

LINKÖPING UNIVERSITY

**TECHNICAL REPORT**



TSBB11

December 15, 2015

## Project Identities

CDIO, Autumn 2015  
Linköping University

Name	Responsibility	Telephone	student ID
Victoria Härd	Document Manager	0706173972	vicha350
Hannes Järrendahl	Test Manager	0706830259	hanja146
Sara Shimekaw	Customer Contact	0762326642	sarsh361
Karin Stacke	Project Leader	0735984511	karst367
Joakim Svensk	Quality Assurance	0709963379	joasv627
Gustav Tapper	Scrum Master	0708224495	gusta217
Patrik Tosteberg	Design Manager	0703537678	patto763

**Customer:** Pelle Carlbom, Vricon, pelle.carlbom@vricon.com  
Folke Isaksson, Vricon, folke.isaksson@vricon.com

**Supervisor:** Fahad Khan, LiU(ISY), fahad.khan@liu.se

**Examiner:** Michael Felsberg, LiU(ISY), michael.felsberg@liu.se

## Summary

Cartomatic has developed and evaluated a method for automatic classification of satellite images. The goal was to develop a high accuracy classifier to automatically highlight water and roads from satellite images provided by Vricon. The system incorporates a neural network which has been trained using a ground truth map. OpenStreetMap (OSM) data is used as ground truth and the method is tested and trained on different parts of central Sydney. Pre-processing was done to extract features from the different color band satellite images to use as input for the neural network. Post processing was done to increase the accuracy of the neural network classification output. The output of the system is a classified map represented as a RGB image with every class in each color band; background, water and road.

The resulting images from the system are highly accurate overlay map with roads and water areas highlighted.

## Keywords

Neural network, machine learning, classifier, satellite images, OpenStreetMap, OSM, road detection, water detection, image features, gray-level co-occurrence matrix, height map, softmax, rasterization, geotiff, openCV, GDAL, numpy, ground truth

## Contents

<b>1</b>	<b>Introduction and background</b>	<b>6</b>
1.1	Aim . . . . .	6
1.2	Abbreviations . . . . .	6
<b>2</b>	<b>System overview</b>	<b>6</b>
<b>3</b>	<b>Development environment</b>	<b>7</b>
<b>4</b>	<b>Input data</b>	<b>8</b>
<b>5</b>	<b>Reference Image Extraction</b>	<b>8</b>
5.1	OpenStreetMap . . . . .	9
5.2	Processing of OSM data . . . . .	9
<b>6</b>	<b>Feature extraction</b>	<b>10</b>
6.1	Gray Level Co-occurrence Matrix . . . . .	11
6.2	Height map feature . . . . .	12
6.3	Filtering . . . . .	12
<b>7</b>	<b>Neural Network</b>	<b>14</b>
7.1	Theory . . . . .	14
7.1.1	Sigmoid neurons . . . . .	14
7.1.2	Stochastic Gradient Descent (SGD) . . . . .	15
7.1.3	Backpropagation . . . . .	16
7.1.4	Regularization and overfitting . . . . .	16
7.2	Implementation . . . . .	16
7.2.1	Parameters . . . . .	18
7.2.2	Output . . . . .	18
<b>8</b>	<b>Data imbalance</b>	<b>19</b>
<b>9</b>	<b>Post processing</b>	<b>19</b>
9.1	Morphological operations . . . . .	19
9.2	ProcessG . . . . .	20
<b>10</b>	<b>Results</b>	<b>20</b>
10.1	Feature evaluation . . . . .	20
10.1.1	Block sizes . . . . .	20
10.1.2	Gray levels . . . . .	21
10.1.3	Feature evaluation . . . . .	21
10.1.4	Height map feature . . . . .	24
10.1.5	Filtering . . . . .	24
10.1.6	Total result for feature evaluation . . . . .	25
10.2	Final result . . . . .	25
<b>11</b>	<b>Discussion</b>	<b>31</b>

11.1 Ground truth extraction . . . . .	31
11.2 Feature evaluation . . . . .	33
11.3 Neural network . . . . .	34
11.4 Post processing . . . . .	34
<b>References</b>	<b>35</b>

## Document history

Version	Date	Description	Performed by	Review
1.0	YY-MM-DD			

## 1 Introduction and background

With today's technology it is possible to get very high resolution images of the earth from satellites. This data could be used to create accurate and up-to-date maps. Creating up-to-date maps by hand is a very time consuming and challenging task. An automatic classification system that can create maps from satellite images would therefore facilitate this process.

Vricon is an international company that develops realistic 3D models from various areas of the globe based on satellite images. OSM is an open source project that allows private users from all over the world to add geographic information and in this way together create maps.

### 1.1 Aim

The goal of this project is to classify geographical structures in satellite images provided by Vricon. This will be done using a neural network together with OSM data as reference information. A statistical evaluation of feature extraction and network performance will also be presented. The project will result in a training module and classified 2D images.

### 1.2 Abbreviations

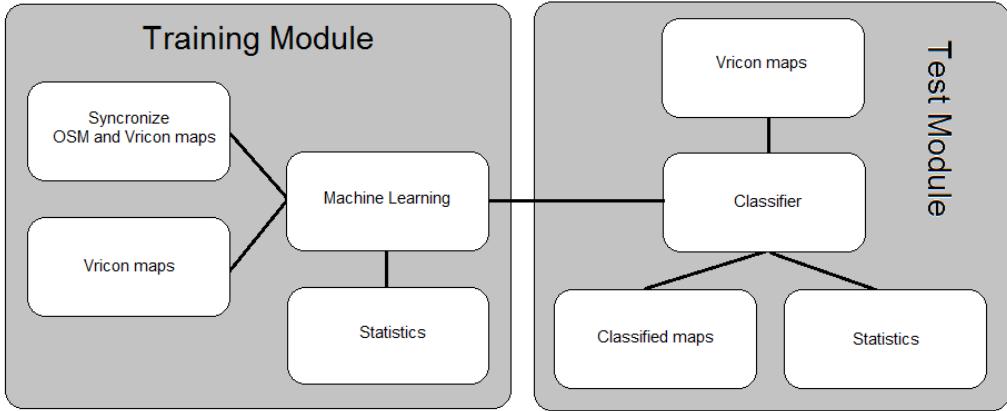
This section lists the abbreviations that are used in this document.

- ANN - Artificial Neural Network
- ASM - Angular Second Moment
- BCC - Background detection correctness coefficient
- CECF - Cross-entropy cost function
- glcm - gray level co-occurrence matrix
- NIR - near infrared
- OSM - OpenStreetMap
- PAN - Panchromatic emulsion
- RCC - Road detection correctness coefficient
- RMSE - Root mean square error
- SGD - Stochastic gradient descent
- WCC - Water detection correctness coefficient

## 2 System overview

A neural network will be trained to identify roads and water from satellite images, and will thereafter produce classified images containing three classes: road, water and background. The training data used as input to the network consists of different features that are extracted from the satellite images. The features aim to give as distinctive description of the three classes as possible in order for the neural network to distinguish between them. In order to verify how accurate the

classification is, a reference image (ground truth) is needed. This reference image was generated from OpenStreetMap data that contains the relevant information (i.e. roads and water). After sufficient training, the network is tested on a different satellite image, and the result of this classification will determine the accuracy of the system. Statistical information about both the training progress and the resulting accuracies are generated. A post-processing step is performed on the resulting image in order to improve on the result further. The aim of this last step is to remove isolated pixels of classes that most likely are misclassified. A flow chart of the system is shown in Figure 1.



*Figure 1: Flow chart of the system. To the left is the training module of the system. The system trains on a Vricon map with a OSM map as ground truth. During the training session, the system output is the accuracy on the training map. The test module (to the right) uses the classifier from the training module to classify another Vricon map cutout and outputs the same statistics of the accuracy and a classified map in RGB (one class in each color band). The only difference between the two statistical outputs from the two modules are that in the training module, the statistics are plotted over time as the network trains, which is not the case for the statistical output of the test module.*

### 3 Development environment

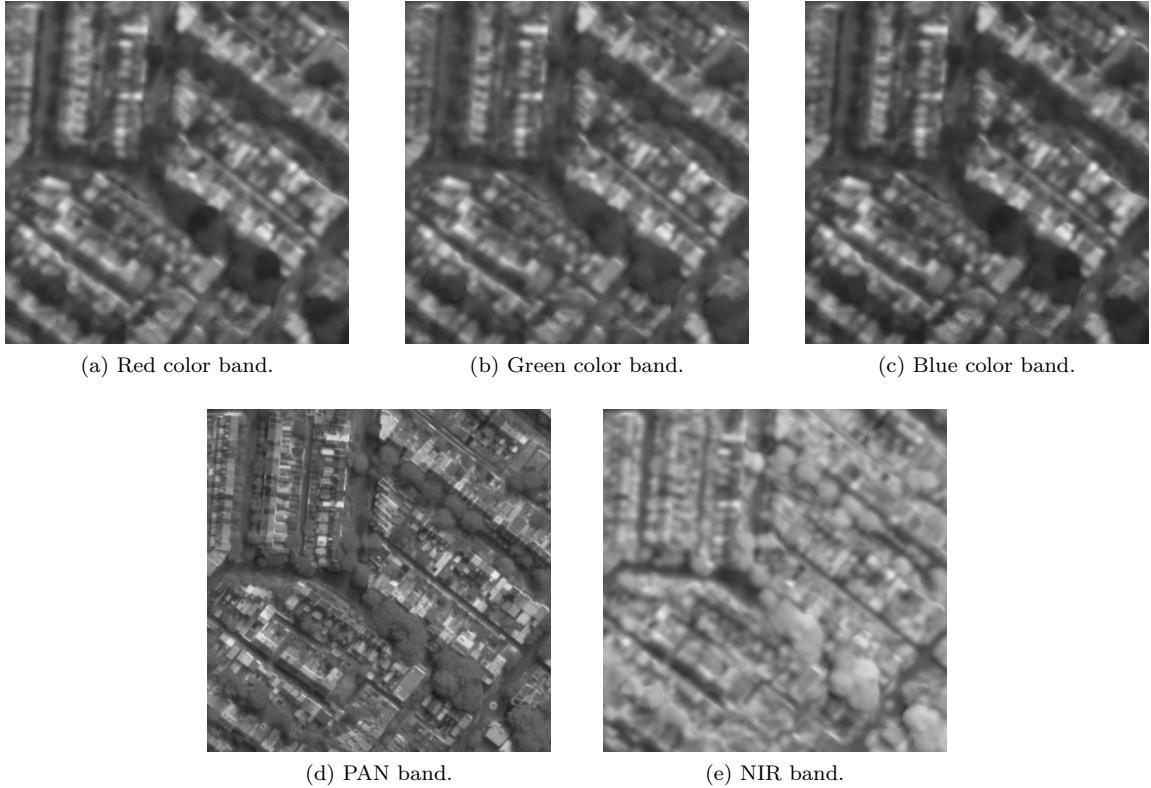
The system was implemented in Python version 2.7. Table 1 lists the extension libraries that were used.

*Table 1: List of extension libraries.*

Library	Version
Numpy	1.9.2
GDAL/OGR	1.11.1
OpenCv	3.0.0
Scipy	0.16.1
Skimage	0.11.3
Sklearn	0.17

## 4 Input data

The data used in this system is satellite images over Sydney, Australia, provided by Vricon. Each image is represented in different color bands: red, green, blue, PAN (panchromatic emulsion) and NIR (near infrared). In addition to these images, height information was available as a dsm-file for all areas covered by the images. Figure 2 shows an example of a small cutout region from one of the provided images in each band.



*Figure 2: Cutout from a satellite image in different color bands provided by Vricon.*

## 5 Reference Image Extraction

A learning system requires a ground truth for some specified area in order to evaluate its results and learn by adjusting its parameters. Our system creates images where roads, water and background are classified so these areas need to be marked in the ground truth. These ground truth images are called reference images, and are in this case images based on OSM data. To create useful reference images, the OSM data had to be transformed, filtered and rasterized. This chapter describes the

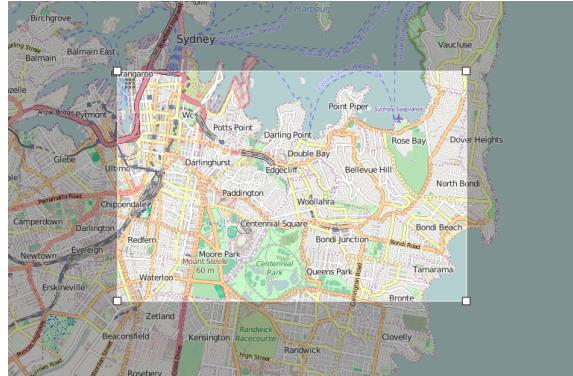
methods used to create these reference images.

## 5.1 OpenStreetMap

OSM is an open source project that aims to create accurate maps of the entire earth. Professional cartographers as well as enthusiastic amateurs can contribute to add new or improve map data. Aerial images, GPS units and low technological field maps are used to make sure that the maps are correct. [1]

OSM maps are available on the web for download, free of charge. The data is downloadable in osm-file format, which is OSM's own file format. It is a type of XML format, i.e. containing text information about nodes, lines and polygons, that has to be rendered using some rendering software to generate the actual map. Each node, line and polygon has a tag, containing a key and a value. By using the keys and values it is possible to choose what type of objects to render. There are thousands of keys and value combinations, ranging from road types to building descriptions to landmarks.

It is possible to retrieve osm-files covering a specified area using an API. The area is specified using latitude and longitude coordinates (WGS84) of four corner points that span a rectangle. Figure 3 shows a print screen from the OSM website that graphically shows what area that is extracted. See figure text for the specific coordinates.



*Figure 3: Print screen from the OSM website graphically showing the area for export, given by coordinates 151.1985, -33.8610 and 151.2788, -33.904.*

## 5.2 Processing of OSM data

The satellite input images were provided in GeoTIFF format. The GeoTIFF format contains information about real world coordinates and map projection, which means that a specific pixel have an exact real world position. The projection and spatial reference system for the input satellite images is UTM zone 56S. The projection and spatial reference system for OSM is WGS84/Web

Mercator. A conversion between the different systems is therefore needed in order to compare and merge the two different data sets. This was done using the open source library GDAL as a plugin to Python.

Our system used the input satellite images areal extent as reference area for the extraction of the OSM data. To create a reference image corresponding to a specific satellite image, the corner coordinates of the satellite image were transformed to the same coordinate system as OSM. The area within these corner coordinates was used as cutout area. To make sure that OSM objects that intersects the cutout area does not disappear, the area was somewhat extended. The corner coordinates of the new, larger cutout area was used as corner points in the API query to extract OSM data as an osm-file.

To render the downloaded data, a couple of extractions and transformations with GDAL were made. SQL queries were used to extract only the wanted object types from the downloaded OSM file, and then convert each of these object types into Shape (.shp) files. This resulted in three shape files, one for thick roads, one for thin roads and one for water. All other data in the OSM file was ignored. All three shape files were then rendered into a single image (.png) file using OpenCV and GDAL, with different parameters for line thickness and color for each of the three object types. The resulting image is our reference image. An example of a reference image created from OSM data is shown in Figure 4.

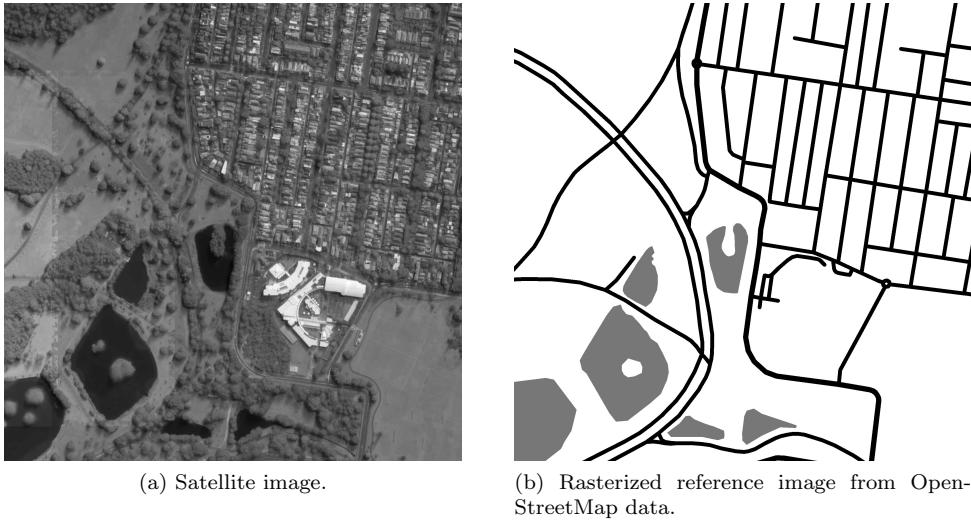


Figure 4: 4a shows the original satellite image and 4b shows the resulting rasterized reference image.

## 6 Feature extraction

The feature extraction subsystem was used to extract and create features from small pixel blocks within the five images (RGB, NIR and PAN). As discussed in [2] it is possible and efficient to use *gray level co-occurrence matrices* (glcm) to extract features from small pixel blocks that can be

used as input to a machine learning algorithm to highlight roads and other geographical structures. The article states that the glcm does not only concern gray level but also the position of the pixels with respect to each other within the block. Each block is constructed from  $N \times N$  pixels. The six features measures that have been evaluated are contrast, dissimilarity, homogeneity, energy, correlation and ASM in each pixel block.

## 6.1 Gray Level Co-occurrence Matrix

To calculate values of the six listed measures, the glcm is calculated by comparing pixel pairs within a block of pixels. A pixel pair is two pixels that lay next to each other in a specific direction. The quantized gray levels (eg. 8 gray levels) of the two pixels are compared. The glcm consists of the combinations of pixel pairs, eg. if one pixel has intensity  $i$  and the next has intensity  $j$  that pixel pair will correlate to a 1 in row  $i$  and column  $j$  in the glcm. The resulting glcm will be  $M \times M$ , where  $M$  is the number of gray levels. [2]

The feature extraction subsystem creates four glcms by comparing pixel pairs in four different orientations: 0,  $\frac{\pi}{4}$ ,  $\frac{\pi}{2}$  and  $\frac{3\pi}{4}$ . These four matrices are normalized by dividing each matrix with the number of pixel pairs, same as the sum of each matrix. The matrices are then added together to form one resulting correlation matrix.

The features are calculated from the gray correlation matrix according to equation 1-6,

$$\text{contrast} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j)^2 a_{i,j} \quad (1)$$

$$\text{dissimilarity} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{i,j} |i - j| \quad (2)$$

$$\text{ASM} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{i,j}^2 \quad (3)$$

$$\text{energy} = \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{i,j}^2} \quad (4)$$

$$\text{correlation} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{(i - \mu_i)(j - \mu_j)a_{i,j}}{\sigma_i \sigma_j} \quad (5)$$

$$\text{homogeneity} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{a_{i,j}}{1 + |i - j|}. \quad (6)$$

where  $a_{i,j}$  is the number of pixel pairs with intensities  $(i,j)$ . These features are stored in a vector that is used as input to the neural network. In conclusion, this part of the subsystem calculates six features for each image, giving the neural network 30 features to work with from each pixel block. An example of how the features look when represented as grayscale images is shown in Figure 5.

As it can be seen in the results, Section 10, different feature combination resulted in different outputs from the classifier. This was expected and therefore an extensive evaluation process was conducted to decide what feature combination that would seem to give the best result. The results of the whole evaluation can be seen in the appendix and the best result is presented in Section 10.1.3 to Section 10.1.6. In addition to this, an evaluation of the optimal size of the pixel blocks and number of gray levels in the glcm was executed.

## 6.2 Height map feature

An additional feature was calculated using height map information provided by Vricon. The height map contains height information relative to the sea level. Since only local height information, i.e. height differences between houses and roads, is of interest in this classification problem, an adaptive threshold was used. The method works by looking at a neighborhood with the size of a user defined kernel. Within this region, the threshold was set to the mean value, and every value above is set to 1 and all other to 0. This results in a binary mapping of the height map, giving information about local height changes. An example is shown in Figure 5i.

## 6.3 Filtering

Filtering the input image before calculating features can be done in order to decrease the effect of noise and make areas more homogeneous. To evaluate the impact of filtering in the resulting classification, some filter tests were included in the feature evaluation in Section 10.1.3. Features were extracted for three different types of input images: no filtering, filtered using a Gaussian filter and filtered using a median filter. Filtering of the input images before calculating the glcm was evaluated using different filter parameters. For Gaussian filters the standard deviation  $\sigma$  was altered and for the median filters different sizes of the kernel were tested.

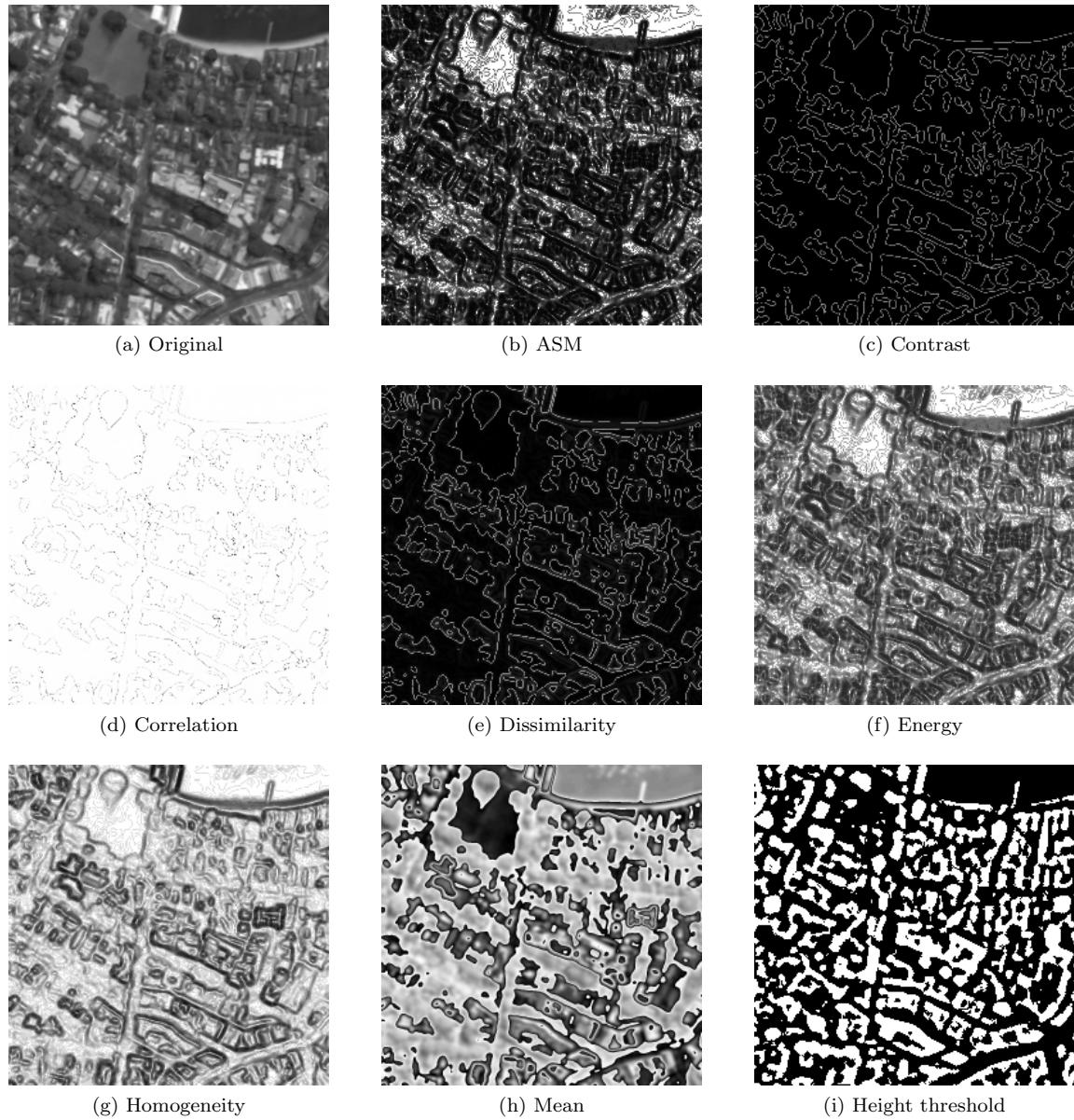


Figure 5: 5a shows the satellite image. 5b-5h shows the different features for the green color band image. 5i show the feature extracted from the height map.

## 7 Neural Network

In this section, the theory of neural networks is described followed by an description of the practical implementation used in this project.

### 7.1 Theory

A neural network consists of multiple neurons arranged in different layers. The input layer contains the same amount of neurons as the number of features and the output layer has the same amount of neurons as classes. The layers between these two layers are called hidden layers and can be used to further enhance the systems performance. When using several hidden layers, the method is called deep learning. In Figure 6 an example of a neural network with only one hidden layer is displayed. The uses of neural networks are nearly endless. Possible applications besides the one presented in this report ranges from automatic recognition of handwritten digits to face detection in mobile phones.

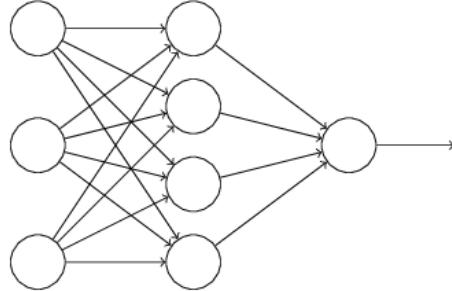


Figure 6: A toy example of a very simple neural network with one hidden layer.[3]

#### 7.1.1 Sigmoid neurons

The neurons that are the nodes in the neural network are often sigmoid functions. A sigmoid neuron can take multiple inputs where each input preferably is a scalar between 0 and 1. Between every neuron in two adjacent layers there is a weight  $w_{jk}$  and every neuron has a bias  $b_j$ . A small change of the weights and biases results in a small change of the neuron output. The output from a neuron is given by  $\sigma(z_j)$ , where  $\sigma$  is the sigmoid function  $\frac{1}{1+e^{-z_j}}$ .  $z_j$  denotes the activation  $w_{jk} \cdot x + b_j$  and  $x$  is the input to the neuron. An illustrative overview of a neuron can be seen in Figure 7. The neuron in the output layer with maximum activation (the neuron with highest output) determines the class label of the input data. The weights and biases are often simply initialized as matrices with Gaussian distributions. For every run epoch of the neural network, the weights and biases from the previous run through are passed on. [3]

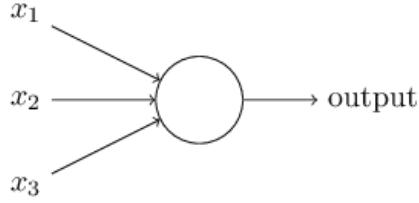


Figure 7: A toy example of a neuron with three inputs and one output.[3]

### 7.1.2 Stochastic Gradient Descent (SGD)

Gradient descent is a first-order optimization algorithm that is used to find a local minimum of some cost function  $C(v) = \frac{1}{n} \sum_x C_x(v)$  that depends on some parameters  $v = \{v_1, v_2, \dots, v_n\}$ . The aim of the algorithm is to update the parameters  $v$  by taking steps in the negative direction of the cost function gradient. This is done according to  $v \rightarrow v' = v - \eta \nabla C$ , where  $\nabla C$  is the vector of partial derivatives  $\frac{\partial C}{\partial v}$ .  $\eta$  is called the learning rate and determines the step size. In practice, the gradient of the cost function  $\nabla C$  is the average of gradients  $\nabla C_x$  of every training input  $x$ . Since the number of training inputs normally is huge, it becomes very time consuming to do a single update of the parameters  $v$ . To overcome this problem, an estimate of the gradient can be calculated for a subset  $m$  of the training inputs. This subset is referred to as a mini batch. This method is called stochastic gradient and speeds up the learning considerably. Applying this in our neural network context, biases and weights are updated according to the equation 8 and 9. This update of weights and biases is done repeatedly to iteratively reduce the cost.

In our implementation, the cross-entropy cost function (CECF) defined in equation 7, was used. The advantages of using CECF instead of eg. a quadratic cost function is that it addresses the issue of slowdown learning. This problem originates from the derivative of the activation ( $\sigma'(z)$ ) in the gradient of the quadratic cost function. If the weights and biases are initialized so that the output of a neuron becomes considerably wrong, the cost function gradient  $\nabla C$  typically becomes very small, and thus the learning will be slow even with a high learning rate. The  $\sigma'(z)$  term is not present in the gradient of the CECF function. [3]

$$C = -\frac{1}{n} \sum_x [y \ln(a) + (1 - y) \ln(1 - a)] \quad (7)$$

$$w_k \rightarrow w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k} \quad (8)$$

$$b_l \rightarrow b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l} \quad (9)$$

### 7.1.3 Backpropagation

The purpose of backpropagation is to compute the partial derivatives  $\frac{\partial C}{\partial w_{jk}}$  and  $\frac{\partial C}{\partial b_j}$  for each layer in the network by introducing an intermediate error  $\delta_j$ . When the input to a neuron comes in, a small change  $\Delta z_j$  is added so the output becomes  $\sigma(z_j + \Delta z_j)$ . This change will then propagate through the layers in the network and it will change the total cost  $\frac{\partial C}{\partial z_j} \Delta z_j$ . The backpropagation algorithm will try to add a  $\Delta z_j$  such that the the total cost is minimized. The error in the j:th neuron in the layer l is defined as  $\delta z_j^l = \frac{\partial C}{\partial z_j^l}$ . This will give us a way to estimate the cost  $\delta_l$  for each layer and then relate it to  $\frac{\partial C}{\partial w_{jk}}$  and  $\frac{\partial C}{\partial b_j}$ . [3]

### 7.1.4 Regularization and overfitting

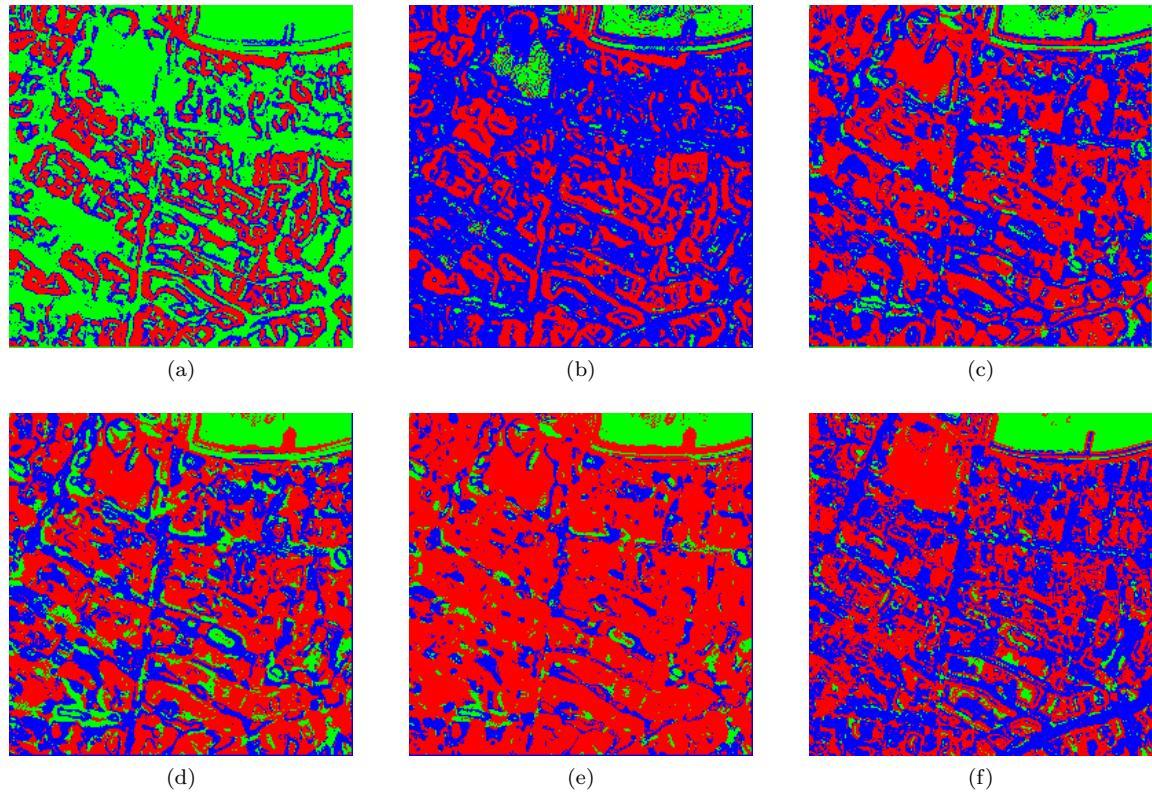
Overfitting occurs if the weights and biases are adapted to give a good classification result on the training data but has low generalization. This may occur if the network is provided with too few input training data samples. By increasing the amount of training data, this problem can be avoided to some extent. To further increase the networks ability to generalize, regularization methods can be implemented. In our system, *L2* regularization is used for this purpose. The *L2* method adds an extra term to the cost function, that is the average over the square of the weights. This will emphasize the use of smaller weights in the network which have been observed to increase generalization. [3]

## 7.2 Implementation

The neural network that was used in the system consists of modified open source Python code, obtained from [3]. The Python implementation of the network is based on the algorithms that are described in Section 7.1 and some less important support functions. The output layer consists of three neurons for each class: background, water and roads. At most, two hidden layers were incorporated and evaluated. The input to the network was constructed as a list, containing tuples for every training sample. Each tuple contains the input feature vector in the first element and the ground truth label in the second element. The ground truth label was represented by a (3,1) vector, where each index represents the class label extracted from the reference image. The vector will contain a 1 at the index of the correct label and zeros at the others. In the case of the validation data, the ground truth is represented by a scalar in the second element of every tuple. This scalar is simply set to 0, 1 or 2, representing the different class labels respectively. At first a simple network was implemented using the main theory described in Section 7.1. However, some of the tweaks to increase the accuracy described were not implemented initially, e.g the regularization parameter and the CECF cost function. In the developing stage, one training image of 2000x2000 pixels as well as a test image of 1000x1000 pixel were used, shown in Figure 8. Since there are a lot of parameters to tune, a lot of effort was put in to getting the best settings. Figure 9 show examples of some early classification results. The main test image is referred to as test number 18 in the rest of the report and was chosen since it contains a fair amount of water and roads and have both difficult and easy areas to classify. For instance, in the bottom right corner there is a very distinct road that is easy to classify but on the other hand, other roads are more complicated since they are covered by trees to a large extent.



*Figure 8: Train and test images used in the developing stage.*



*Figure 9: Examples of early results using the training and test images shown in Figure 8, but with different features and network parameters. Blue indicates road, green indicates water and red indicates background. The result were poor since to little training data was used, to few features were fed to the network and the parameters for network were not well tuned.*

### 7.2.1 Parameters

Network parameters that had to be initialized and optimized included the size of the mini batches, the number of neurons in the hidden layer(s), regularization and the learning rate. After each epoch, information about the class accuracies and cost function are provided to easier draw conclusions about the parameter settings. These statistics have also been used in the extensive feature evaluation. After every epoch, the cost of the training data and evaluation data are stored in different lists. The accuracy for every class is also calculated and stored. After the targeted number of epochs, accuracies and costs are plotted and saved as complementary outputs from the system.

### 7.2.2 Output

The output from the neural network is a RGB classification image of the input test image and a confidence image representing how sure the network is that the classification in different pixels is correct. The network also generates some statistics, including a plot of the cost over time, a plot of the accuracy for the three classes over time and a confusion matrix. The accuracies for roads (RCC), water (WCC) and background (BCC) are calculated simply by dividing the number of correct classified samples with the total number of samples of that class. Examples of a confusion matrix, accuracies and the cost over time can be seen in Figure 10. The y-axis in the confusion matrix corresponds to the actual classes and the x-axis corresponds to predicted classes. The diagonal of the confusion matrix corresponds to the RCC, BCC and WCC accuracies. The other values corresponds to the proportion of a class that is wrongly classified, for example how much of the predicted background that actually is road.

Since the input features are calculated from  $N \times N$  pixel blocks of the input image, the output classification will have to be resized by the factor  $N^2$ . This is done using the cv2 function *resize()* with nearest neighbour interpolation.

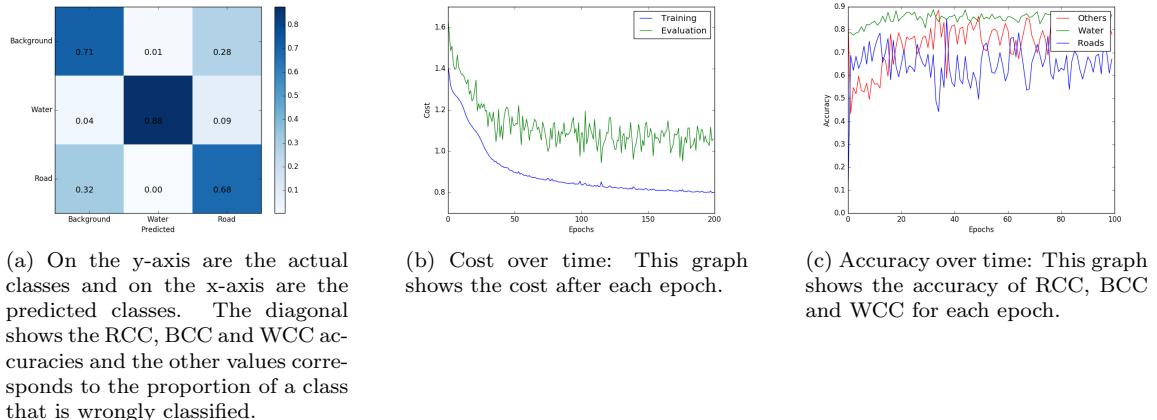


Figure 10: Examples of a confusion matrix, a cost plot over time and a plot of accuracy over time.

## 8 Data imbalance

Learning algorithms are sensitive to imbalances in the class distributions of the training data. From the reference images it can be seen that there are a lot more pixels belonging to the background compared to roads and water. For the image shown in Figure 4b, the occurrence of background is 10 times higher than for road! This over representation of background in the training process will lead to the neural network classifying a vast majority of the pixels as background. One way of solving this problem is to use data reduction. By analysing the class distribution in the reference image, pixels from all classes are chosen such that the total number that is used in the training of the network is approximately the same for all classes. This will in our case result in heavily removing (i.e. not using) pixels belonging to the background class. This solution however introduces another problem. Since the imbalance is so great, the reduction results in too little data and the classifier will not be sufficiently good. An approach tested was then to do the opposite, instead of reducing the overrepresented classes, the data from the underrepresented classes were duplicated until all classes contained the same amount of data. Even though this meant that the imbalance was corrected, and that all the background samples were kept, the duplication of the other classes did not introduce any new data. The network would not learn to classify these two classes any better than using the reduction method.

The solution the group came up with was to divide a satellite image into 1000x1000 sized pixel blocks, and sending features for a number of these blocks as input data to the network, where the data reduction method was used. By doing this, the areas that the network would use for training could be chosen carefully, and did not need to be next to each other in the real world (as would have been the case if just one, larger training image would have been used). Blocks were selected that contained a lot of roads and water in order to get the smallest class as large as possible (all these blocks will naturally introduce a great deal of background). This solved the data imbalance problem as well as supplying the network with considerably more data, to the cost of prolonged running time for the system.

## 9 Post processing

Two methods for post processing were evaluated. This was done to reduce the number of clearly mislabeled pixel blocks. The first method uses the morphological operations erode and dilate, which are common operations in image processing. The other method, "ProcessG", is a self developed method. The post processing was performed on the confidence output from the network, i.e. the neuron values in the output layer values for each pixel block.

### 9.1 Morphological operations

The morphological methods erode and dilate was used to improve the classification. As the name suggests, the dilate part will add a padding around all pixel regions, and erode will remove it. Using these two operations in sequences is an efficient way to remove single pixels of a type that are surrounded by another type, i.e. most certainly a misclassification. If there is too much "noise" in the classification image this may result in pixel clustering instead of removal. The misclassified

areas will in that case grow.

## 9.2 ProcessG

The *ProcessG* method is a method developed by Cartomatic, that is more specific to this classification problem than the above mentioned method. This method uses a specific low-pass Gaussian filter for each class with user specified settings. In the most successful parameter setting configuration, roads and background are filtered with  $\sigma = 2$ . This is to maintain balance between removing road blocks within background regions while still preserving actual road classifications. The roads are typically thin and should for that reason not be extensively filtered. The water is most likely to be found in big, homogeneous sections. Any water appearing in smaller regions is therefore more likely to be a misclassification. The filter standard deviation for this layer is therefore the largest of the three with  $\sigma = 5$ . After filtering, the values for each pixel position  $(i, j)$  are compared between the RGB layers. The layer that has the highest value is set to 1 and the other layers are set to 0. This method will therefore take neighboring pixel values into account and reclassify any pixels that are more likely to be in another class due to their neighborhood.

# 10 Results

In this section the results from the neural network and feature evaluation are presented.

## 10.1 Feature evaluation

An extensive evaluation was executed to find out what features that give the best classification performance. The evaluation included alteration of the following parameters: filtering the image before feature extraction, gray scale level, block size used in the glcm and what features to use.

Below, the results from this evaluation are presented. In all tests, the network was trained for 40 epochs, the learning rate was set to 0,01 and the training data was reduced. One hidden layer was used, containing twice as many neurons as the input layer. To decide which output that gave the best result, the cost function of the network, CECF was used. Lower CECF value means lower cost which indicates a better result. The training and test images used are shown in Figure 8. The result from each test fluctuated and therefore a mean of multiple tests was done to evaluate the result. The result of every test from the feature and filter evaluation can be seen in the appendix.

### 10.1.1 Block sizes

To evaluate what block size to use, the used features as well as the gray level (set to 64) were kept constant, and the block size was varied. 16 features were used in this test; contrast, energy, correlation and homogeneity for the red, green, blue and NIR images. The training data is by the network restricted to be evenly dividable with the block size. Since the training data were

2000x2000 pixels, the four tested block sizes were chosen as 4x4, 5x5, 8x8 and 10x10 pixels. Using larger blocks would not describe the image details and were therefore not tested.

*Table 2: Shows the impact of changing the block size in the feature extraction. The gray level was set to 64.*

	<b>4x4</b>	<b>5x5</b>	<b>8x8</b>	<b>10x10</b>
<b>CECF</b>	1,64	1,66	1,67	1,92
<b>RCC(%)</b>	46,40	46,76	48,79	0
<b>BCC(%)</b>	53,46	52,46	46,46	71,04
<b>WCC(%)</b>	69,43	69,69	71,76	82,77

As can be seen in Table 2, the smallest block size gave the best result. Using smaller sizes was too computationally heavy to be evaluated and was therefore not tested. A block size of 4x4 was used from now on.

### 10.1.2 Gray levels

This section show the results of varying the gray level parameter. A block size of 4x4 pixels was used, and the same 16 features as for the block size test. Using 256 gray levels was too computationally demanding and was therefore not tested. The results were obtained by averaging over the 10 last epochs after 40 epochs.

*Table 3: Shows the impact of gray level reduction in the feature extraction.*

	<b>8</b>	<b>16</b>	<b>32</b>	<b>64</b>	<b>128</b>
<b>CECF</b>	1,86	1,87	1,73	1,64	1,55
<b>RCC(%)</b>	33,31	26,65	36,46	46,40	50,68
<b>BCC(%)</b>	25,69	33,01	47,97	53,46	54,76
<b>WCC(%)</b>	49,89	66,54	60,50	69,43	65,83

As can be seen in Table 3, more gray levels gives a lower cost. Therefore, 128 gray levels was used from now on.

### 10.1.3 Feature evaluation

The feature evaluation was performed by trying both different number of features and different feature combinations. The pixel block mean value was always used as a baseline for the input feature since the intensity in natural images is one of the main features the human visual system uses for object identification. The complete results of this evaluation can be found in the appendix. In the tables below, the mean value as well as the best (i.e. lowest) value of the cost function for every feature combination is shown. The feature combination which gave the best result for a specific number of features are listed in Table 4. The mean values as well as the best values for the respective feature combinations are shown in Table 5 and Table 6.

Table 4: The best feature combinations for 2-7 features (mean is always used as baseline). The gray level was set to 128 and block size 4x4 pixels.

Nr of features	Feature combinations with mean
<b>Baseline (mean)</b>	-
<b>2</b>	Dissimilarity
<b>3</b>	Homogeneity and ASM
<b>4</b>	Homogeneity, Energy and ASM
<b>5</b>	Dissimilarity, Homogeneity, Energy and ASM
<b>6</b>	Contrast, Homogeneity, Energy, Correlation and ASM
<b>7</b>	Contrast, Dissimilarity, Homogeneity, Energy, Correlation and ASM

Table 5: Best CECF and accuracies of the best feature combinations listed in Table 4.

	Baseline (mean)	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>CECF</b>	1,88	1,76	1,45	1,49	1,47	1,47	1,50
<b>RCC(%)</b>	61,53	49,50	44,98	52,37	47,57	47,45	48,77
<b>BCC(%)</b>	28,71	42,75	67,76	60,90	65,20	67,07	63,59
<b>WCC(%)</b>	81,91	83,79	80,11	74,51	79,78	77,21	77,88

Table 6: Mean CECF and accuracies of the best feature combinations listed in Table 4.

	Baseline (mean)	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>CECF</b>	1,89	1,81	1,56	1,53	1,49	1,55	1,57
<b>RCC(%)</b>	61,74	55,53	51,91	51,77	47,71	52,08	54,39
<b>BCC(%)</b>	34,12	36,16	57,65	59,79	63,54	60,79	57,96
<b>WCC(%)</b>	82,39	84,28	78,66	77,71	78,83	79,07	79,15

The CECF values for the best result and the mean results are plotted together in Figure 11. As can be seen from the plot, Table 5 and Table 6, the lowest CECF for both measurements was not obtained using the same number of features. The lowest overall CECF was acquired using three features, but the lowest mean CECF was obtained using five features. Using five features resulted therefore in a more stable result.

In Figure 12, the accuracy measurements (RCC, BCC and WCC) stated in Table 5 and 6 are plotted. From Figure 11 and 12 it is now easy to see that the difference between the mean and best result of one, two, three, six and seven features differs a lot. This implies that the results are unstable and therefore not reliable. Using four or five features gave a bit more stable results. Together with the low and stable CECF result stated above, a conclusion can be drawn that using five features gives the best result.

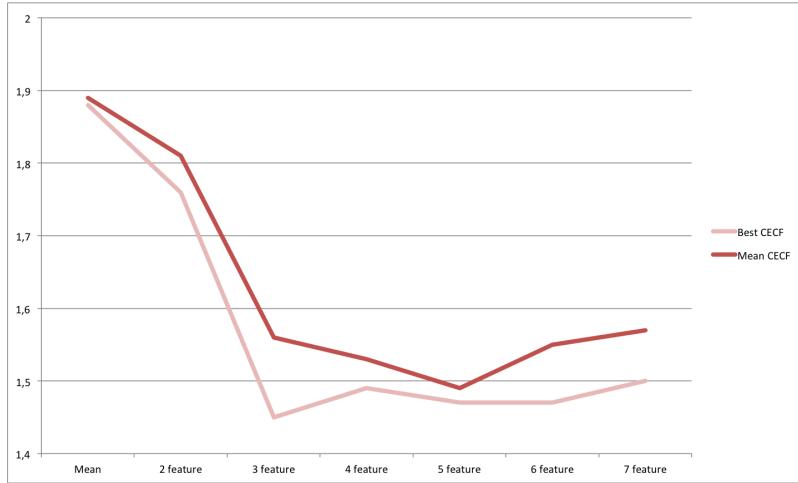


Figure 11: CECF for both the best and the mean results from the feature combination with lowest CECF.

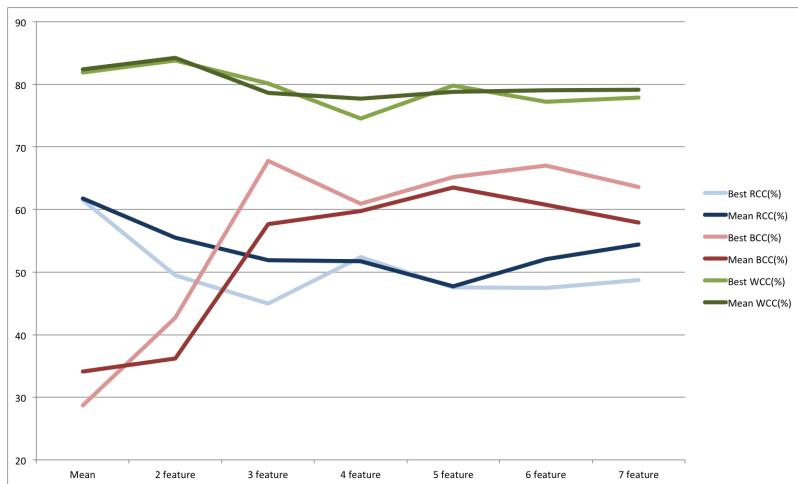


Figure 12: Graph over the class accuracies corresponding to the best and mean value of the CECF for the feature combination with lowest CECF.

#### 10.1.4 Height map feature

To further improve the results, the use of the height map feature was evaluated. The test was done with the feature combination that gave the best result from the previous section. As can be seen in Table 7, adding the height map to the other features decreased the CECF for both the best CECF as well as the mean CECF. Therefore, the height map was included from now on.

*Table 7: Best and mean CECF results with and without using the height feature.*

	Best		Mean	
	with height	without height	with height	without height
<b>CECF</b>	1,42	1,47	1,44	1,49
<b>RCC(%)</b>	62,85	47,57	62,61	47,71
<b>BCC(%)</b>	60,37	65,20	59,90	63,54
<b>WCC(%)</b>	79,16	79,78	80,18	78,83

#### 10.1.5 Filtering

This section shows the results of using different filters to process the image before extracting the features. The first test was done using a Gaussian filter with varying standard deviation ( $\sigma$ ). The result from this test is shown in Table 8. The second test was done using a median filter with varying block size. The results are shown in Table 9. From the values in Table 8 and Table 9, it is easy to see that the filtering did not improve on the resulting classification. Therefore, no filtering was done prior to feature extraction.

*Table 8: Shows the results for the best feature combination (including the height feature) with Gaussian filtration, using different  $\sigma$  values.*

	$\sigma = 2$	$\sigma = 4$	$\sigma = 6$	$\sigma = 8$
<b>CECF</b>	1,50	1,50	1,51	1,51
<b>RCC(%)</b>	65,30	59,59	62,80	77,33
<b>BCC(%)</b>	55,99	60,08	56,77	51,85
<b>WCC(%)</b>	79,60	81,29	83,30	83,90

*Table 9: Shows the results for the best feature combination (including the height feature) with median filtration, with varying kernel size.*

	<b>3x3</b>	<b>5x5</b>	<b>7x7</b>	<b>9x9</b>
<b>CECF</b>	1,57	1,54	1,56	1,55
<b>RCC(%)</b>	61,58	64,49	65,38	62,89
<b>BCC(%)</b>	54,82	56,33	54,94	55,50
<b>WCC(%)</b>	81,12	80,73	78,69	80,29

#### 10.1.6 Total result for feature evaluation

The optimal setting combination from the feature evaluation described above is summarized in Table 10. Using these settings, the resulting input to the classifier would be 26 features (5 features for each 5 images + height map feature). As it can be seen in Table 11, the improvement of the output from the classifier in relation to the baseline when using the parameters which the feature evaluation suggested, improved the result with 24%.

*Table 10: Shows all the optimal parameters from the previous sections.*

<b>block size</b>	4x4
<b>gray levels</b>	128
<b>features</b>	mean, dissimilarity, homogeneity, energy and ASM
<b>height map</b>	yes
<b>filtration</b>	no

*Table 11: Shows the improvement of the result from the classifier after the feature evaluation in relation to the baseline.*

	<b>Baseline</b>	<b>Optimized</b>	<b>Improvement (%)</b>
<b>Best</b>	1,88	1,42	24
<b>Mean</b>	1,89	1,44	24

## 10.2 Final result

The final and best results using the features from Table 10 were achieved with the network parameters presented in Table 12.

*Table 12: Shows all the chosen parameter from the previous sections.*

<b>Layer sizes</b>	[26,52,20,3]
<b>Learning rate</b>	0.005
<b>Mini batch size</b>	10
<b>Regularization</b>	0.5
<b>Achieved after epoch no</b>	106
<b>Training data samples</b>	660000

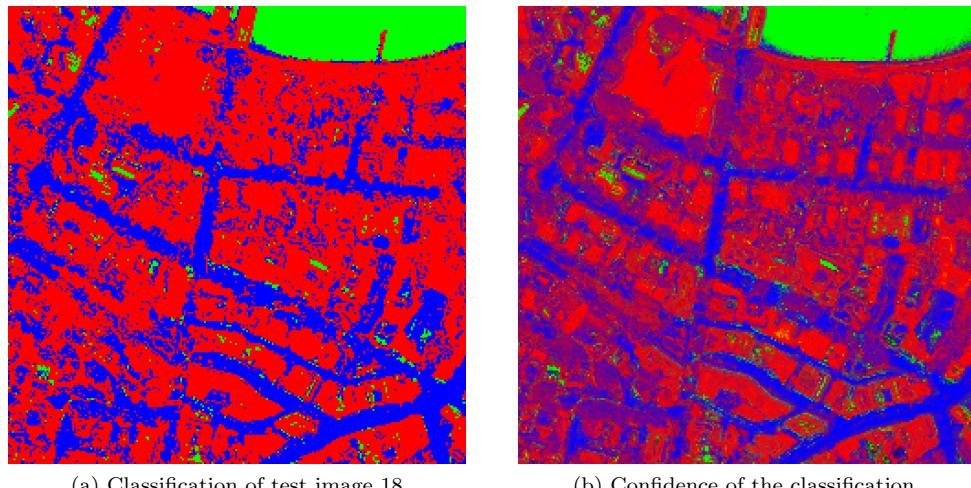
The raw output and confidence information from the network described in Table 12 when evaluated on test image 18 (shown in Figure 8) is shown in Figure 13. Red represents background, green water and blue roads. The confidence image is created by blending the class confidences from the network output into an RGB image. For example, brighter red means that the system is very certain that the classified area actually is background. Purple means that the system is unsure if it should classify these areas as road or background. The accuracies of the classifications in Figure 13a were 73.3 % for background, 90.7 % for water and 73.1 % for roads.

Since the output in Figure 13a is quite noisy in areas that should be homogeneous, the classification accuracies can be increased by applying some post processing. In Figure 14a the output from test image 18 after post processing is shown. The accuracies are ramped up to 82.9 % for background, 94.5 % for water and 76.7 % for roads.

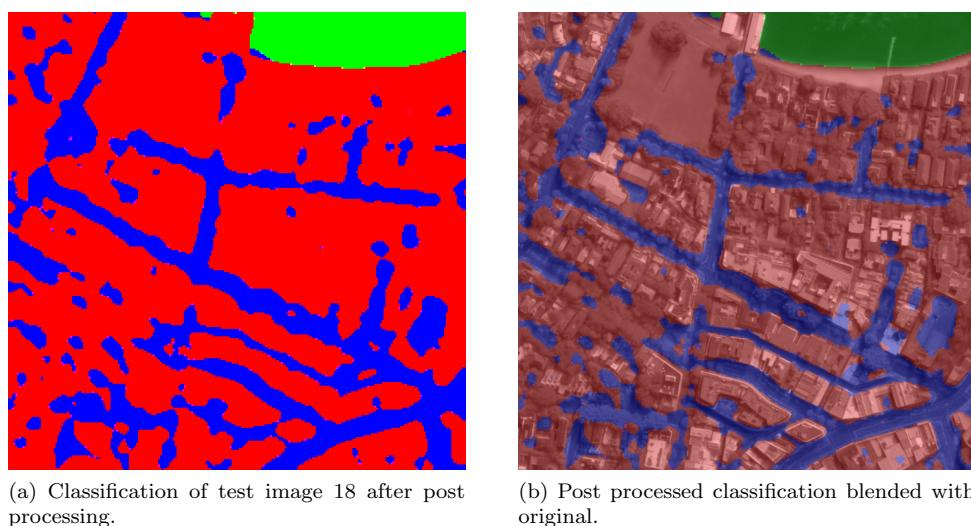
The network used to give the results shown in Figure 14 was also used to classify some other regions besides the main test image 18. In Figure 15, the raw output and confidence information from the network when evaluated on test image 23 is shown. The accuracies of the classifications in Figure 15a were 74.0 % for background, 81.4 % for water and 76.1 % for roads. In Figure 16a the output from test image 23 after post processing is shown. The accuracies are ramped up to 83.1 % for background, 89.3 % for water and 79.2 % for roads. If it would not have been for some inaccuracies in the ground truth data for this image, the accuracies for especially water and roads would have been higher.

In Figure 17 the raw output and confidence information from the network when evaluated on test image 9 is shown. This image contains a distinct and clear water area but much more difficult roads since they are very narrow and blocked by trees to a large extent. The accuracies of the classifications in Figure 17a were 74.2 % for background, 92.5 % for water and 66.7 % for roads. In Figure 18a the output from test image 9 after post processing is shown. The accuracies are ramped up to 84.7 % for background, 98.3 % for water and 68.4 % for roads. This illustrates the systems capability of classifying water.

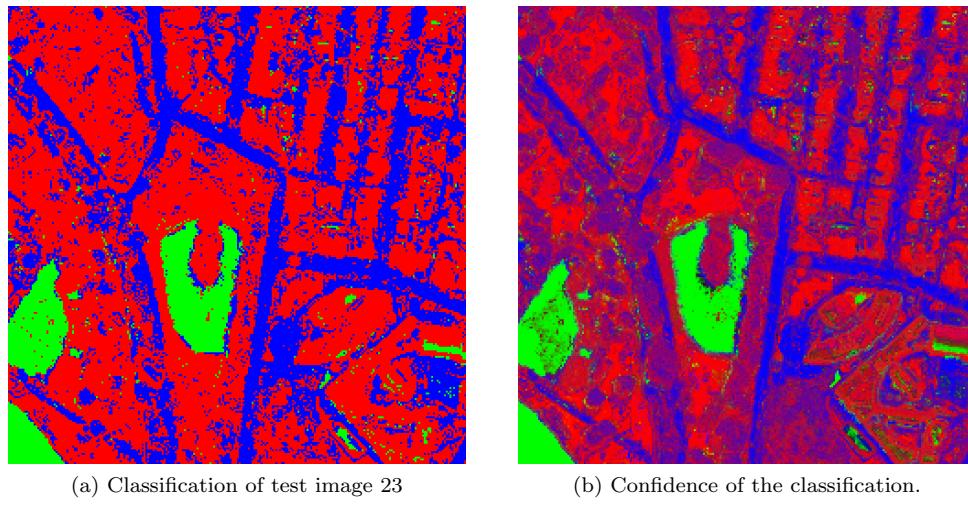
The roads in the previous images have been quite difficult to classify due to blocking trees and narrow widths. In Figure 19 the raw output and confidence information from the network when evaluated on test image 40 is shown. This image contains no water but clearly visible roads. The accuracies of the classifications in Figure 19a were 76.4 % for background and 89.2 % for roads. In Figure 20a the output from test image 40 after post processing is shown. The accuracies are ramped up to 85.6 % for background and 96.4 % for roads. This illustrates that the system is capable of successfully classifying roads with high accuracy in areas with distinct and broad roads. Confusion in the background in 20a are mostly due to driveways since these asphalt patches are predicted as road but in our ground truth, these areas are presented as background.



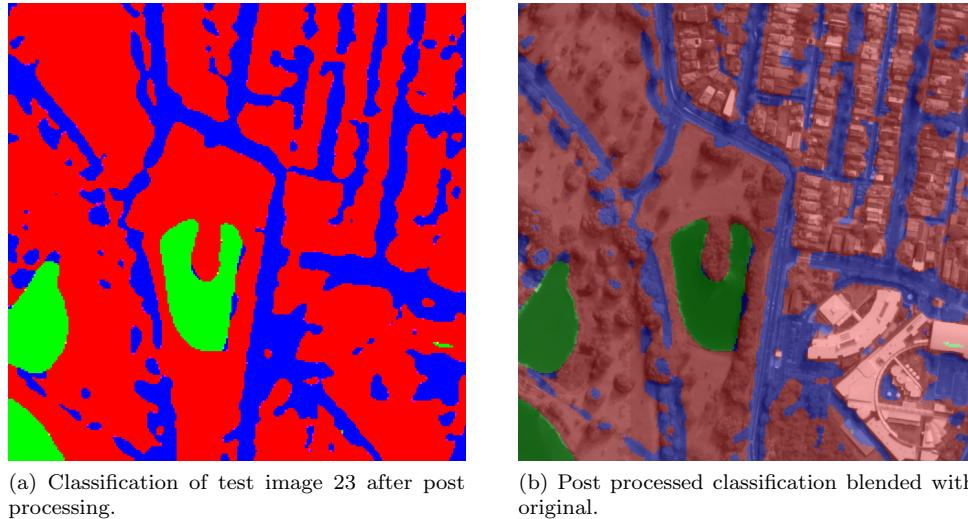
*Figure 13: Classification and confidence results from test image 18. 13a shows the maximum of the output from the classifier without using any post processing. 13b shows the raw confidence output from the classifier. The confidence image is created by blending the individual class confidences from the network output into an RGB image. The accuracies of the classifications in 13a were BCC: 73.3 %, WCC: 90.7 % and RCC: 73.1 %.*



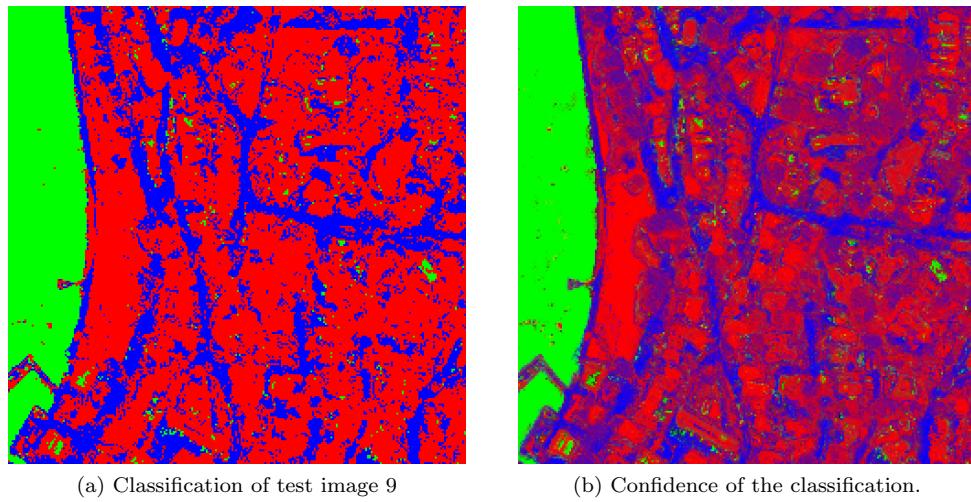
*Figure 14: Post processing results from test image 18 are shown in Figure 13. The post processing was done on the confidence output which are shown in Figure 13b with ProcessG. The accuracies of the classification after post processing in 14a were BCC: 82.9 %, WCC: 94.5 % and RCC: 76.7 %.*



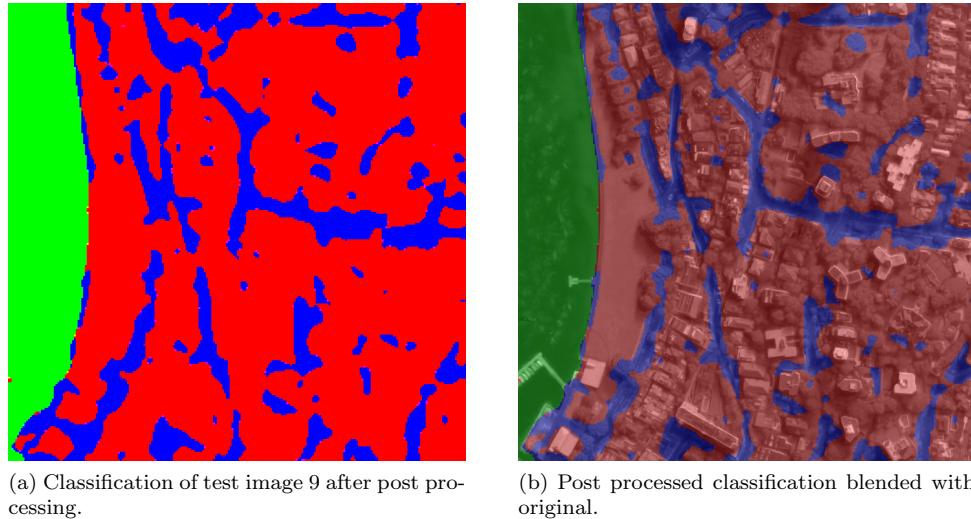
*Figure 15: Classification and confidence results from test image 23. 15a shows the maximum of the output from the classifier without using any post processing. 15b shows the raw confidence output from the classifier. The confidence image is created by blending the individual class confidences from the network output into an RGB image. The accuracies of the classifications in 15a were BCC: 74.0 %, WCC: 81.4 % and RCC: 76.1 %.*



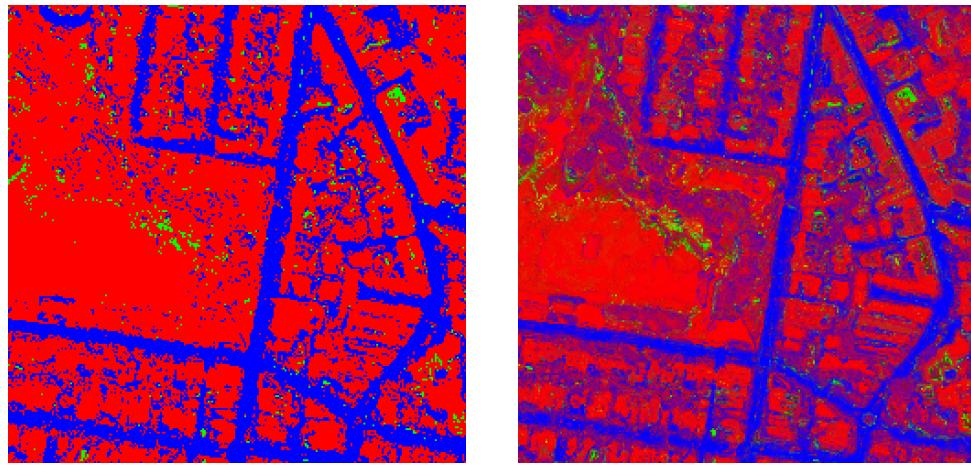
*Figure 16: Post processing results from test image 23 are shown in Figure 15. The post processing was done on the confidence output which are shown in Figure 15b with ProcessG. The accuracies of the classification after post processing in 16a were BCC: 83.1 %, WCC: 89.3 % and RCC: 79.2 %.*



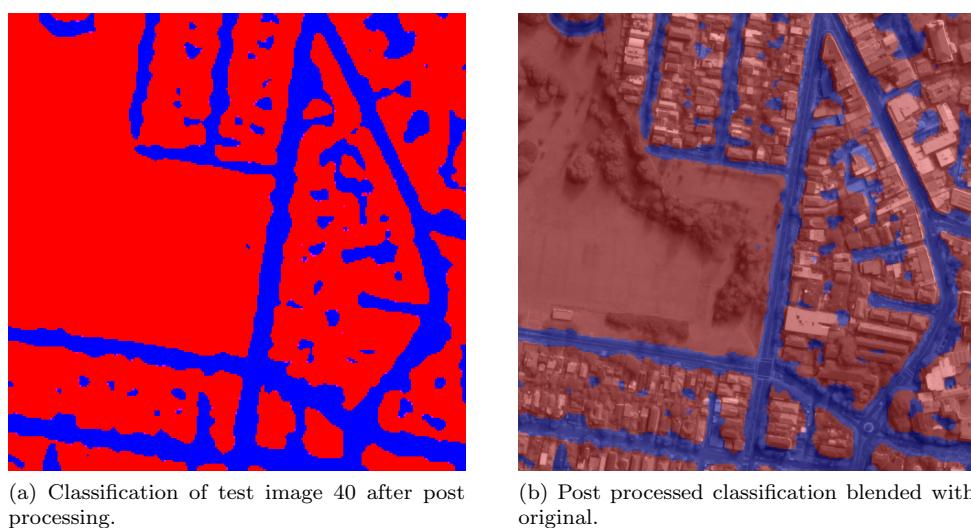
*Figure 17: Classification and confidence results from test image 9. 17a shows the maximum of the output from the classifier without using any post processing. 17b shows the raw confidence output from the classifier. The confidence image is created by blending the individual class confidences from the network output into an RGB image. The accuracies of the classifications in 17a were BCC: 74.2 %, WCC: 92.5 % and RCC: 66.7 %.*



*Figure 18: Post processing results from test image 9 are shown in Figure 17. The post processing was done on the confidence output which are shown in Figure 17b with ProcessG. The accuracies of the classification after post processing in 18a were BCC: 84.7 %, WCC: 98.3 % and RCC: 68.4 %.*



*Figure 19: Classification and confidence results from test image 40. 19a shows the maximum of the output from the classifier without using any post processing. 19b shows the raw confidence output from the classifier. The confidence image is created by blending the individual class confidences from the network output into an RGB image. The accuracies of the classifications in 19a were BCC: 76.4 and RCC: 89.2 %.*



*Figure 20: Post processing results from test image 40 are shown in Figure 19. The post processing was done on the confidence output which are shown in Figure 19b with ProcessG. The accuracies of the classification after post processing in 20a were BCC: 85.6 % and RCC: 96.4 %.*

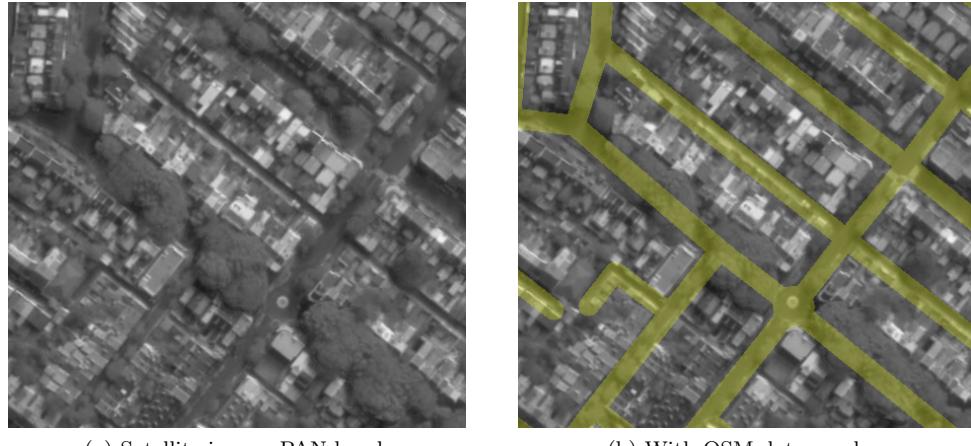
## 11 Discussion

In this section, different issues and problems that have been encountered in the project are discussed, and possible improvements or alternative solutions are mentioned.

### 11.1 Ground truth extraction

The extraction of the desired object types from OSM data involved the determination of what object types to use or not. OSM data contains a large amount of different road object types, ranging from paths and stairways to highways. To decide which object types to use and not to use, each relevant object type was visually inspected one at a time together with the satellite image over the same area. It was then decided for each object type if it was reasonable to include or not. This could be refined by better research about OSM data specifications. When rasterizing the chosen object types, the line thickness for roads was determined in the same way by simple testing and making observations. This could be refined, for example by using more than two different road thicknesses.

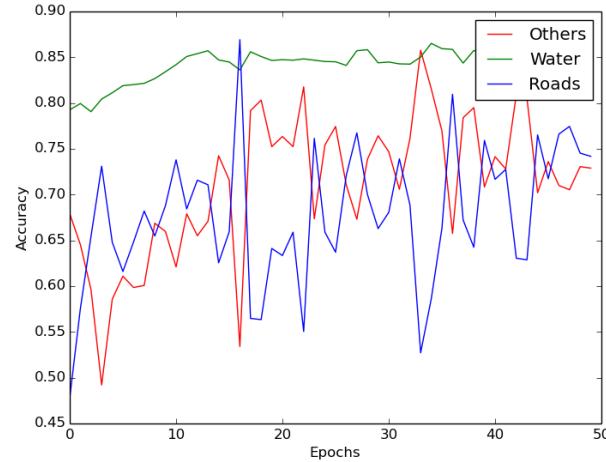
Although there are some possibilities to improve the method for creating reference images from OSM data, the main issue with the reference images is not due to the creation method but to the limitations in the OSM data. The OSM data is not a perfect ground truth since it is a schematic representation of real world features and it is not perfectly drawn. The meaning of schematic representation here is that a map is always a generalization of the complex real world and cannot be perfect. This is of course a problem with all kind of maps. The schematic representation also means that trees for example cannot be distinguished. They will always be part of a more general areal description, for example park or forest. In the provided satellite images, a lot of trees occur close to the roads and frequently cover them partly or completely. This is the main reason that the network has ambiguities in classifying roads and background. A cutout from a satellite image of central Sydney is displayed next to the same image overlaid with the corresponding ground truth in Figure 21. Since there are trees over the roads where the reference image says there are roads, the classifier will train on false data, i.e. learn that roads look like trees. However, since there are trees in the background as well, the classifier will get contradicting information and does not know what to classify trees as - are they road or background? The resulting classification will therefore oscillate between background and road, which is visible in the accuracy plot shown in Figure 22. This also means that the error measurement in this case gives a somewhat misleading figure. Trees on the road should be classified as background, but this will result in a bigger error since the reference image indicates road. In the same way, the error will be reduced if these trees are classified as road. Our conclusion of this is that OSM data is not optimal as ground truth if very accurate classification maps are of high importance. For this, the usage of ground truths with more detailed background is vital, e.g reference images that distinguish houses from trees.



(a) Satellite image, PAN band.

(b) With OSM data overlay.

*Figure 21: Illustration of the problem with trees covering the road. 21b show an overlay of the OSM data (shown in yellow).*



*Figure 22: Accuracy plot clearly showing the oscillation of the accuracy between road and background. This oscillation is due to the contradicting information introduced by trees overlaying roads.*

Another example of how the OSM data is an incorrect representation of the real world is shown in Figure 23. As seen in the figure, a perfect classification (i.e. very high accuracy) is not possible, since the ground truth is not correct.



*Figure 23: In 23b it is apparent that the OSM data (shown in yellow) is not a perfect ground truth. The upper road is slightly shifted from the actual one, and in other parts the road overlays buildings.*

## 11.2 Feature evaluation

Doing feature extraction block by block instead of evaluating one pixel at the time can be insufficient, since the contents of each block might vary too much to give any valuable information. To classify each pixel by looking in a neighboring region of size  $N \times N$ , would however be very computationally demanding. This would result in run times that are too long to be practical, but might generate better results.

As discussed above, the computational time was necessary to be kept at a reasonable level, so the amount of gray levels had to be limited. Since the glcm creates a matrix for each block of the same size as the amount of gray levels, it was very computationally demanding to use 256 levels. But as can be seen in the result, more gray levels seem to increase the output accuracy. Therefore, it might be good to use 256 levels if the computation time is not a crucial factor.

The feature evaluation approach can be criticized. Since the output result fluctuated quite much between epochs, the mean of the accuracies and costs for the last 10 epochs was used to draw conclusions. However, since this measure also varied to some extent, the tests were repeated several times to be able to draw conclusions. Perhaps the test still is not statistically accurate, but we were able to see a clear trend in the tests for different features. Therefore, we felt confident enough to draw the conclusion stated in section 10.1.6, that the five chosen features were optimal to use for our problem.

### 11.3 Neural network

The choice of using an artificial neural network (ANN) for our classification problem was not obvious. There are numerous other learning algorithms that could have been used as well. Major advantages with ANNs are their ability to learn and generalize. On the other hand, the path towards getting good results is often not very straight forward and includes a vast amount of parameter tuning. There are not concrete rules of how these parameters should be tuned to get optimal results. Except some rules of thumb, the only way of achieving as good results as possible is to tune different parameters iteratively, one at a time. This is extremely time consuming and hard to interpret since different combinations might behave in unexpected ways.

The possibility of going towards a deep neural network was not extensively investigated. At most two hidden layers were used and this was also the setting that gave the best final results. By incorporating more hidden layers, the results could be improved on even further. However, optimizing parameters for these more complicated networks is a time consuming task and did not fit within the limitations of this project.

### 11.4 Post processing

Our post processing methods improves on the classification in most cases according to the accuracy measure. However, roads might be easier to distinguish to the human eye before the processing. Our brain is magnificent at extracting important structures from noisy data if it is clear what we are looking for. However, the post processing in combination with overlaying the classification result on the original image gives a very distinct result. The post processing is very successful in removing small areas of misclassified pixels, but this may also result in removing correctly classified data. In the ideal case, post processing would not be necessary since the classifier then would correctly classify almost all the pixels. In our case, post processing is absolutely needed due to the ambiguous information in the background areas in the ground truth data.

The two evaluated methods gave similar results. Both share the same problem of being highly dependant on the quality of the input image. Since different regions look different the classifier will be better at classifying some images than others. Harder regions will therefore result in a larger error, and a higher degree of post processing is then necessary. Making the processing methods in some way dependant of the classification error might be a solution, but that would require a more complicated algorithm. Another interesting approach would be to try and find oblong, narrow structures and join these parts with road pixels where these structures are disrupted by small amounts of background.

## References

- [1] *About.* URL: <http://www.openstreetmap.org/about>.
- [2] M. Mokhtarzade, H. Ebadi, and M.J. Valadan Zoej. “Optimization of road detection from high-resolution satellite images using texture parameters in neural network classifiers”. In: *Canadian Journal of Remote Sensing* 33.6 (2007), 481–491.
- [3] M. A. Nielsen. *Neural Networks and Deep Learning*. 2015. URL: <http://neuralnetworksanddeeplearning.com/index.html>.

# APPENDIX

## Feature evaluation

### Evaluation of one and two features

*Table 13: Statistics of one and combinations of two features*

	mean	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	ASM	
Others %	28,71	39,88	25,14	45,38	42,75	40,13	30,42	
Water %	81,91	86,38	84,1	84,84	83,79	79,05	88,48	
Road %	61,53	49,64	66,65	42,9	49,5	53,48	62,49	
CECF	1,88	1,84	1,87	1,86	1,76	1,89	1,88	
	35,23	34,84	32,97	43,5	34,39	25,01	39,27	
	79,83	77,56	85,8	76,91	82,58	84,18	81,02	
	60,13	57,4	54,55	55,97	59,77	61,16	54,37	
	1,91	1,91	1,89	1,83	1,82	1,98	1,98	
AVG CECF	32,42	42,07	23,24	52,5	31,33	53,25	31,32	
	85,43	76,45	83,69	84,66	86,46	85,17	84,04	
	63,55	51,51	65,66	40,81	57,32	38,23	54,86	
	1,89	1,85	1,89	1,84	1,85	1,81	2,03	
AVG CECF	1,89	1,87	1,88	1,84	1,81	1,89	1,96	

**Evaluation of three features***Table 14: Statistics of combinations of three features*

mean								
Contrast								
Dissimilarity								
Homogeneity								
Energy								
Correlation								
ASM								
Others %	48,15	43,08	42,71	41,75	39,85	45,39	46,39	38,5
Water %	84,12	87,97	88,63	87,34	87,14	88,16	86,96	89,01
Road %	48,93	50,89	52,54	51,89	56,83	51,12	50,2	53,69
CECF	1,69	1,78	1,75	1,81	1,75	1,74	1,68	1,88
	45,87	34,9	44,52	36,48	36,55	44,59	46,97	42,88
	87,64	88,59	85,65	87,54	88,15	87,12	86,66	86,39
	50,47	55,09	52,48	58,06	59,62	49,1	49,78	54,83
	1,73	1,87	1,74	1,82	1,85	1,69	1,71	1,73
	40,26	40,17	47	29,61	43,02	33,91	42,45	45,54
	87,13	89,26	86,59	85,05	81,47	87,19	87,18	83,1
	55,16	51,2	45,97	67,36	52,61	60,75	57,82	56,12
	1,81	1,85	1,74	1,91	1,76	1,88	1,75	1,69
	46,43	36,61	40,56	38,83	34,35	43,76	46,72	44,54
	90,04	88,99	88,51	87,48	78,83	87,2	87,12	87,37
	46	58,91	53,93	53,79	61,13	51,7	49,07	51,82
	1,77	1,86	1,81	1,86	1,86	1,76	1,73	1,71
AVG CECF	43,2	46,97	38,29	41,3	47,87	47,24	42,13	51,18
	87,9	87,93	88,57	86,78	89,83	84,73	86,13	87,24
	50,91	48,42	54,77	52,89	35,64	50,74	54,8	48,09
	1,83	1,75	1,88	1,84	1,93	1,6	1,76	1,66
AVG CECF	1,766	1,814	1,784	1,848	1,83	1,734	1,726	1,734

Table 15: Continuance of table 14

mean							
Contrast							
Dissimilarity							
Homogeneity							
Energy							
Correlation							
ASM							
Others %	41,39	49,85	61,02	46,58	49,97	51,98	40,1
Water %	87,11	79,12	80,15	80,75	80,3	78,34	88,29
Road %	57,05	60,23	49,41	65,91	54,91	56,26	46,01
CECF	1,72	1,66	1,54	1,69	1,67	1,61	1,97
	37,64	67,96	55,32	58,37	52,72	55,59	32,99
	86,78	80,25	80,81	82,15	78,83	78,88	83,66
	57,8	39,35	55,78	50,64	53,54	53,78	61,21
	1,84	1,48	1,6	1,6	1,61	1,57	1,94
	30,79	58,48	52,24	59,99	60,39	59,45	37,46
	86,86	82,19	79	79,58	79,54	77,49	85,23
	66,11	49,23	60,03	53,45	42,66	48,52	51,24
	1,79	1,58	1,62	1,52	1,57	1,56	1,96
	40,82	49,18	55,11	51,57	58,39	53,45	46,7
	88,06	80,43	79,78	78,16	78,04	78,48	82,74
	51,74	64,73	55,74	61,78	51,86	56,03	45,16
	1,81	1,63	1,6	1,61	1,57	1,61	1,83
	28,92	56,08	58,14	54,49	45,44	67,76	43,19
	87,53	79,4	82,57	80,48	79,01	80,11	86,31
	66,03	56,39	49,26	56,66	60,92	44,98	50,69
	1,84	1,57	1,57	1,62	1,69	1,45	1,83
AVG CECF	1,8	1,584	1,586	1,608	1,622	1,56	1,906

**Evaluation of four features***Table 16: Statistics of combinations of four features*

	mean							
Contrast								
Dissimilarity								
Homogeneity								
Energy								
Correlation								
ASM								
Others %	48,78	46,87	43,78	49	59,78	51,13	54,18	58,48
Water %	87,38	87,92	87,37	87,35	80,3	82,87	80,21	79,98
Road %	49,16	94,86	52,59	48,11	53,06	58,9	60,54	45,65
CECF	1,7	1,71	1,76	1,68	1,52	1,68	1,57	1,59
	44,14	40,06	38,68	43,09	62,92	53,41	65,87	55,66
	88,15	86,8	86,51	86,3	80,18	80,21	79,99	79,23
	50,21	56,59	61,63	53,77	47,37	57,61	45,16	52,76
	1,76	1,77	1,73	1,75	1,52	1,63	1,47	1,58
	40,12	46,42	40,29	40,81	60,08	64,35	62,43	43,02
	87,96	87,17	87,63	86,39	80,8	80,81	82,76	80,14
	54,55	53,64	57,14	58,71	49,06	48,8	47,56	62,27
	1,82	1,72	1,82	1,74	1,58	1,47	1,54	1,75
	51,07	51	39,35	32,12	57,65	60,28	62,36	55,26
	87,29	86,98	87,38	87,34	80,67	80,93	82,15	81,47
	46,71	45,85	56,64	64,58	54,82	48,69	44,65	52,09
	1,66	1,65	1,85	1,93	1,57	1,56	1,57	1,63
AVG CECF	45,71	39,05	41,03	50,62	56,17	58,45	59,81	46,21
	86,52	86,5	88,16	86,82	81,66	82,28	80,96	79,3
	52,35	61,09	54,79	47,66	54,93	54,11	53,05	61,84
	1,69	1,82	1,86	1,61	1,63	1,58	1,55	1,66
AVG CECF	1,726	1,734	1,804	1,742	1,564	1,584	1,54	1,642

Table 17: Continuance of table 16

	mean							
Contrast								
Dissimilarity								
Homogeneity								
Energy								
Correlation								
ASM								
Others %	55,01	46,35	60,73	50,57	60,07	63,81	58,56	56,04
Water %	80,55	86,02	80,68	82,46	79,35	76,97	79,22	78,89
Road %	50,37	45	51,85	57,21	52,65	48,65	51,83	53,08
CECF	1,65	1,76	1,54	1,72	1,51	1,48	1,55	1,6
	48,07	43,85	61,03	59,03	54,14	50,67	60,9	58,68
	80,47	86,86	80,27	80,73	81,39	77,79	74,51	78,42
	59,57	48,03	48,98	51,99	55,8	62,78	52,37	48,33
	1,71	1,86	1,52	1,56	1,61	1,65	1,49	1,6
	45,67	46,85	60,06	61,95	57,49	45,49	60,67	56,45
	80,43	87,77	81,2	81,34	80,26	80	77,87	79,34
	60,71	45,12	47,6	50,18	56,94	63,37	53,03	52,96
	1,7	1,79	1,57	1,54	1,54	1,74	1,52	1,62
	39,28	45,03	55,7	64,1	52,53	58,41	57,24	49,73
	76,85	88,1	80,57	81,57	81,67	78,25	79,69	79,42
	69,45	46,69	56,54	46,6	58,18	55,79	51,21	59,08
	1,72	1,82	1,6	1,51	1,66	1,55	1,59	1,67
	55,5	37,78	60,87	60,84	64,92	58,3	61,56	59,96
	80,12	86,63	80,65	79,65	79,13	75,97	77,29	79,12
	50,79	58,25	50,92	48,89	51,4	56,63	50,42	47,99
	1,61	1,86	1,52	1,54	1,44	1,52	1,52	1,57
AVG CECF	1,678	1,818	1,55	1,574	1,552	1,588	1,534	1,612

Table 18: Continuance of table 16

	mean			
Contrast				
Dissimilarity				
Homogeneity				
Energy				
Correlation				
ASM				
Others %	41,62	51,1	50,33	59,68
Water %	88,6	80,49	82,54	83,42
Road %	55,33	56,86	52,85	49,6
CECF	1,81	1,6	1,63	1,51
	44,8	47,94	45,29	49,13
	87,42	78,61	80,18	80,41
	53,41	57,6	58,61	59,36
	1,81	1,67	1,71	1,68
	47,97	44,43	38,27	52,99
	89,06	83,3	83,29	79,2
	49,07	59,95	62,84	54,85
	1,76	1,7	1,85	1,61
	50,35	39,7	50,26	59,04
	88,21	80,08	79,71	81,26
	47,47	65,21	57,74	52,18
	1,73	1,76	1,62	1,52
	43,48	53,89	32,28	52,76
	89,53	83	84,86	80,73
	47,58	52,01	67,97	58,41
	1,88	1,57	1,8	1,6
AVG CECF	1,798	1,66	1,722	1,584

**Evaluation of five features***Table 19: Statistics of combinations of five features*

mean								
Contrast								
Dissimilarity								
Homogeniety								
Energi								
Correlation								
ASM								
Others %	56,39	64,26	61,47	42,7	64,16	60,36	50,81	54,18
Water %	81,31	80,33	79,78	80,38	77,93	80,04	80,77	84,26
Road %	55,01	47,68	50,9	65,11	48	47,65	59,32	47,43
CECF	1,59	1,47	1,53	1,78	1,48	1,57	1,67	1,64
	45,47	53,94	49,02	43,9	58,82	54,71	53,6	45,02
	81,85	81,94	83,75	80,18	78,63	79,2	79,09	86,62
	65,58	57,17	59,28	65,88	53,36	56,9	55,8	53,26
	1,66	1,66	1,67	1,75	1,53	1,6	1,62	1,78
	45,93	55,08	38,09	46,65	65,98	58,43	53,44	39,73
	80,24	83,12	84,6	82,28	78,96	80,86	79,5	87,64
	66,3	57,52	70,05	62,31	41,91	49,65	58,14	56,58
	1,65	1,56	1,79	1,64	1,47	1,6	1,6	1,85
	59,18	44,02	50,86	54,99	65,2	55,47	35,44	51,37
	80,77	82,65	83,26	79,04	79,78	79,86	82,77	86,87
	53,15	63,79	60,53	59,13	47,57	54,62	69,22	48,12
	1,5	1,7	1,59	1,52	1,47	1,59	1,86	1,67
AVR CECF	1,6	1,5975	1,645	1,6725	1,4875	1,59	1,6875	1,735

Table 20: Continuance of table 19

mean							
Contrast							
Dissimilarity							
Homogeniety							
Energi							
Correlation							
ASM							
Others %	49,46	55,6	53,25	64,91	66,42	56,9	64,27
Water %	80,06	80,91	80,81	79,32	78,14	79,78	78,42
Road %	60,93	58	52,75	45,81	47,64	51,6	47,07
CECF	1,65	1,61	1,65	1,5	1,43	1,6	1,48
	50,63	55,05	43,8	57,72	65,34	52,66	61,49
	78,02	81,37	81,9	81,49	79,91	79,48	78,77
	61,42	56,64	58,99	53,29	43,27	60,43	54,52
	1,6	1,63	1,77	1,6	1,5	1,63	1,51
	49,82	53,95	64,31	52,42	57,29	55,2	62,05
	81,27	81,41	79,06	80,2	78,81	80,56	79,65
	57,13	59,47	45,43	59,93	56,05	57,82	47,65
	1,69	1,66	1,49	1,65	1,55	1,62	1,56
	56,7	48,79	60,04	56,18	56,07	53,46	58,01
	80,42	82,21	81,18	80,91	78,67	81,19	81,72
	48,61	62,22	49,42	57,27	56,43	56,74	49,41
	1,59	1,73	1,57	1,6	1,57	1,64	1,61
AVR CECF	1,6325	1,6575	1,62	1,5875	1,5125	1,6225	1,54

### Evaluation of three features

*Table 21: Statistics with combinations of six and seven features*

	mean						
Contrast							
Dissimilarity							
Homogeneity							
Energy							
Correlation							
ASM							
Others %	57,05	61,34	59,94	55,77	57,1	61,9	55,73
Water %	79,96	80,13	78,57	79,51	81,63	81,24	80,05
Roads %	56,12	47,47	51,41	56,57	51,93	51,2	57,24
CECF	1,6	1,57	1,54	1,63	1,6	1,5	1,59
	56,72	61,87	59,46	59,9	55,22	60,88	49,75
	78,86	77,47	78,51	78,97	80,75	80,37	79,7
	56,77	50,49	53,82	52,7	51,11	52,62	61,54
	1,68	1,49	1,53	1,54	1,64	1,55	1,71
	50,34	64,99	54,53	58,29	49,41	53,69	60,63
	77,48	78,43	78,89	79,26	80,5	81,15	78,8
	62,17	49,08	57,38	54,61	59,4	57,71	51,97
	1,68	1,46	1,61	1,6	1,64	1,65	1,55
	57,85	53	56,59	62,91	54,63	53,91	60,1
	78,74	79,15	79,83	80,42	80,64	80,76	79,33
	55,38	61,3	54,92	49,09	53,08	59,34	52,41
	1,59	1,6	1,6	1,51	1,6	1,63	1,52
AVR CECF	62,25	56,32	56,69	67,07	48,93	60,47	63,59
	80,33	79,89	79,37	77,21	80,84	80,61	77,88
	48,34	52,32	54,7	47,45	55,65	54,99	48,77
	1,53	1,63	1,58	1,47	1,73	1,53	1,5
AVR CECF	1,616	1,55	1,572	1,55	1,642	1,572	1,574

### Evaluation of height feature

Table 22: Statistics of feature combination with the height feature added

	mean	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	ASM	Height
Others %	53,95	55,02	64,32	59,29				
Water %	81,1	78,73	81,92	80,15				
Roads %	67,46	69,82	57,05	60,98				
CECF	1,61	1,54	1,4	1,45				
	46,84	58	58,45	60,37				
	83,16	83,62	82,22	79,16				
	67,72	58,02	59,74	62,85				
	1,72	1,55	1,54	1,42				
	55,53	56,36	52,03	60,66				
	81,13	80,68	82,57	80,04				
	64,19	64,04	68,39	62,03				
	1,54	1,59	1,61	1,43				
	56,51	60,45	61,39	59,28				
	82,61	79,1	82,07	81,35				
	62,54	57,89	58,49	64,57				
	1,54	1,48	1,47	1,44				
AVR CECF	1,6025	1,54	1,505	1,435				

### Evaluation of gaussian filtration

Table 23: Statistics of the best feature combination with different sigma for the gaussian filter

	2	4	6	8
mean				
Contrast				
Dissimilarity				
Homogeneity				
Energy				
Correlation				
ASM				
Height				
Gauss	2	4	6	8
Others %	57,07	56,15	54	53,95
Water %	77,56	81,12	80,93	83,26
Roads %	60,23	66,57	69,05	69,44
CECF	1,52	1,53	1,53	1,522
	55,99	61,41	58,06	58,16
	79,6	80,92	82,39	85,97
	65,3	59,33	67,23	62,07
	1,5	1,48	1,42	1,45
	59,95	60,08	56,77	51,85
	82,24	81,29	83,3	83,9
	60,58	59,59	62,8	77,33
	1,47	1,5	1,51	1,51
AVR CECF	1,496666667	1,503333333	1,486666667	1,494

### Evaluation of median filter

*Table 24: Statistics of using different sizes of the median filter*

	3x3	5x5	7x7	9x9
mean				
Contrast				
Dissimilarity				
Homogeneity				
Energy				
Correlation				
ASM				
Height				
Mean	3x3	5x5	7x7	9x9
Others %	59,3	56,33	52,95	56,74
Water %	78,46	80,73	80,85	79,56
Roads %	63,62	64,49	66,43	63,29
CECF	1,43	1,54	1,61	1,56
	51,33	53,51	60,31	60,39
	79,15	80,22	80,18	79,55
	68,82	63,33	60,29	60,4
	1,6	1,62	1,47	1,43
	54,82	64,36	54,94	55,5
	81,12	80,92	78,69	80,29
	61,58	55,75	65,38	62,89
	1,57	1,4	1,56	1,55
AVR CECF	1,533	1,52	1,547	1,513