144 | 1 | 30 | 0.5 | 45360 | 22680 | 22680 | 0.29184 | 2000 | .7301 | 0.6204 |
| 6 | 6_10 | SCG | 27-26-26-144 | 2 | 30 | 0.6 | 45360 | 22680 | 22680 | 0.29531 | 1034 | 0.7257 | 0.6210 |
| 7 | 6_11 | BFG | 27-9-9-144 | 2 | 30 | 0.6 | 45360 | 22680 | 22680 | 0.32712 | 328 | .6933 | n/a |

## 8-class cases

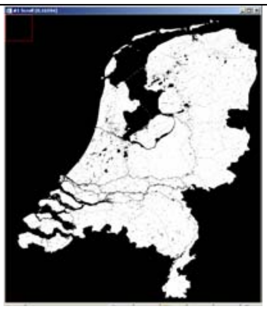
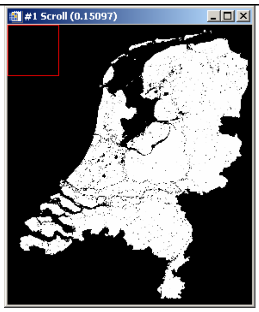
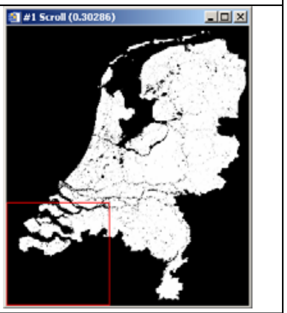
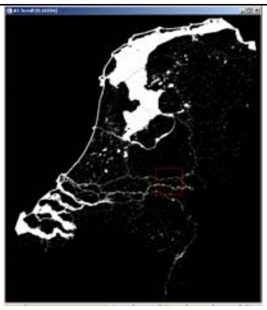
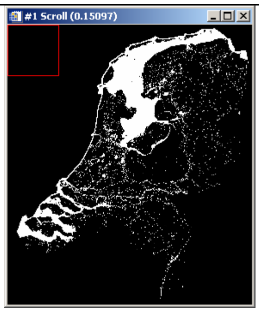
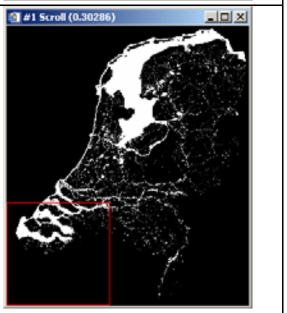
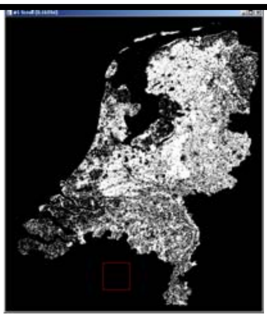
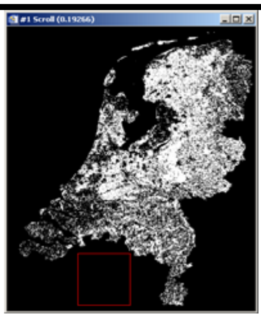
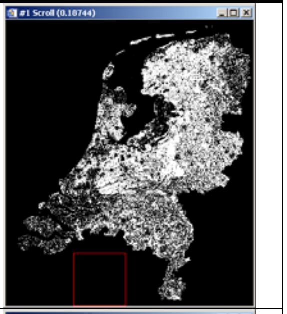
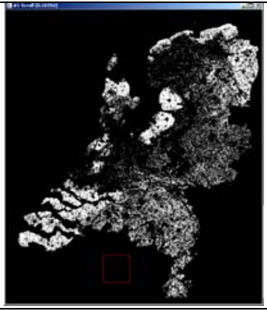
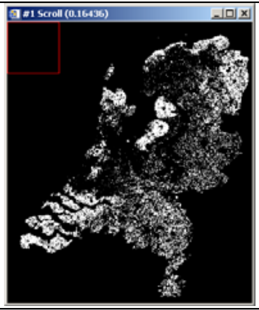
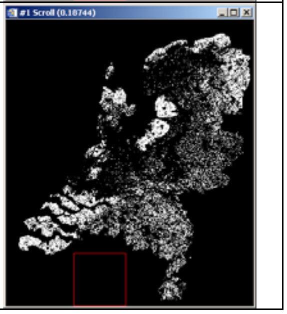
| | Train Alg | Train alg | Structure | Nhl, | $S_r$ | $N_f$ | Train | Vald | Test | MSE | Epochs | OAA Site 1 | OAA Site 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Case 7 8 classes & 75m** | | | | | | | | | | | | | |
| 1 | 7_1 | SCG | 63-68-9 | 1 | 30 | 0.95 | 120960 | 60480 | 60480 | 0.47313 | 610 | 0.8714 | 75.72 |
| **Case 8: 8 classes & 150** | | | | | | | | | | | | | |
| 1 | 8_1 | SCG | 63-66-36 | 1 | 35 | 0.67 | 141120 | 70560 | 70560 | | | | |
| 2 | 8_4 | SCG | 63-25-25-36 | 2 | 30 | 0.5 | 120960 | 60480 | 60480 | 0.458 | 1297 | 0.777 | 0.6592 |
| **Case 9: 8 classes & 300** | | | | | | | | | | | | | |
| 1 | 9_1 | SCG | 63-93-144 | 1 | 22 | 0.45 | 88704 | 44352 | 44352 | | | 0.0186 | |
| 2 | 9_2 | SCG | 7-23-23-144 | 2 | 10 | 0.3 | 45794 | n/a | n/a | 1.3342 | 933 | 0.5151 | 0.381 |

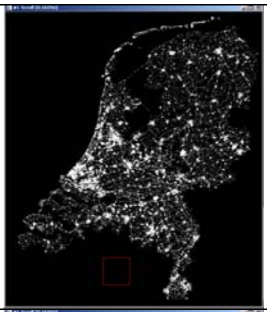
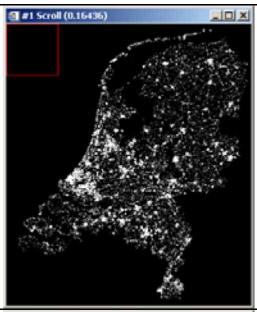
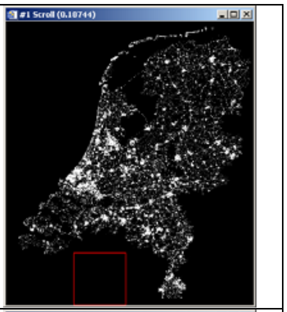
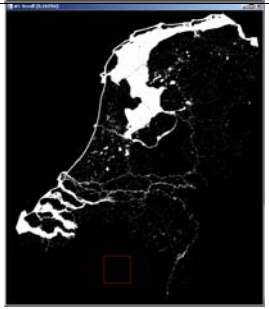
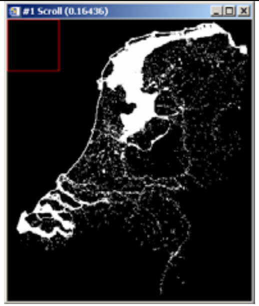
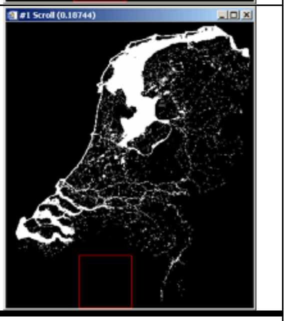
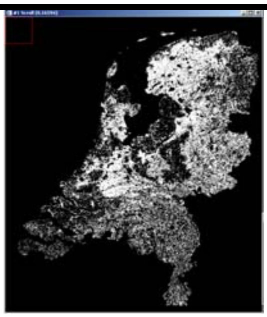
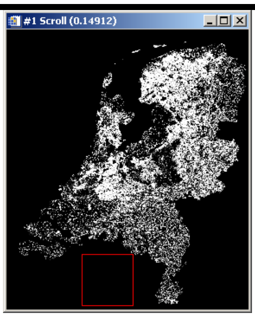
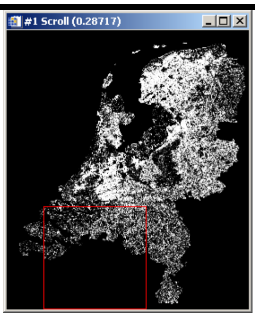
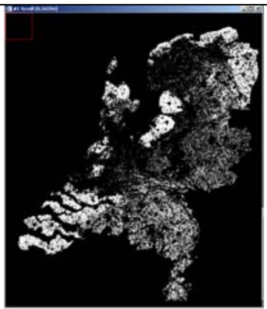
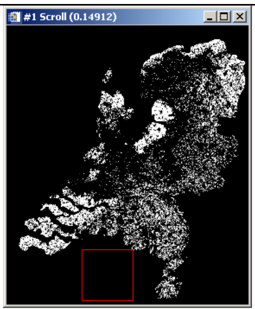
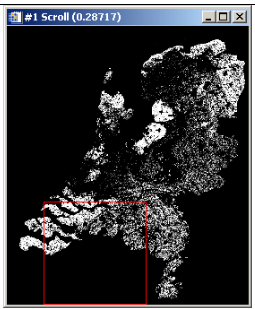
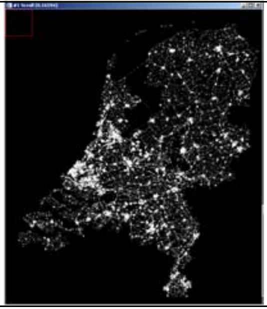
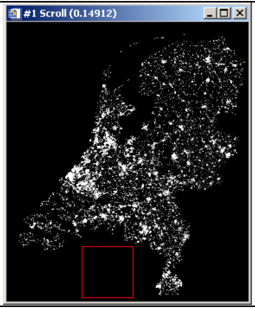
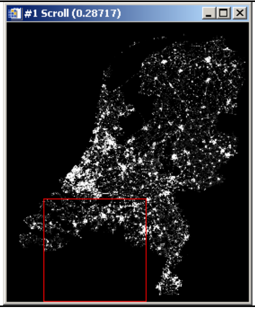
# Appendix C
# Thematically and spatially aggregated maps

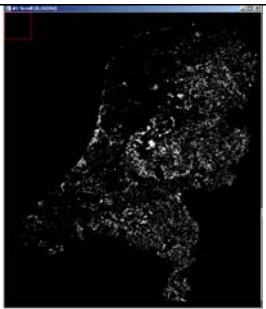
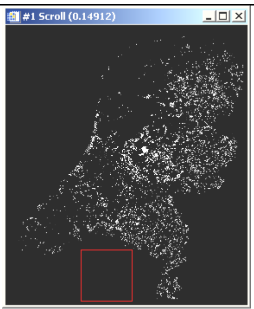
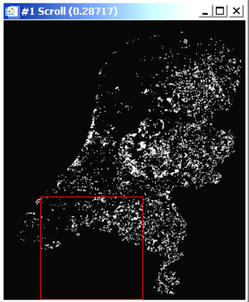
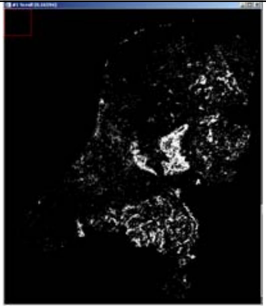
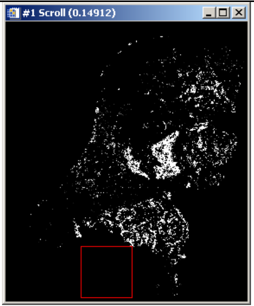
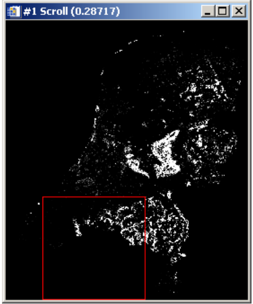
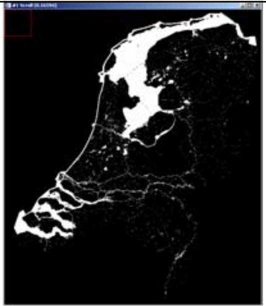
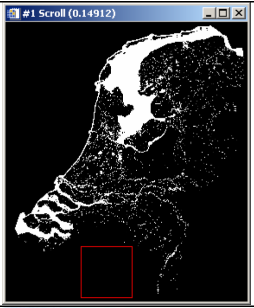
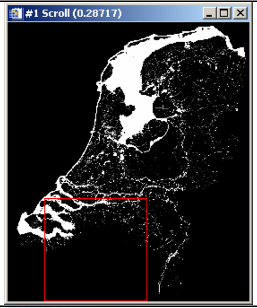
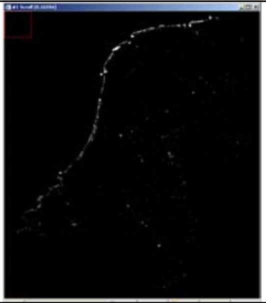
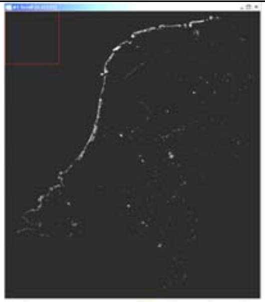
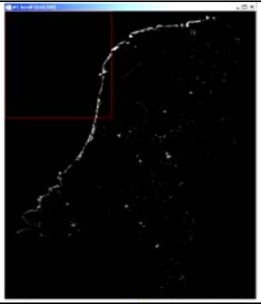
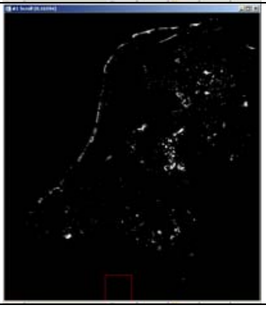
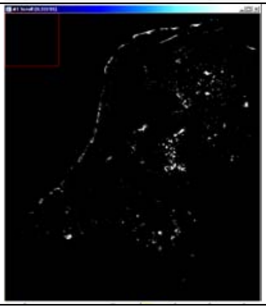
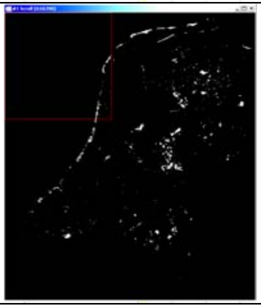
| | 75mx 75m(3 step size) | 150m x 150m(6 step size) | 300m x 300m(12 step size) |
|---|---|---|---|
| 2 classes |  a |  b |  c |
| 4 Classes |  d |  e |  f |
| 8 Classes |  g |  h |  i |

# Appendix D

# Class fraction images

| | Class fraction | 75mx 75m($S_f$=3 ) | 150m x 150m($S_f$=6) | 300m x 300m($S_f$=12) |
|---|---|---|---|---|
| 2 classes | Land |  |  |  |
| | Water |  |  |  |
| 4 classes | Class fraction | 75mx 75m($S_f$=3 ) | 150m x 150m($S_f$=6) | 300m x 300m($S_f$=12) |
| | Vegetation |  |  |  |
| | Arable land |  |  |  |

| | | | | |
|---|---|---|---|---|
| | Built-up |  |  |  |
| | Water |  |  |  |
| 8 Classes | Class fraction | 75mx 75m($S_f$=3 ) | 150m x 150m($S_f$=6) | 300m x 300m($S_f$=12) |
| | `Grassland` |  |  |  |
| | `Arable land` |  |  |  |
| | `Built-up` |  |  |  |

| | | | | |
|---|---|---|---|---|
| | Deciduous forest |  |  |  |
| | Coniferous forest |  |  |  |
| | Water |  |  |  |
| | Bare soil |  |  |  |
| | Natural vegetation |  |  |  |